Universidade de Lisboa

Instituto Superior Técnico

Socio-Emotional Reward Design for Intrinsically Motivated Learning Agents

Pedro Daniel Barbosa Sequeira

Supervisor: Doctor Ana Maria Severino de Almeida e Paiva

Co-Supervisor: Doctor Francisco António Chaves Saraiva de Melo

Thesis approved in public session to obtain the PhD Degree in

Information Systems and Computer Engineering

Jury final classification: Pass with Merit

Jury

Chairperson: Chairman of the IST Scientific Board

Members of the Committee:

Doctor Eugénio da Costa Oliveira, Full Professor at Faculdade de Engenharia, Universidade do Porto

Doctor Pedro Manuel Urbano de Almeida Lima, Associate Professor at Instituto Superior Técnico, Universidade de Lisboa

Doctor Ana Maria Severino de Almeida e Paiva, Associate Professor at Instituto Superior Técnico, Universidade de Lisboa

Doctor Douwe Joost Broekens, Assistant Professor at MMI, Delft University of Tecnology, The Netherlands

Doctor Francisco António Chaves Saraiva de Melo, Assistant Professor at Instituto Superior Técnico, Universidade de Lisboa

Doctor Manuel Cabido Lopes, Researcher at Institut National de Recherche en Informatique et en Automatique (INRIA), France

Funding Institutions

Fundação para a Ciência e Tecnologia through PhD grant SFRH/BD/38681/2007 and under the EU project SEMIRA through grant ERA-Compl/0002/2009

2013

Abstract

Reinforcement learning (RL) is a computational approach which models autonomous agents facing a sequential decision problem in a dynamic environment. The behavior of the agent is guided by a reward mechanism embedded into the agent by its designer. Designing flexible reward mechanisms, capable of guiding the agent in learning the task intended by its designer, is a very demanding endeavor: on one hand, artificial agents have inherent limitations that often impact the ability to actually solve the task they were initially designed to accomplish; On the other hand, traditional approaches to RL are too restrictive given the agents limitations, potentially leading to poor performances. Therefore, applying RL in complex problems often requires a great amount of manual fine-tuning on the agents so that they perform well in a given scenario, and even more when we want them to operate in a variety of different situations, often involving complex interactions with other agents.

In this thesis we adopt a recent framework for intrinsically-motivated reinforcement learning (IMRL) that proposes the use of richer reward signals related to aspects of the agent's relationship with its environment that may not be directly related with the task intended by its designer. We propose to take inspiration from information processing mechanisms present in natural organisms to build more flexible and robust reward mechanisms for autonomous RL agents. Specifically, we focus on the role of *emotions* as an evolutionary adaptive mechanism and also on the way individuals *interact and cooperate with each other* as a social group.

In a series of experiments, we show that the adaptation of emotion-based signals for the design of rewards within IRML allows us to achieve general-purpose solutions and at the same time alleviate some of the agent's inherent limitations. We also show that social groups of IMRL agents, endowed with a reward mechanism inspired by the way humans and other animals exchange signals between each other, end up maximizing their collective fitness by promoting socially-aware behaviors. Furthermore, by emerging reward signals having dynamic and structural properties that relate to emotions and the way they evolved in nature, we show that emotion-based design might have a greater impact for the adaptation of artificial agents than thought before.

Overall, our results support the claim that, by providing the agents with reward mechanisms inspired by the way that emotions and social mechanisms evaluate and structure natural organisms' interactions with their environment, we provide agent designers with a *flexible* and *robust* reward design principle that is able to overcome common limitations inherent to RL agents.

Keywords: Reinforcement Learning, Intrinsic Motivation, Reward Design, Autonomous Agents, Multiagent Systems, Emotions, Cooperation, Evolution, Appraisal, Social Theories

Resumo

A aprendizagem por reforço (AR) é uma abordagem computacional que modela agentes autónomos com um problema de decisão sequencial com um ambiente dinâmico. O comportamento do agente é guiado por um mecanismo de recompensa incorporado no agente pelo seu construtor. Criar mecanismos de recompensa flexíveis, isto é, capazes de guiar o agente na aprendizagem da tarefa pretendida pelo seu criador, é um esforço muito exigente: por um lado, os agentes artificiais têm limitações inerentes que muitas vezes os impedem de realizar a tarefa para a qual foram inicialmente desenhados; Por outro lado, as abordagens tradicionais em AR propõem soluções que são demasiado restritivas dadas as limitações dos agentes, levando potencialmente ao fraco desempenho por parte destes. Desta forma, o uso de AR em problemas complexos requer muitas vezes uma grande quantidade de afinação manual dos agentes para que eles tenham um bom desempenho num determinado cenário, e ainda mais quando queremos que eles sejam capazes de executar tarefas numa variedade de situações diferentes, envolvendo muitas vezes interacções complexas com outros agentes.

Nesta tese adoptamos uma recente abordagem à aprendizagem por reforço com motivação intrínseca (ARMI), que propõe o uso de sinais de recompensa mais ricos relacionados com aspectos da relação do agente com o seu ambiente que podem não estar directamente relacionados com a tarefa pretendida pelo seu criador. A nossa proposta passa pela inspiraĲŃo em mecanismos de processamento de informação presentes em organismos naturais e a sua os adaptação de forma a construir mecanismos de recompensa para agentes autónomos de AR mais flexíveis e robustos. Mais especificamente, focamo-nos no papel das *emoções* como um mecanismo de adaptação evolutivo na natureza e também na forma como as pessoas *interagem e cooperam umas com as outras* como um grupo social.

Através de uma série de experiências, mostramos que a adaptação de sinais emocionais para o desenho de recompensas no contexto da ARMI nos permite obter soluções robustas e ao mesmo tempo dissimular algumas das limitações perceptuais do agente. Mostramos também que grupos sociais de agentes, dotados de um mecanismo de recompensas inspirado na forma como os humanos e outros animais sinais trocam sinais entre si, acabam por maximizar o seu desempenho colectivo através da promoção de comportamentos socialmente conscientes. Para além disso, ao fazermos emergir sinais de recompensas com propriedades dinâmicas e estruturais relacionadas com as emoções e a forma como estas evoluíram na natureza, mostramos que a inclusão de sinais emocionais pode ter um impacto maior na capacidade adaptativa dos agentes artificiais do que era até agora suposto.

Em geral, os nossos resultados suportam a hipótese de que ao fornecer aos agentes sinais de recompensa inspirados pela forma como as emoções e os mecanismos sociais avaliam e estruturam as interacções dos organismos naturais com o seu ambiente, proporcionamos aos criadores de agentes uma filosofia de desenho de recompensa *flexível* e *robusta* capaz de superar limitações comuns em agentes de AR.

Palavras-chave: Aprendizagem por Reforço, Motivação Intrínseca, Desenho de Recompensas, Agentes Autónomos, Sistemas Multiagente, Emoções, Cooperação, Evolução, Appraisal, Teorias Sociais

Acknowledgments

First of all, this thesis is entirely dedicated to my father. Thank you for, during the course of all these years, investing in me an ability to constantly fight against all odds and never give up in difficult times, to make me believe that one can always do more and do better, and especially for showing me that the future is built by observing the past with serenity.

I would also particularly like to thank Prof. Francisco Melo for having introduced me to the fantastic and challenging area of reinforcement learning. Your guidance, patience, excitement, supervision, and all the improvised workshops, advices and rectifications allowed me to think out of the box, apply ideas, test them, have new ideas, in sum, allowed me to write this thesis as it is.

I also thank Prof. Ana Paiva for believing I had the capacity to do a PhD in the first place. Thank you for all your guidance and vision, and for always making me want to solve new and exciting challenges.

I would like to thank Prof. Stacy Marsella at the USC in Los Angeles for his hospitality, kindness, accessibility and guidance during the most important part of my PhD research. Thank you for all your insights and advices and for making me not be afraid and “get my feet wet”.

I acknowledge all the feedback provided by Prof. Joost Broekens and Prof. Manuel Lopes that allowed for significant improvements of this document.

I also acknowledge all the people from GAIPS that during my PhD gave me useful ideas, advices, critics, references and guidelines. I particularly thank António “Xor Bitore” Brisson for his friendship, companionship and good humor. Thank you for making our workplace a place for fun, reflection and improvisation.

I thank INESC-ID and all associated staff for the support and work conditions provided throughout the years and the Portuguese Fundação para a Ciência e a Tecnologia (FCT) for all the financial support provided by the PhD grant.

Last but not least, I would like to thank my girlfriend Estela for putting up with me all these years and always encouraging me to keep going. Thank you for all your support, help and understanding.

A special mention goes to Sock and Ming, two very inspiring dogs!.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Motivation

Over the past 70 years, the field of artificial intelligence (AI) has been dedicated to studying and designing *intelligent* or *autonomous agents* that can be perceived as "thinking" and "acting" much like humans do (Russell and Norvig, 2003). Given this goal, the design of such agents has been greatly influenced by the way humans and other animals interact with their environment (Maes, 1994). Similar to the way biological organisms live and interact in nature, autonomous agents are entities that inhabit dynamic and often unpredictable environments, where their *actions* are influenced by a set of *goals* together with information *perceived* from the environment through its sensors (Franklin and Graesser, 1997; Maes, 1994; Russell and Norvig, 2003). As such, one of the major challenges within AI has do with building mechanisms that enable the agent to "correctly" map its perceptions to actions in order to achieve its goals *on its own* and *while interacting* with the environment.

Provided such challenge, this thesis is motivated by the design of more *autonomous*, *flexible* and *robust* mechanisms for intelligent agents. Generally speaking, this means that we want to build agents that require less the intervention of humans or other external agents—autonomy—and that are able to operate on a larger variety of different domains each offering distinct challenges—flexibility and robustness (Franklin and Graesser, 1997; Russell and Norvig, 2003). The ability of an agent, either natural or artificial, to adjust its behavior according to changes perceived in its environment is a process known as *learning* (Anderson, 2000). For autonomous agents, this is a key factor for it to "perform well" according to its designers' expectations (Maes, 1994). Within the field of machine learning, reinforcement learning (RL) is precisely the discipline concerned with providing mechanisms that allow an agent to accomplish a task through trial-and-error interactions with a dynamic environment(Kaelbling et al., 1996; Sutton and Barto, 1998). Throughout this thesis we will dedicate our attention to the design of learning mechanisms for RL agents.

1.2 Problem

RL models autonomous agents facing a sequential decision problem of learning *what to do* given a certain situation in order to achieve a particular goal. The *situation* describes the agent's state of affairs which is relevant for it to solve the given task. The *goal* is often characterized as the maximization of a certain numerical environmental feedback signal, also referred to as the *reinforcement*, *reward* or *return*. By analyzing the reward received throughout time, the agent *learns* a mapping from situations to actions which is referred to as a *policy* (Kaelbling et al., 1996; Sutton and Barto, 1998).

RL has been successfully applied to develop systems that autonomously learn from experience. Common applications of RL include game-playing agents (Samuel, 1959; Schaeffer et al., 2001; Tesauro, 1995; Yan et al., 2005), mobile robotic control (Bagnell and Schneider, 2001; Bentivegna et al., 2002; Fidelman and Stone, 2004; Kohl and Stone, 2004; Kwok and Fox, 2004; Ng et al., 2006), operations research (Abe et al., 2004; Proper and Tadepalli, 2006; Rusmevichientong et al., 2006), human-computer interaction (HCI) (Isbell et al., 2006; Singh et al., 2002) and others (Moody and Saffell, 2001; Stone and Sutton, 2001).

A major problem faced by learning agents, both natural and artificial, is the fact that they suffer from perceptual, motor and adaptive limitations: they often do not have access to "all" the information necessary to make the best decisions and normally do not know the environment's dynamics or the exact consequences of their actions. All of this makes it difficult to cope with abrupt and profound changes in the environment.

Given such limitations and in light of our motivation to build more autonomous, flexible and robust mechanisms for RL agents, existing RL techniques present several design challenges. This is especially true when applying RL to practical applications and complex problems, where designers have to provide the agent with the necessary built-in information and skills for it to correctly learn a given task, often involving a great amount of fine-tuning and user expert knowledge (Kaelbling et al., 1996; Littman, 1994; Singh et al., 1994; Sutton and Barto, 1998). Moreover, the perceptual limitations of the agents make the design assumptions in classical RL methods to be too restrictive, potentially leading to poor performances by the agent (Kaelbling et al., 1996; Littman, 1994; Loch and Singh, 1998; Singh et al., 1994; Sutton and Barto, 1998). Also, in complex environments, the demand for *sufficient* built-in knowledge is particularly critical as sometimes it is impossible for the agent's designer to know beforehand what the exact optimal behavior is in *some* situation, let alone design mechanisms that are flexible enough to be applied in a variety of different situations (Sorg et al., 2010a).

In this thesis we focus on the *reward mechanism* of RL agents as a means to alleviate some of the aforementioned problems. In RL, the reward mechanism that the designer builds into the agent guides it throughout learning and implicitly defines the *task* that it must accomplish (Sutton and Barto, 1998). It critically impacts both the *time* taken to learn a task and *what is learned* (Kaelbling et al., 1996). As such, building a "good" reward mechanism[1] is crucial for the performance of the agent. A major challenge when designing RL agents is then to build reward mechanisms that allow them to learn the intended task as efficiently as possible (Abbeel and Ng, 2004; Ng and Russell, 2000; Sorg et al., 2010a). We can therefore state the research problem behind this thesis as:

> We want to *design reward mechanisms* for RL agents that are able to *alleviate their inherent perceptual limitations* and make them *operate in a wide variety of domains* without the explicit intervention of others or relying on expert or domain knowledge about a particular task.

[1] We define the specific criteria to evaluate a reward mechanism throughout the thesis. Generally speaking, we can define a good reward mechanism as one that enables the agent to learn the intended task in the shortest time possible.

1.3 Approach

We have established the importance of designing good reward mechanisms that take the most out of the agent's perceptions in order to circumvent some of its perceptual limitations. In order to address the above-stated problem one must consider how to design the reward mechanism and, most importantly, which kinds of rewards we want to provide our agents. In that respect, we follow a recent framework for intrinsically-motivated reinforcement learning (IRML) (Singh et al., 2009, 2010), in which an agent is rewarded for behaviors other than those strictly related to the task being accomplished, *e.g.*, by exploring or playing with elements of its environment. Interestingly, studies performed using this approach have demonstrated the usefulness of such intrinsic rewards in mitigating inherent computational limitations of learning agents (Bratman et al., 2012; Singh et al., 2009, 2010; Sorg et al., 2010a). The question remains, however, about which kinds of rewards are able to both alleviate perceptual limitations and be generic enough to be used in a variety of domains without the need for fine adjustments for a particular situation.

To answer that question we return to the parallel between natural and artificial agents drawn earlier and propose to look at the motivational mechanisms that humans and other animals inherently possess that enable them to solve complex tasks. The rationale behind our approach stems from the fact that, just like autonomous agents, animals inhabit very dynamic and sometimes unpredictable environments and are also unable to perceive everything from their environment. Still they are able to adapt to demanding conditions and overcome difficult situations, *e.g.*, in face of imminent danger.

Furthermore, humans and other animals do not inhabit their environments by themselves. They live within social groups as a way to augment their survival chances, *e.g.*, by hunting or performing complex actions together. Nevertheless they still compete for resources or dispute social status as a way to improve their reproductive power. Throughout evolution, nature has also shaped *social mechanisms* that facilitate the way animals communicate their intentions and evaluate the actions of others, even in inherently competitive settings.

Having into account this analogy between biological and artificial agents, we describe our hypothesis to address the problem defined above through the following statement:

> In this thesis we take inspiration from information processing mechanisms shaped throughout evolution that are behind the adaptive success of humans and other biological organisms. We focus on the role of *emotions* and also on the way individuals *interact and cooperate with each other* as a social group to design more *flexible* and *robust* reward mechanisms that enhance the *autonomy* of RL agents in both single and multiagent settings.

Specifically, we start by analyzing the role of emotions as a powerful adaptive mechanism that influences cognitive and perceptual processing (Cardinal et al., 2002; Dawkins, 2000; Phelps and

LeDoux, 2005). Emotions indirectly drive behaviors that lead individuals to achieve their goals, and the absence of such mechanism was shown to impact the ability to properly take advantageous decisions (Bechara et al., 2000; Damasio, 1994; LeDoux, 2000). We therefore take inspiration from this role of emotions and propose the design of a general-purpose, domain-independent intrinsic reward mechanism that, much like emotions do in natural organisms, evaluates the agent's history of interaction with its environment and enables it to solve complex tasks under limited perception conditions.

We also study some social mechanisms of humans and other animals that allow for cooperation to occur even in competitive contexts. We follow the idea that closely-related individuals share a *need for affiliation* that makes them engage in altruistic behaviors despite possible momentary losses for the contributor (Axelrod, 1984; de Waal, 2008; Dörner, 1999; Hamilton, 1964; Trivers, 1971). We take inspiration from mechanisms of *reciprocation* by which animals evaluate the "kindness" of others' actions and respond accordingly (Axelrod, 1984; Falk and Fischbacher, 2006; Trivers, 1971) to design intrinsic social rewards that enable the emergence of "socially-ware" individual behaviors within competitive multiagent settings.

1.4 Contributions

We now examine in more detail the contributions stemming from our approach. The results from this thesis have implications to the field of RL and also to *affective computing* (AC) and other research fields within the computational sciences. Specifically:

1. We propose a set of domain-independent emotion-based reward features, namely *novelty*, *valence*, *goal relevance* and *control*, based on dimensions of *appraisal of the emotional significance of events*, commonly found in the psychology literature.

   (a) We show that the proposed rewards provide a general-purpose guiding system for autonomous learning agents, alleviating the need for fine-tuning of reward functions to specific environments;

   (b) On the other hand, the design of rewards inspired by emotions endows learning agents with some of the benefits that emotions bring to natural agents, namely a greater *adaptability* to different dynamic and unpredictable environments and the *mitigation of perceptual limitations* commonly found in RL agents;

   (c) We consider our mechanism *adaptive* in that the emotion-based rewards are generic enough to allow an optimization procedure to select the best emotional agent for a particular set of environments without having to adjust the RL algorithm's parameters or the state representation for that particular domain;

   (d) Our approach departs from previous works within the field of affective computing

(AC) advocating the need for emotion-based mechanisms by using domain-independent appraisal-based reward features that are independent of the particular RL method used and also can be applied in a variety of domains, instead of focusing on a small set of basic emotions or defining a set of emotion-based rules controlling the agent's behavior;

2. We propose an information-processing reward mechanism having evaluative properties very similar to those proposed by appraisal theories of emotions. This mechanism is composed of features that focus on sources of information emerged by means of an evolutionary algorithm using genetic programming;

   (a) We used a novel method for assessing the significance of embedding emotions into artificial agents that, unlike previous approaches within AC, uses an *evolutionary computation mechanism* for the emergence of emotion-like informative signals;

   (b) The evolutionary procedure resembles the way emotions evolved in nature. This enables an interesting computational parallel to what occurs in nature, where emotions allow individuals to better adapt to their environments, and where arguably the most fit species (humans) is that with more complex emotional mechanisms;

   (c) The evolutionary procedure emerged four domain-independent, general purpose reward features that can be applied in different scenarios, thus leading to the construction of more robust, flexible and autonomous agents;

   (d) The emerged signals have dynamic properties that *resemble the way humans evaluate their environment*, according to appraisal theories of emotions;

   (e) We go beyond the idea that emotions are only a *useful* mechanism for autonomous agents. Our results point towards the idea of emotions having a *major impact* for the adaptation of artificial agents to their environments than thought before;

3. We extend previous work on IMRL to multiagent scenarios. We develop reward mechanisms for IMRL agents that take into consideration social interactions among agents, inspired by the notion of *group affiliation* from multiagent systems;

   (a) We show that the IMRL framework can be used to endow agents with *social motivation* that drives them to learn behaviors that benefit the whole population of agents;

   (b) We show that it is possible for agents to *individually* acquire *socially aware behaviors* that trade-off individual well-fare for social acknowledgment, leading to a more successful performance of populations as a whole;

   (c) Our experiments are in accordance with several social theories claiming that altruistic and cooperative behaviors can thrive in populations with mechanisms capable of rewarding or punishing behaviors that are considerate (or not) for the well-being of the social group in which they are inserted.

1.5 Thesis Organization

This thesis is organized as follows:

- In Chapter 2 we provide technical background on RL and IMRL, necessary to set up nomenclature and notation. We also identify a number of computational limitations and challenges commonly found in traditional approaches to RL that will be used throughout the thesis to demonstrate the potential of our approach;
- Chapter 3 provides theoretical background on emotions, the main source of inspiration for our approach presented in the following chapters. We discuss the importance of emotions in nature for the survival of biological organisms. We also overview some works on the field of AC that relate to our approach within this thesis;
- In Chapter 4 we present our proposal for the design of an emotion-based reward mechanism inspired by appraisal theories of emotions. We show that emotions can be a powerful mechanism when adapted into reward signals in a series of learning experiments within IMRL. We also discuss the relationship between our approach and the intrinsic motivation provided by emotions to natural agents;
- In Chapter 5, by means of an evolutionary computation procedure using genetic programming, we show that emotions are a natural candidate when considering possible sources of information available to an IMRL agent that relate to its relationship with the environment. We show that the reward mechanism that best guides an agent through a series of learning scenarios has characteristics that can be related with the way emotions evaluate events in nature;
- Chapter 6 presents the first steps towards extending the IMRL framework to multiagent scenarios. We simulate social groups of agents having a reward mechanism inspired by the way humans signal their behaviors and social interactions. We test our approach in multiagent foraging scenarios where groups of agents compete for food resources. We show that the social group gains strength as a whole when the agents exhibit cooperative behaviors;
- In Chapter 7 we discuss in greater detail the overall contributions of this thesis and provide future research directions.

CHAPTER 2

Computational Reinforcement Learning and Intrinsic Motivation

In order to better understand the framework of computational reinforcement learning (RL) and take meaning of where our contributions fit within this paradigm, in this chapter we provide the necessary technical background on RL. We present the traditional model for RL and the recently proposed framework for intrinsically motivated reinforcement learning (IMRL) that we will use as a testbed for our experiments throughout the thesis. We also discuss in detail some challenges within RL that motivated the appearance of IMRL, and how does our contribution fit within this framework.

2.1 Introduction

Over the years, autonomous agent design has been greatly inspired by the way humans and other animals learn by interacting with their environment, and a striking example of such line of thought is the field of RL which, in a sense, is the computational counterpart of operant conditioning[1] (Sutton and Barto, 1998; Watkins, 1989). In fact, some of the first RL algorithms were based on the behaviorist[2] paradigms of classical and operant conditioning (*e.g.*, Rescorla and Wagner, 1972; Sutton and Barto, 1981, 1987). Much like biological learning from the perspective of behaviorism, computational RL also approaches learning by studying the *changes* occurring in behavior through trial-and-error interactions with a dynamic environment (Kaelbling et al., 1996). Despite these similarities, RL is more concerned at formulating classes of learning problems and developing algorithms to solve them, independently of the biological validity of the methods used (Kaelbling et al., 1996; Sutton and Barto, 1998). RL is one of the most successful approaches to enable autonomous agents to learn from interactions with their (often unpredictable) environment (Kaelbling et al., 1996).

One of the things that makes this framework so attractive is the fact that it assumes learning under uncertainty in a dynamic environment. The agent learns "on its own" with no examples of correct behavior being provided. Through consecutive interactions with the environment it continuously evaluates its own actions in relation to the observations being made.

2.2 The Reinforcement Learning Problem

Reinforcement learning (RL) addresses the general problem of an agent faced with a sequential decision problem (Kaelbling et al., 1996; Sutton and Barto, 1998). The RL model is depicted in Figure 2.1. By a process of trial-and-error, the agent's *decision making* component must learn a "good" mapping that assigns *states* perceived from its environment to *actions*. Such mapping

[1] Skinner (1938) developed the theory of *operant* or *instrumental conditioning* to describe the learning that takes place when some positive (rewarding) or negative (punishing) *reinforcers* change the relative frequency of neutral responses known as *free operants* in the environment (Anderson, 2000).

[2] Behaviorism is an approach to psychology that appeared in the beginning of the 20th century by the hands of Watson (1913). Behaviorists developed theories about the behavior of organisms with *no reference to the mental processes* associated therein.

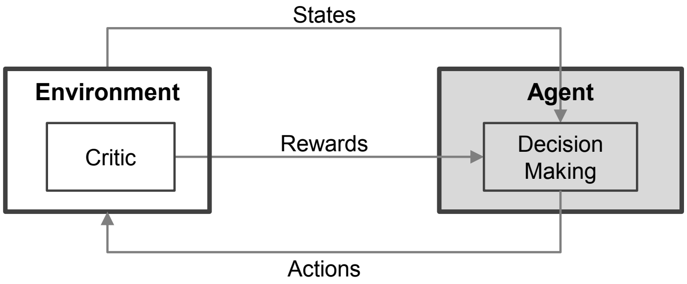


Figure 2.1: The traditional reinforcement learning model in which a (external) critic evaluates the performance of the agent and provides the rewards to the decision-making module. Adapted from (Singh et al., 2010).

determines how the agent acts in each possible situation and is commonly known as a *policy*. A *critic* situated in the environment provides a feedback signal at each time-step, know as the *reward*, that basically indicates the appropriateness of having executed some action given a certain state.

2.2.1 Running Example: Moving Preys Scenario

Let us consider the following grid-world scenario that we use as a running example throughout this chapter. Figure 2.2 shows the environment used in the Moving Preys Scenario, which is inspired in the foraging environment in (Singh et al., 2010).

> **Moving Preys Scenario**: We model our agent as a predator, represented by a fox, trying at each time-step to eat a prey, represented as a hare, which is available in one of the three possible end-of-corridor positions, for example as depicted in Figure 2.2. At each time-step the predator is located in one of the cells of the environment and its internal state can be described as being either hungry or full. Whenever the predator eats a hare it becomes full for the duration of one time-step and another prey appears in a different end-of-corridor location of the environment *chosen randomly*. The agent then becomes hungry until capturing another prey. Our objective is to model an agent that “eats” as many preys as possible during its lifetime. ◊

2.2.2 Partially-Observable Markov Decision Processes

Formally, the sequential decision problem faced by the agent can be modeled as a *partially-observable Markov decision process* (POMDP) (Cassandra, 1994, 1998; Kaelbling et al., 1998), denoted as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{Z}, \mathsf{P}, \mathsf{O}, r, \gamma)$, where

- $\mathcal{S}$ is the set of all possible environment states;
- $\mathcal{A}$ is the action repertoire of the agent;
- $\mathcal{Z}$ is the set of all possible agent observations;

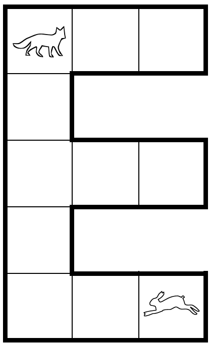


Figure 2.2: The environment for the Moving Preys Scenario, inspired by the foraging environment in (Singh et al., 2010). The agent tries to capture a prey, represented by a hare, which at each time-step can be in one of the three end-of-corridor positions in the environment. See text for more details.

- $\mathsf{P}(s' \mid s, a)$ indicates the probability that the state at time-step $t+1$, s_{t+1}, is s', given that the state at time-step t, s_t, is s and the agent selected action $a_t = a$. Formally,

$$\mathsf{P}(s' \mid s, a) = \mathbb{P}\left[s_{t+1} = s' \mid s_t = s, a_t = a\right].$$

- $\mathsf{O}(z \mid s, a)$ indicates the probability that the observation of the agent at time-step $t+1$, z_{t+1}, is z, given that the state at time $t+1$ is s and the agent selected action a at time t. Formally,

$$\mathsf{O}(z \mid s, a) = \mathbb{P}\left[z_{t+1} = z \mid s_{t+1} = s, a_t = a\right].$$

- $r(s, a)$ represents the *average reward* that the agent expects to receive for performing action a in state s.

- $0 \leq \gamma < 1$ is some *discount factor*.

A POMDP evolves as follows. At every discrete time-step $t = 0, 1, 2, 3, \ldots$, the environment is in some state $s_t = s$. The agent perceives an *observation* $z_t = z$ that depends on but is often insufficient for the agent to unambiguously infer the underlying state of the environment s. The agent then performs an action $a_t = a$ and the environment transitions from state s to state $s_{t+1} = s'$ with probability $\mathsf{P}(s' \mid s, a)$. The agent receives a *reward* $r(s, a) \in \mathbb{R}$ representing the *desirability* for having executed action a in state s. Besides receiving the reward, the agent also makes a new observation $z_{t+1} = z$ with probability $\mathsf{O}(z \mid s, a)$, and this process repeats.

Markov Decision Processes

Typical approaches to RL mainly focus on scenarios where $\mathcal{Z} = \mathcal{S}$ and $\mathsf{O}(z \mid s, a) = \delta(z, s)$, where δ denotes the Kronecker delta (Sutton and Barto, 1998), *i.e.*, where the observations z_t do allow the agent to unambiguously determine the underlying state s_t. Such scenarios are said to have *full* *observability*, *i.e.*, we can predict the next state and expected reward at any given time from only the current state and action (Puterman, 2009; Sutton and Barto, 1998). When this is the case, the POMDP parameters $\mathcal{Z}$ and O can be safely discarded. The simplified model thus obtained, represented as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathsf{P}, r, \gamma)$, is referred as a *Markov decision process* (MDP) (Puterman, 2009).

Resorting to the Moving Preys Scenario, we can now model our problem as an MDP in which the state at time-step t, s_t, describes both the state of the prey and the state of the predator at time-step t, *i.e.*, $s_t = (s_{prey,t}, s_{pred,t})$. The state of the prey at time-step t corresponds to its (x, y) position at time-step t, *i.e.*, $s_{prey,t} = P_{prey}$; the state of the predator, on the other hand, corresponds to its position at time-step t, and whether or not it is hungry, *i.e.*, $s_{pred,t} = (P_{pred}, H)$, where H is a boolean flag that is set to 1 whenever the predator is hungry. The state-space, $\mathcal{S}$, is then set of all possible values that s_t can take. In order to move in its environment and eat preys we can define the agent's actions as the set $\mathcal{A} = \{Up, Down, Left, Right, Eat\}$, each movement action moving the agent deterministically one cell in the corresponding direction in the environment. Action Eat consumes a prey whenever the agent is collocated with it and has no effect otherwise, *i.e.*, the agent remains in its current position.

However, if the bold lines delimiting the scenario in Figure 2.2 represent high walls in the environment, our agent will not always be able to observe the current position of the prey P_{prey}. In such cases, we can use the POMDP model described earlier and design the agent as observing only some elements of the underlying state, *e.g.*, by setting the space $\mathcal{Z}$ of possible observations as the set of all possible values of z_t, where $z_t = (s_{prey,t}, s_{pred,t})$, and $s_{prey,t}$ is now a boolean variable that takes the value of 1 only if $P_{pred} = P_{prey}$, *i.e.*, the agent can only observe the prey if collocated with it. Therefore, the exact location of the prey P_{prey} is said to be *hidden* from the agent's observations.

2.2.3 Optimal Policy

In the classical view of RL, the reward function *evaluates* the agent's behavior with respect to its designer's objectives, acting as a *critic* residing in the external environment (Singh et al., 2010), as illustrated in Figure 2.1. The goal of the agent is to choose its actions so as to gather as much reward as possible during its lifespan, according to some model of optimal behavior that tries to optimize some cumulative measure of the received reward (Kaelbling et al., 1996; Sutton and Barto, 1998). For example, the *infinite-horizon discounted reward* model optimizes the expected reward received discounted by γ. Formally, the optimal behavior should then maximize the value

$$v = \mathbb{E}\left[\sum_t \gamma^t r(s_t, a_t)\right]. \tag{2.1}$$

The *reward function* $r(s, a)$ implicitly encodes a *task* that the agent must complete. In order to maximize the value in (2.1), the agent must learn a mapping that, depending on its history of observations and actions, determines the next action that the agent should take. Such mapping, denoted as $\pi : \mathcal{S} \rightarrow \mathcal{A}$, is known as a *policy*, and is typically learned through a process of trial-and-error by some learning algorithm residing in the *decision-making* component inside the agent, as depicted in Figure 2.1. Furthermore, in the case of the infinite-horizon discounted model in MDPs, it is possible to find a deterministic stationary[3] policy $\pi^* : \mathcal{S} \rightarrow \mathcal{A}$ maximizing the value in (2.1) (Bellman, 2003; Kaelbling et al., 1996; Littman, 1994).

Let us return to our running example of the Moving Preys Scenario. Because we want our agent to consume as many preys as possible throughout time, one possibility is to provide a reward of $r = 1$ whenever it is collocated with a prey in the environment, *i.e.*, we can define a reward function $r(s_t, a_t)$ that takes the value of 1 only when $P_{pred} = P_{prey}$ in s_t and $a_t = Eat$ and is 0 for all other states and actions. In this manner, the agent's task is *explicitly determined* by the provided reward function, *i.e.*, its goal can be directly translated as eating as many preys as possible during its lifespan, as this is the only "source" of reward that can be gathered from the environment.

Let us now denote by V^π the *value* obtained by an agent following some policy π, *i.e.*,

$$V^\pi(s) = \mathbb{E}\left[\sum_t \gamma^t r(s_t, a_t) \mid s_0 = s, a_t \sim \pi\right], \tag{2.2}$$

where $a_t \sim \pi$ indicates that the actions a_t are selected according to policy π.

In this thesis we will focus on *memoryless agents* (also known as *reactive agents*), corresponding to agents that select their action based only on their current observation (Littman, 1994). In other words, a memoryless agent necessarily follows a policy $\pi : \mathcal{Z} \rightarrow \mathcal{A}$ that maps each observation $z \in \mathcal{Z}$ directly to an action $\pi(z) \in \mathcal{A}$. If the state is fully observable, then $\mathcal{Z} = \mathcal{S}$ and $z_t = s_t$. When this is the case, there is a policy $\pi^* : \mathcal{S} \rightarrow \mathcal{A}$, known as the *optimal policy*, such that

$$V^{\pi^*}(s) \geq V^\pi(s) \tag{2.3}$$

for any policy π and any initial state $s \in \mathcal{S}$. We write V^* to denote V^{π^*} as the *optimal state-value function*. We can also associate with π^* an *optimal action-value function* $Q^* : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ that verifies the recursive relation:

$$Q^*(s, a) = r(s, a) + \gamma \sum_{s' \in \mathcal{S}} \mathsf{P}(s' \mid s, a) \max_{b \in \mathcal{A}} Q^*(s', b). \tag{2.4}$$

[3] A *deterministic policy* π is a policy that consistently chooses an action $a = \pi(s)$ given the current state s, with probability 1, for all $s \in \mathcal{S}$ (Littman, 1994). A *stationary policy* π is a policy which mapping from states to actions $a = \pi(s)$ does not change over time, for all $a \in \mathcal{A}$ and $s \in \mathcal{S}$ (Singh et al., 1994).

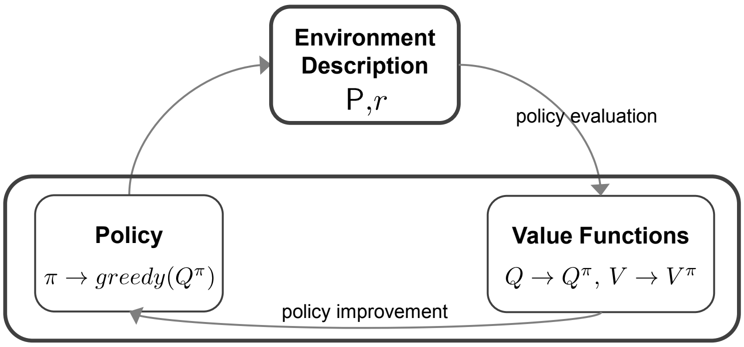


Figure 2.3: Generalized policy iteration (GPI). Adapted from (Sutton and Barto, 1998).

Figure 2.3 depicts the generalized policy iteration (GPI) scheme, in which the processes of policy evaluation—approximating V^* and/or Q^*—and policy improvement—approximating π^*—interact until they are consistent with each other (Sutton and Barto, 1998).

Almost all RL algorithms can be described by the GPI scheme (Sutton and Barto, 1998). Relation (2.4) can be used to iteratively compute Q^* for all pairs $(s, a) \in \mathcal{S} \times \mathcal{A}$. Additionally,

$$V^*(s) = \max_{a \in \mathcal{A}} Q^*(s, a), \tag{2.5}$$

and the optimal policy at state s is given by

$$\pi^*(s) = \operatorname*{argmax}_{a \in \mathcal{A}} Q^*(s, a), \tag{2.6}$$

as denoted by the *greedy* operator in Figure 2.3. From the above, we can restate the goal of the RL agent as that of learning Q^*, since from the latter it is possible to derive the optimal policy π^*.

2.2.4 Learning the Optimal Policy

RL research addresses the problem of designing algorithms to learn the optimal policy π^* from sampled interactions of the agent with its environment, usually by successively updating the approximations of $Q^*(s, a)$ for every possible state-action pair s, a. Common examples of such methods include Q-learning (Watkins, 1989), Dyna-Q/prioritized sweeping (Moore and Atkeson, 1993) and others (Sutton and Barto, 1998).

Since RL agents typically have no knowledge of either the state transitions in P or the rewards in r, one possibility is to *explore* the environment—*i.e.*, select actions in some exploratory manner—building estimates for P and r, denoted by $\hat{\mathsf{P}}$ and $\hat{r}$, respectively, and then using these estimates to successively approximate Q^*. After exploring its environment, the agent can then *exploit* its knowledge and select the actions that maximize (its estimate of) Q^*. These methods, known as *model-based*, use information from the successive interactions with the environment to build a

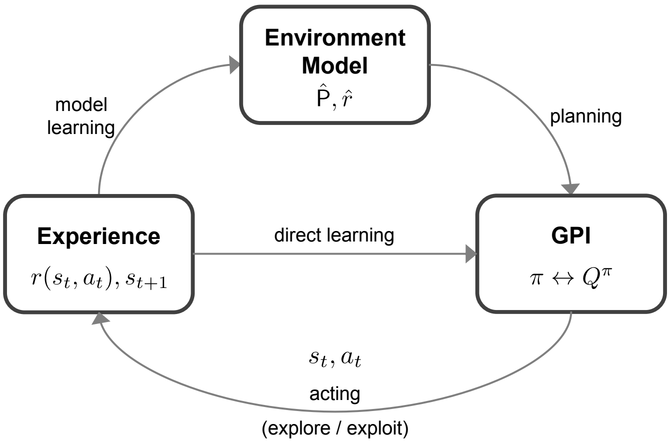


Figure 2.4: Model-based learning scheme, where acting, learning and planning interact. Adapted from (Sutton and Barto, 1998).

model of the world and then use planning to simulate experience and update the value functions, as depicted in Figure 2.4 (Sutton and Barto, 1998).

Many RL methods also come with guarantees of asymptotic convergence to the optimal policy, as long as some conditions are met (*e.g.*, that the agent visits all state-action pairs an infinite number of times) (Kaelbling et al., 1996; Sutton and Barto, 1998).

Applying *learning* to our Moving Preys Scenario example means that we want our predator to explore its environment, trying each action in each perceived state a certain number of times. In a sense, this will eventually enable our agent to learn a "mental map" of the environment that allows it to go from point (x_1, y_1) to point (x_2, y_2) by performing a specific sequence of movement actions. It will also be able to learn that eating hares when collocated with them is somehow "good". In this manner, learning the optimal policy for this scenario can be seen as "knowing" the direct path (sequence of movement actions) from the agent's current location to the prey at each time-step. Furthermore, provided that the agent has access to the current position of the prey P_{prey} at each time-step t, we can use *e.g.*, prioritized sweeping, to determine the optimal policy for this problem.

2.3 Design Challenges in RL

The RL framework presented earlier allows for the development of elegant solutions that are guaranteed to converge to the optimal behavior. However, despite all of its success, RL algorithms present several design challenges and such guarantees come with several requirements that the problem must meet (Kaelbling et al., 1996; Littman, 1994; Singh et al., 1994; Sutton and Barto, 1998). Figure 2.5 delineates the diagram of design challenges within the frameworks of RL and IMRL that we discuss throughout this section.

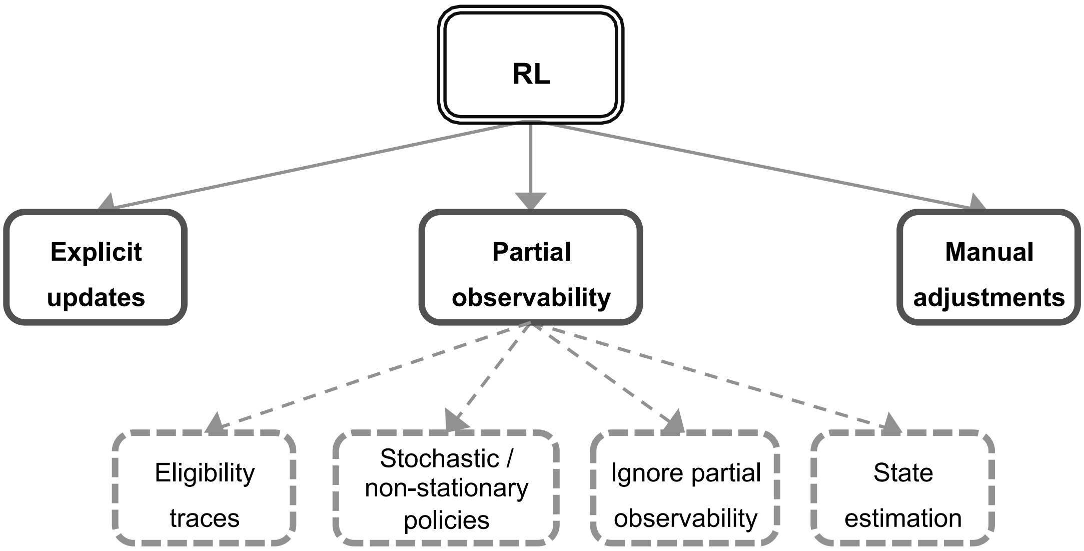


Figure 2.5: Diagram containing some design challenges within the framework of RL discussed in the context of this thesis. Closed boxes represent design challenges and dashed boxes represent common solutions for them.

2.3.1 Explicit Updates

Several RL algorithms come with a guarantee of convergence to the optimal behavior. However, some conditions must be satisfied, namely the agent has to explicitly update every state-action pair an *infinite* number of times (Kaelbling et al., 1996). In practical cases, we expect algorithms to find a good policy within a finite amount of updates. Still, "sufficient" exploration can quickly become unfeasible considering large environments, thus following this approach is potentially tedious and can take a great amount of time before achieving the desired behavior[4].

However, a computational technique known as *reward shaping*[5] directly adjusts the reward mechanism incrementally in order to scale up RL methods to handle complex problems (Dorigo and Colombetti, 1994; Ng et al., 1999; Randløv and Alstrøm, 1998). Other approaches reward local behaviors that approximate the final goal (Mataric, 1994) or provide the agent with simpler versions of the problem in order to learn the behavior of interest (Selfridge and Sutton, 1985). All these methods seek to speed-up learning by progressively approximating the intended behavior. For simple and sequential tasks, we can somehow adjust the behavior of the agent by providing "extra" relevant rewards that direct its actions towards the desired objective.

For example, in our Moving Preys Scenario, there are several states, one for each cell in the environment, that are somehow irrelevant for the completion of the intended task, *i.e.*, what it matters

[4] We refer to (Kakade and Langford, 2002; Thrun, 1992) for examples and discussion about the performance of exploratory techniques for these classes of problems.

[5] This computational procedure is inspired by a training technique pioneered by Skinner (1938) that allows to train an animal to perform a complex task by selectively reinforcing successive approximations towards the intended behavior (Anderson, 2000), and was an idea central to the theory of operant conditioning (Anderson, 2000; Skinner, 1938).

is that the agent is able to go to one of the three end-of-corridor positions of the environment. To speed-up learning, we could therefore provide the agent with an extra reward according to its position in the environment. The closer the predator is to its prey the higher the reward. In this manner, the location of the predator itself is informative for the task accomplishment.

Another important implication of explicit updates is that many classical methods require tabular representations[6] for the target functions, which is impractical in large scenarios. Therefore, there is also the need for *generalization*: in real settings, the states experienced will not be exactly the same, so the problem is to use the experience from a limited part of the state space and *generalize* it to unvisited states (Kaelbling et al., 1996; Sutton and Barto, 1998).

A problem related with explicit updates in large state spaces has to do with the specific characteristics of the space itself, which may lead to inefficient use of experience. Some environments present a state space where the information which is relevant for the *task* to be learned is concentrated in a small set of states. This means that the majority of the observations experienced by the agent during learning are *irrelevant* for it to solve the task, and require a great deal of useless experimentation (Jong and Stone, 2005; Kroon and Whiteson, 2009). The way to behave effectively in a wide range of environments is to use memory about previous interactions (states and actions) to disambiguate new states, thus reducing the amount of experimentation required (Kaelbling et al., 1996; Sequeira et al., 2013).

The most common methods to solve this kind of problems is to use supervised learning techniques and estimate the value function through function approximation, using more compact parameterized representations (see Szepesvári, 2010, for references and discussion). Another possible direction is to structure the problem in smaller problems that can be solved locally, thus reducing the search space (Sutton and Barto, 1998).

2.3.2 Partial Observability

Another problem about the original theory of RL is that it assumes that the agent learns in a Markov environment, *i.e.*, most of the techniques originally proposed to solve RL problems are limited to be used in the context of MDPs (Kaelbling et al., 1996; Littman, 1994; Loch and Singh, 1998; Singh et al., 1994; Sutton and Barto, 1998). In fact, RL methods usually assume two restrictions regarding the relationship between the agent and its environment (Kaelbling et al., 1996):

1. The first one is *full observability* over a Markov environment which allows us to predict future observations and rewards (Sutton and Barto, 1998);

[6] *Tabular methods* build up approximations of value functions, policies and models using arrays or tables containing entries for each state or state-action pair (Sutton and Barto, 1998).

2. The second is that the transition function P remains fixed or slowly changes throughout time, *i.e.*, that the environment is stationary or slowly-varying non-stationary.

In practice, this requires that the agent is provided with a complete description of the state of the environment (in terms of the *relevant* information to make predictions), and that such predictions are dependent only on the current state and are independent of past states or other variables (such as time) (Singh et al., 1994).

Perceptual Aliasing

In realistic settings (*e.g.*, the real-world), however, the agent often does not have a complete perception of the state of the environment, which can cause the agent to suffer from *perceptual aliasing*, *i.e.*, mapping the same observation to different states of the environment (Kaelbling et al., 1996; Loch and Singh, 1998; Singh et al., 1994). Perceptual aliasing may be caused by perceptual and motor limitations (Bratman et al., 2012; Sorg et al., 2010a): learning agents often do not have access to all state information necessary to *decide*, besides not knowing the environment's dynamics or the exact consequences of their actions. This is particularly bad as the agent may not gather sufficient information from the environment to take the optimal decision in a given situation, which can ultimately lead to arbitrarily large losses in performance (Singh et al., 1994). Another common cause for this problem is the presence of noise in sensor data, providing *incomplete* information about the current state (Kaelbling et al., 1996). Due to partial observability the Markov property no longer holds *from the agent's perspective*, and as such in many real-world applications the conditions supporting the appealing theoretical properties of RL methods are often violated (Littman, 1994).

Let us recall the problem posed by the Moving Preys Scenario under partial observability described earlier in Section 2.2.2, *i.e.*, where the agent cannot observe the prey's position P_{prey} unless collocated with it. Figure 2.6 represents an abstraction of the state transition model for this problem. Each prey State indicates any state where the agent is hungry and the prey is located in the corresponding end-of-corridor location, *i.e.*, top, middle or bottom. Each full State corresponds to the state in which the agent is full after eating the corresponding prey. Each go to action is an abstraction that corresponds to the sequence of movement actions necessary for the agent to move from its current position to the respective end-of-corridor location. The dashed circle represents the prey unknown State that in fact the agent observes, since it cannot observe the prey's location.

Under such conditions, each observation made when the agent is hungry can therefore correspond to one of 3 different underlying states, according to the particular location of the prey P_{prey}. Applying classical RL algorithms under such conditions may prevent the agent from learning the optimal behavior as it cannot determine whether going into a specific end-of-corridor position will potentially lead to a prey, which is the only source of reward in the environment. In this case we

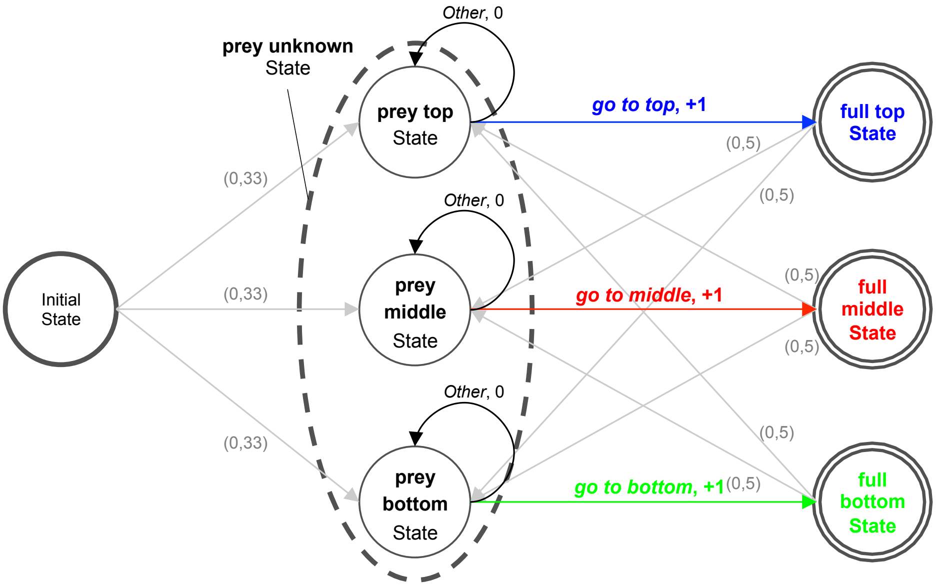


Figure 2.6: Abstraction of the state transition model for the Moving Preys Scenario. The dashed circle represents the prey unknown State that the agent in fact observes. Numbered arrows represent state transitions and the associated transition probabilities. Labeled arrows represent the effects of action execution in terms of state transition and the reward received. See text for more details.

would have to provide the agent with additional data. For example, we could use knowledge about the domain and take advantage of the fact that a prey randomly appears on an end-of-corridor location different from the one where it last appeared to "tell" the agent to explore one of the other possible prey locations.

Ignoring Partial Observability

One possible strategy to deal with partial observability is to just ignore it, treating the observations as if they were states, and then applying a traditional RL algorithm to solve the POMDP as if it was an MDP (Kaelbling et al., 1996). The first consequence of doing so is that common algorithms such as Q-learning (Watkins, 1989), although handling well partial observability in some scenarios, are not guaranteed to converge to an optimal policy (Kaelbling et al., 1996; Littman, 1994). Even employing model-based approaches to collect the transition model of the environment brings no guarantees, as the transition probabilities estimated depend on the policy being used (Kaelbling et al., 1996; Singh et al., 1994). Moreover, Littman (1994) showed that finding the optimal deterministic policy in such conditions is NP-hard (more on this ahead).

From an algorithmic perspective, it is also important to determine exactly what are the consequences of employing traditional RL techniques in POMDPs, and what kinds of strategies there are to alleviate such drawbacks. Singh et al. (1994) showed that just confounding two states can lead to an arbitrarily high loss in the cumulative payoff received. Also, this loss is not proportional to the degree of "non-Markovianness" of the environment, *i.e.*, one cannot assume that because an environment suffers from "mild" partial observability (*e.g.*, only one of the state's variables is not observable) the loss in return will also be small.

Stochastic and Non-stationary Policies

As mentioned earlier, in MDPs it is possible for an RL algorithm to find an optimal stationary deterministic policy. However, in POMDPs, if the agent treats observations as states, the best stationary deterministic policy can be arbitrarily worse than the best *stationary stochastic*[7] *policy*, although the latter can be arbitrarily worse than the optimal policy in the underlying MDP (Singh et al., 1994). Additionally, optimal policies in POMDPs can be *non-stationary*. This has to do with the fact that, due to hidden state, in POMDPs there may not be a policy that maximizes the value of each observation simultaneously (Singh et al., 1994).

All these facts open the possibility for a new class of policies which can perform *well* in POMDPs. These strategies try to attenuate the fact of not being able to observe the complete state of the environment by using policies that change over time—*non-stationary policies*—or choose actions according to some probability—*stochastic policies*. For example, in our Moving Preys Scenario and by looking at the state transitions in Figure 2.6 we could consider policies that choose going to each of the end-of-corridor locations with a probability of 1/3. Still this would not guarantee that the agent would go to a corridor and find a prey there. Furthermore, the task of determining optimal non-stationary policies or stationary stochastic policies is generally hard. The policy space is infinite, thus searching for the best strategy can be very computationally demanding. In fact, Littman (1994) has proved that finding the optimal policy in such conditions is NP-hard.

Other Approaches

Alternative approaches to learning in POMDPs apply state-estimation, building internal representations of the state of the environment (Singh et al., 1994). However, such approaches involve strong assumptions about the environment (*e.g.*, the number of states is known). Also, the state estimation procedure may be computationally expensive by requiring large amounts of samples. Finally, if state estimation and learning are performed at the same time, learning computation is wasted until good state estimations are accurate enough (Singh et al., 1994).

Another approach presented by Loch and Singh (1998) proved that algorithms that use eligibility traces (such as SARSA (Rummery and Niranjan, 1994)), work well in POMDPs that have good memoryless or low-order memory-based policies. This is due to the fact that the eligibility traces allow that each visited observation-action pair has access to the information about reward during a certain number of following learning steps, thus *diluting* the problem of the uncertainty about the hidden state (Loch and Singh, 1998).

[7] A *stochastic policy* π is a policy that chooses an action $a = \pi(s)$ given the current state s according to some probability distribution, for all $s \in \mathcal{S}$ (Singh et al., 1994).

2.3.3 Manual Adjustments

As discussed in Section 1, practical reinforcement learning often requires providing the learning algorithms with extra *a priori* knowledge. This requires human design effort in manually tuning the algorithms for particular domains.

One of the design challenges within RL that we focus in this thesis has to do with building reward functions that allow the agent to learn the task intended by its designers (Abbeel and Ng, 2004; Ng and Russell, 2000). In complex environments even the agent's designer cannot easily determine the "correct" behavior for the agent in a particular task, making the task of handcrafting reward functions possibly unpractical (Sorg et al., 2010a).

Another case within RL requiring a great effort of adjustment has to do with the state space representation. Cases demanding special manual tuning include large or continuous state spaces, where agent designers are sometimes required to break the tasks in smaller sub-tasks or manually discretize the state space to reduce the number of dimensions (Kaelbling et al., 1996).

2.4 Intrinsically-Motivated Reinforcement Learning

In this thesis we follow a recently proposed approach for dealing with the problem of reward function design from an ecological perspective that successfully mitigates some of the limitations found in computationally bounded agents, including partial observability. This approach is defined within the framework for *intrinsically-motivated reinforcement learning* (IMRL) (Singh et al., 2009, 2010), and the problem is referred to as the *optimal reward problem* (ORP) (Singh et al., 2009; Sorg et al., 2010a).

2.4.1 Motivation

To motivate the idea behind IMRL and the ORP, we continue to analyze the problem behind the Moving Preys Scenario. So far we have considered rewards that directly relate to the the task being solved, *i.e.*, provide a positive reward when eating a prey and rewards that take into account the distance to the prey's position by means of reward shaping. As we have seen, none of these solutions are satisfactory given the agent's problem of not being able to observe the prey's current location. We have also seen that stochastic and/or non-stationary policies can perform well in environments with partial observability such as this one. Such policies may work well for this particular case, however we would like to provide the learning agents with flexible and robust mechanisms that are able to allow good performances in a wide range of environments. Moreover, as mentioned earlier, searching for stochastic or non-stationary policies is hard as there could be infinite possibilities.

2.4.2 Separating Designer and Agent Objectives

As we have seen in Section 2.2, in classical RL it is assumed that the reward function used by the agent to learn the intended task is part of the external environment, as illustrated in Figure 2.1, and that the provided rewards *directly relate* to the task being accomplished.

According to Sorg et al. (2010a), there is a misconception with classical agent design within RL that “confounds” the designer’s and the agent’s objectives by only using reward functions directly associated with the task intended by the designer. Due to the aforementioned challenges faced by both the agent and its designer when adjusting a reward mechanism for a specific task, the authors proposed a separation between the *designer*’s and the *agent*’s objectives in order to achieve flexible solutions for reward function design while mitigating agent computational bounds. The idea is that, because of their limitations, agents should follow their *own* goals while learning, and their performance should be later evaluated against the designer’s objectives.

Returning to our Moving Preys Scenario example, instead of directly providing rewards upon task completion, we can encode within the learning agent reward functions that provide rewards for actions performed some time ago, thus promoting exploration in the environment. Because the environments are sometimes very dynamic and unpredictable from the perspective of the agent, this general exploratory approach could guide the agent into choosing prey locations visited a long time ago. In this manner, the position of prey that was just consumed will less likely be chosen next, and thus our agent will have a greater probability of choosing the correct location. We note that performing less experienced actions has nothing to do with the task intended by the designer. Nonetheless, we can still measure whether the agent was “successful” by *e.g.*, counting the number of preys eaten. As we shall discuss further ahead in this chapter, these kind of exploratory rewards provide *intrinsic motivation* for the agent.

2.4.3 The IMRL Model

In order to facilitate the design of reward functions and at the same time mitigate computational limitations associated with learning agents, Singh et al. (2009, 2010) proposed a framework for *intrinsically-motivated reinforcement learning* (IRML). Figure 2.7 shows the components for the IMRL model. As depicted, IMRL makes explicit that the rewards used to guide the agent are provided by a critic within its internal environment, contrasting to the traditional RL perspective in Figure 2.1.[8] Also, from an ecological perspective, the rewards with which the agent learns in RL should be referred to as *reward signals*, since the rewards are the external elements in the

[8] It is noted that one must not confound the critic providing the reward in IMRL with the *adaptive-critic* of “actor-critic” architectures (Barto et al., 1983) residing within the RL agent that evaluates the agent’s performance in the long-term perspective, usually by defining a state or action-value function. In an ecological perspective, the adaptive critic provides a form of *secondary* or *learned reward*, contrasting with the *primary* or *innate rewards* provided by the critic inside the internal environment of the agent depicted in Figure 2.7 (Singh et al., 2010).

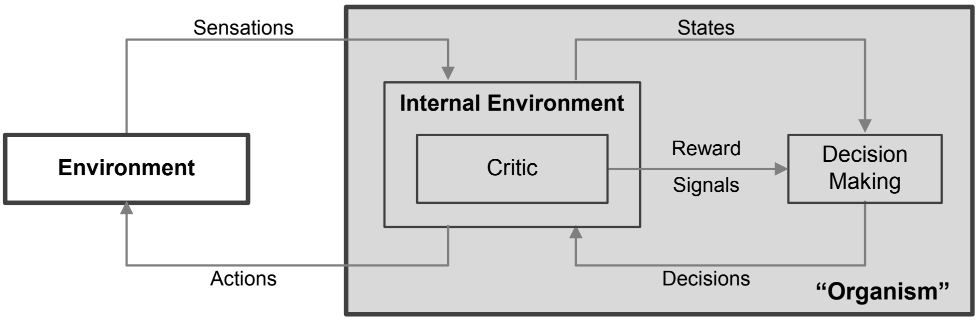


Figure 2.7: The model for intrinsically motivated reinforcement learning, where a critic belonging to the internal environment provides the reward signals to the learning agent; adapted from (Singh et al., 2010).

environment, *e.g.*, food, carrying advantageous properties that make the agent interact with them. Such external elements are perceived alongside other *sensations* to form the *states* perceived by the agent from the environment. Another distinction has to do with the *actions* performed by the agent, that are considered to occur in the external environment according to the *decisions* deliberated at each time-step by the decision-making component (Singh et al., 2010).

Formally, the IRML framework slightly extends the standard RL model presented in Section 2.2. Specifically, in this framework:

- let $\mathcal{E}$ denote some *set of environments* where we want the agent to perform well (Singh et al., 2009). Contrasting to classical RL, $\mathcal{E}$ directly relates to the problem of designing agents that perform well in a (limited) variety of environments;

- let $\{\rho_\tau, \tau = 1, \dots, t\}$ correspond to an *external evaluation signal* that, at each time-step t, depends only on the underlying state s_{t-1} of the environment and the action a_{t-1} performed by the agent. This signal can be either environment feedback—for example, when an agent receives a monetary prize for performing some action—or physiological feedback—for example, when an agent feels satisfied after feeding.

- let $h_t = \{z_0, a_0, \rho_1, \dots, \rho_t, z_t\} \in \mathcal{H}$ denote one particular *history of interaction* of the agent with some environment up to time-step t, taken from a set $\mathcal{H}$ of possible (finite) histories. Such history corresponds to all the information perceived by the agent directly from the environment: sequence $\{z_\tau, \tau = 0, \dots, t\}$ corresponds to observations about the environment state[9]; similarly, $\{a_\tau, \tau = 0, \dots, t-1\}$ corresponds to the sequence of actions performed by the agent;

- the *designer's objectives* are encoded via an *objective* or *fitness-based reward function*, denoted by $r^{\mathcal{F}} : \mathcal{S} \times \mathcal{A} \times \mathcal{H} \to \mathbb{R}$, which, if used by the agent, prescribes *preferences* over its behavior during some history of interaction (Bratman et al., 2012). $r^{\mathcal{F}}$ therefore directly rewards

[9] Here we assume partial observability by the agent, but the model is extensible to the case of full observability where $z_t = s_t$.

behaviors related to the task being learned and throughout this thesis we define it as

$$r^{\mathcal{F}}(s,a,h) = \mathbb{E}\left[\rho_{t+1} \mid s_t = s, a_t = a\right]. \tag{2.7}$$

- let $\mathcal{R}$ denote some *space of possible reward functions* containing $r^{\mathcal{F}}$ that the agent can use to learn the intended task;
- the *agent's objective* is represented via an *agent* or *primary reward function* $r : \mathcal{S} \times \mathcal{A} \times \mathcal{H} \to \mathbb{R} \in \mathcal{R}$. Separated from its designer's objectives, the agent's goal is the goal of any other RL agent, *i.e.*, to attempt to maximize the cumulative reward as defined by r, which is used to *guide* its behavior while learning;
- let $p_H(h_t \mid r, e)$ denote the probability of observing history h_t in environment $e \in \mathcal{E}$ given the reward function r.

IMRL extends classical RL by considering reward functions which map from agent histories to scalar values. The *agent designer's objective* is to build agents that perform well in a set of environments of interest (Bratman et al., 2012; Singh et al., 2009, 2010; Sorg et al., 2010a). For the agent to meet the objectives of its designer, its performance over history h_t is evaluated by some real-valued *objective* or *fitness function* $f : \mathcal{H} \to \mathbb{R}$. This function measures how the fitness-based reward is accumulated during h_t and throughout this thesis we define it as

$$f(h_t) = \sum_{\tau=1}^{t} \rho(s_\tau, a_\tau), \tag{2.8}$$

Furthermore, each reward function $r \in \mathcal{R}$ can be evaluated according to the scalar value $\mathcal{F}(r)$, which can be estimated from a set of histories of interaction, $\{h_t^1, \ldots, h_t^N\}$, as

$$\mathcal{F}(r) = \sum_{i=1}^{N} p_H(h_t^i \mid r, e^i) p_E(e^i) f(h). \tag{2.9}$$

Each history h_t^i is sampled according to $p_H(h_t^i \mid r, e^i)$, where e^i is an environment sampled according to a distribution $p_E(\mathcal{E})$ over the environments in $\mathcal{E}$ and in which the reward function r is held fixed (Singh et al., 2009).

Figure 2.8 delineates the extended learning scheme proposed by the IMRL framework. The result of each learning process i is a history of interaction h_t^i given a reward function r and an environment e^i sampled as described above. We note that, for the purpose of generating each history h_t^i, the agent is only constrained to learn with reward function r, independently of the method used to learn the optimal policy/optimal value functions therein (Singh et al., 2009). Moreover, each $r \in \mathcal{R}$ is constrained only to be any reward function mapping from agent histories to scalar values (Singh et al., 2009). The result of the overall learning procedure is the evaluation

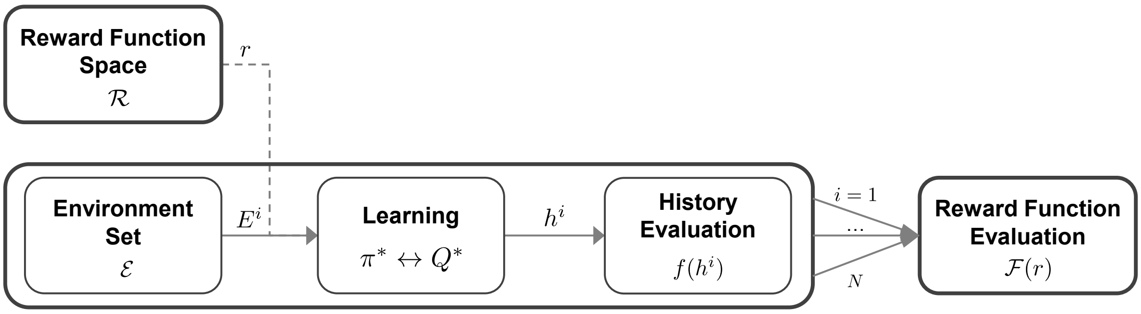


Figure 2.8: IMRL learning scheme, where the agent learns with a given reward function r and its performance is evaluated given a fitness function f and a set of environments of interest $\mathcal{E}$ (Singh et al., 2009, 2010). The *Learning* block represents any RL method that learns an optimal policy from experience given r (see text for details).

of reward function r according to the fitness function $f(h_t)$ as given by (2.9). In a sense, $f(h)$ measures how *well-adapted* the agent is to its intended purpose/environment during some history of interactions with it (Singh et al., 2010). In the same manner, $\mathcal{F}(r)$ measures the *impact* of reward function r on the agent's *fitness* in a set of environments of interest $\mathcal{E}$.

Provided a formal manner of evaluating histories of interaction with the environment, the *optimal reward problem* (ORP) is then defined as an *optimization problem*: that of choosing an *optimal reward function*, denoted by $r^* \in \mathcal{R}$, that maximizes the expected fitness or *objective return* with respect to a distribution over possible environments (Bratman et al., 2012; Singh et al., 2009, 2010; Sorg et al., 2010a), *i.e.*, a reward function such that

$$r^* = \underset{r \in \mathcal{R}}{\operatorname{argmax}}\, \mathcal{F}(r) = \underset{r \in \mathcal{R}}{\operatorname{argmax}}\, \mathbb{E}_{p_H(h_t^i|r,e^i)p_E(e^i)}\left[f(h)\right]. \tag{2.10}$$

Table 2.1 summarizes the idea behind IMRL and the ORP formulation through the separation of the agent and its designer's objectives. It also presents the nomenclature and notation used within IMRL/ORP in comparison to classical RL.

2.4.4 Mitigating Computational Bounds

One advantage of using the ORP formulation is that it often has the potential to alleviate limitations in computationally bounded agents. Sorg et al. (2010a) designed a series of experiments targeted at assessing how well-designed rewards can mitigate some limitations of computationally-bounded learning agents. These experiments showed that by means of some internal reward features, well-designed reward functions can enhance the performance (as measured by the designer's fitness utility function) of agents that have limitations in planning depth, impoverished state representations, locally inaccurate models, and also some limitations inherent to function approximation and specific learning algorithms used (Sorg et al., 2010a).

Table 2.1: Separation between the agent's and its designer's objectives (Bratman et al., 2012; Singh et al., 2009, 2010; Sorg et al., 2010a).

	Designer	**Agent**
Objective	design agents that maximize fitness / objective utility	maximize utility as provided by $r(s, a, h)$
Reward function	$r^{\mathcal{F}}$	r
Designation	*fitness-based* / *objective*	*primary* / *agent*
Purpose	*evaluate* the agent's behavior	*guide* the agent through learning
Reward type	extrinsic or fitness-related / task-related	intrinsic or fitness-inducing
Example for the Moving Preys Scenario	+1 for eating a prey	+1 for going to a prey position visited a long time ago
Utility function	$f(h)$	*e.g.*, the value in (2.1)
Designation	*fitness* / *objective* utility	*guidance* / *agent* utility
Example for the Moving Preys Scenario	number of preys eaten	—

Bratman et al. (2012) further proposed a formal manner to assess the benefits for the design of learning agents when solving the ORP. The authors propose a distinction between *strong* and *weak mitigation*. *Weak mitigation* has to do with finding rewards other than the designer's implicit fitness-based reward function that enhance the agent's performance, regardless of the method that is used for that purpose or the agent's limitations. *Strong mitigation* is a setting in which both the cost of searching for good reward functions and the cost of evaluating those functions is taken into account, more specifically by splitting the computational resources available to the designer in order to achieve these two tasks.

Figure 2.9 presents the *nested optimal reward and control* (NORC) architecture for strong mitigation proposed in (Bratman et al., 2012). This model extends the IMRL standard model depicted in Figure 2.7 by searching for good reward functions online, *i.e.*, while the agent is learning the intended task. It defines two RL agents learning and acting interchangeably: the *critic agent* evaluates and adjusts the reward function used by the IMRL *internal critic*; the *decision-making agent* is the normal agent learning the best policy at each time given that reward function. The critic agent gathers statistics about the designer's fitness-based reward received during a certain period of time (*e.g.*, by using $\mathcal{F}(r)$) and uses such information to improve the intrinsic reward function being used. In this manner it is possible to split the available time to both learn the optimal policy and improve the reward function online, thus achieving strong mitigation.

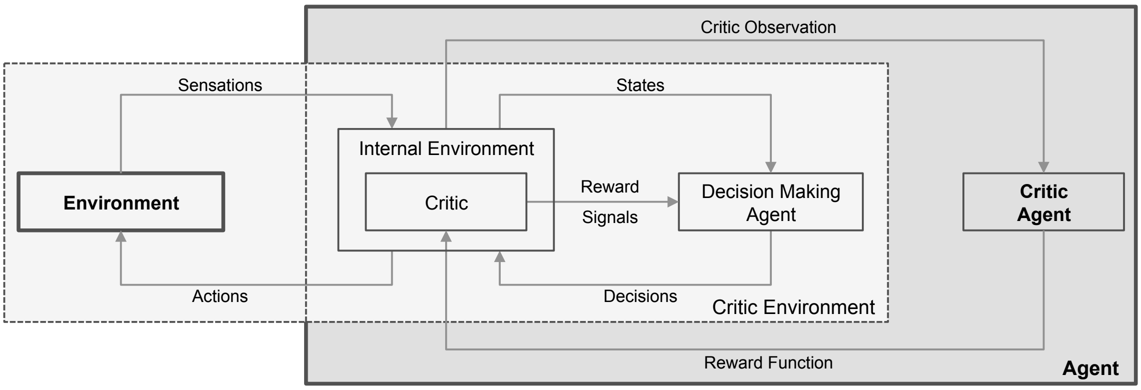


Figure 2.9: The model for strong mitigation provided by the NORC architecture, where a "critic agent" adjusts the reward function used by the internal critic to provide the reward signals during learning; adapted from (Bratman et al., 2012).

2.5 Design Challenges in IMRL

To better understand the implications and the computational focus of our approach we summarize the challenges involved in designing RL agents that were discussed throughout this chapter:

1. First of all, reinforcement learning involves the design of a *learning algorithm* that given some task—including a *state-space representation* and a *reward function*—is able to compute an optimal policy for it;

2. If the reward function and/or the state transitions are unknown we can delineate an *exploration strategy* to gather useful information from the interactions with the environment and thus estimate the desired optimal policy;

3. Research within the framework of IMRL has shown that using *intrinsic reward functions* relating to aspects of the agent's history of interaction with the environment that may not be directly related to the task being learned can actually lead to good performances by the agent, particularly in the presence of perceptual limitations—which is termed *weak mitigation*;

4. By considering further computational bounds of the agents *e.g.*, limited planning depth, we can design *optimization mechanisms* that improve the intrinsic reward functions online, *i.e.*, while the agent is interacting with the environment—which is referred to as *strong mitigation*.

So far we have provided a formal description of the ORP and the IMRL framework but did not explain why or how this approach will help us to alleviate some of the aforementioned challenges. As we have seen, because of their limitations, agents should *follow their own distinct goals* while learning and their performance should be later *evaluated* against the designer's objectives. The key element of this approach lies precisely in discovering an *optimal reward function* as described earlier (Singh et al., 2009; Sorg et al., 2010a).

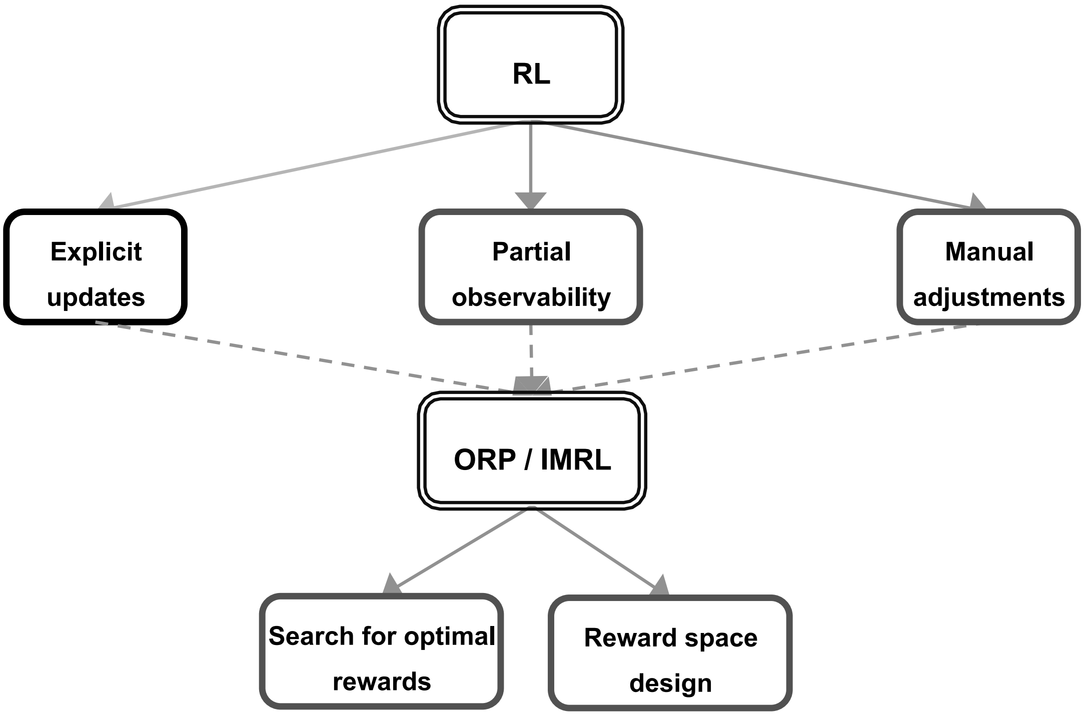


Figure 2.10: Diagram of the RL design challenges addressed by the framework of IMRL, and some challenges deriving from this approach. As before, closed boxes represent design challenges and dashed boxes represent common solutions for them.

The IMRL framework proposes an elegant solution by considering rewards not directly related to the agent's task while mitigating some of its limitations, as indicated in Figure 2.10. However, it also poses a set of design challenges or choices that one must deal with when building such reward mechanisms. Namely, one must decide how the optimization procedure will *discover the optimal reward function* and also address the *nature of the rewards* provided to guide the agent throughout learning. We summarize both challenges and also solutions proposed so far to tackle them.

2.5.1 The Search for Optimal Rewards

The ORP formulation presented above outlines two distinct search problems, namely that of *finding the optimal reward function* and that of *discovering optimal policies/value functions* (Bratman et al., 2012; Niekum et al., 2010; Singh et al., 2010; Sorg et al., 2010b). The latter can be achieved by employing traditional RL techniques given a particular reward function and a set of environments of interest, as discussed in Section 2.2.4.

Linearly Parameterized ORP

Solutions for finding the optimal reward function commonly propose a space of reward functions $\mathcal{R}$ as the linear span of some set Φ of real-valued *reward features*, $\Phi = \{\phi_1, \ldots, \phi_p\}$ (Singh et al., 2009, 2010; Sorg et al., 2010a,b), *i.e.*,

$$\mathcal{R} = \{\boldsymbol{\phi}^\top \boldsymbol{\theta} \mid \boldsymbol{\theta} \in \mathbb{R}^p\}.$$

In other words, each reward $r \in \mathcal{R}$ is a linear combination of the features in Φ,

$$r(s, a, h) = \sum_{k=1}^{p} \phi_k(s, a, h)\theta_k = \boldsymbol{\phi}^\top(s, a, h)\boldsymbol{\theta}, \tag{2.11}$$

where the $\theta_k, k = 1, \ldots, p$, correspond to the parameters of the linear combination (Singh et al., 2010). We henceforth write $r(\boldsymbol{\theta})$ to explicitly denote the reward function corresponding to the *parameter vector* $\boldsymbol{\theta}$, *i.e.*, $r(\boldsymbol{\theta}) = \boldsymbol{\phi}^\top \boldsymbol{\theta}$. We refer to this formulation of the ORP as *linearly parameterized approach to ORP*.

With this formulation, the task of the optimization procedure is to find the optimal combination of parameters denoted by $\boldsymbol{\theta}^*$. In other words, the ORP reduces to finding the parameter vector $\boldsymbol{\theta}^*$ such that the corresponding reward function, $r(\boldsymbol{\theta}^*)$, has maximal fitness, *i.e.*,

$$\boldsymbol{\theta}^* = \operatorname*{argmax}_{\boldsymbol{\theta} \in \mathbb{R}^p} \mathcal{F}(r(\boldsymbol{\theta})), \tag{2.12}$$

where $\mathcal{F}(r(\boldsymbol{\theta}))$ is given by (2.9).

The above optimization can be conducted using different techniques. Some approaches, more concerned with discussing the ORP itself, used quasi-exhaustive methods to search for good parameter configurations (Singh et al., 2009, 2010; Sorg et al., 2010a). The optimization can also be performed online, *i.e.*, while the agent is learning in a particular environment. The *policy gradient for reward design* (PGRD) algorithm (Sorg et al., 2010b) proposes a gradient ascent approach to solve the ORP by improving the agent's reward function (and the corresponding parameter vector) online. This method has been used together the NORC architecture described earlier to have into account computational limitations of the agent by adjusting the parameters while the agent is learning (Bratman et al., 2012).

Other Approaches

Apart from the aforementioned linearly parameterized approach to ORP, Niekum et al. (2010) propose the use of *genetic programming* to search for good reward functions, whereby populations of reward functions (as programs) are evolved, the fitness of which is calculated according to the fitness function[10].

An approach similar to that of the NORC architecture is presented in (Hester et al., 2013) in which an algorithm learns to choose online from several predefined exploration strategies using bandit-type algorithms. The exploration strategies rely on combinations of intrinsic rewards col-

[10] We refer to Chapter 5 for a similar approach to search for domain-independent reward functions.

lecting information about the structure and quality of the model built by the agent. Such method is useful when learning across several domains to reduce the amount of fine-tuning needed to adjust exploration strategies and rewards for a specific task.

2.5.2 The Nature of the Reward Functions

The IMRL framework proposes the use of richer reward functions that implicitly encode information to potentially overcome the agents' perceptual limitations. And, in fact, this approach was shown useful both to facilitate reward design (Niekum et al., 2010; Sequeira et al., 2011a) and to mitigate agent limitations (Bratman et al., 2012; Sorg et al., 2010a,b). However, a design challenge within IRML has to do with the *nature* of the agent reward functions $r \in \mathcal{R}$ that serve as parameters to the optimization problem described earlier.

Computationally, in IMRL, this problem relating the possible sources of reward has been addressed by providing the agent with both domain-dependent and domain-independent reward features fostering exploration and manipulation behaviors (Singh et al., 2010). Domain-dependent features use (combinations of) the values of immediate state/observation components to generate reward, for example using the location of the agent at some time-step (Niekum et al., 2010), or combinations of “thirst” and “hunger” status of the agent (Niekum et al., 2010; Singh et al., 2009, 2010). Domain-independent features relate to statistical properties of the history of interaction with the environment.

For example, some solutions have been based on (inverse) recency features relating the number of times the agent has visited some state-action pair or (inverse) frequency features measuring the time since the agent last visited some state, all rewarding exploratory or “play” behaviors (Bratman et al., 2012; Singh et al., 2010; Sorg et al., 2010a). From a perspective of measuring curiosity and creativity in learning agents, Schmidhuber (2010) proposed a reward feature that measures improvements in the agent's world-model to provide intrinsic reward. Other solutions include using the Manhattan distance to some goal location (Bratman et al., 2012), rewarding behaviors that get the agent closer to its objective, or using recent transition errors to measure the quality of the agent's model (Sorg et al., 2010a), rewarding situations in which the agent has more control over its environment.

2.6 Research Problem

As mentioned in Chapter 1, in this thesis we are interested in designing reward functions that augment the *autonomy*, *robustness* and *flexibility* of autonomous learning agents. This means that our research focus is in discovering reward functions that are both capable of alleviating perceptual limitations inherent to learning agents and that are generic enough to be used in a variety of distinct scenarios thus reducing the amount of manual fine-tuning or expert knowledge on a specific domain.

In other words, our "computational focus" resides in the challenge regarding the nature of the reward functions described in the previous section. This implies that computationally we aim at weak mitigation as defined in (Bratman et al., 2012).

However, a couple of considerations must be taken about the difficulties involved in designing reward mechanisms having the above-mentioned attractive properties:

- First of all, practical reinforcement learning often requires providing the learning algorithms with *relevant a priori* knowledge, adequately tuning the algorithm for the situation of interest, thus decreasing the autonomy in the agent's decision process (Kaelbling et al., 1996; Littman, 1994);

- In complex environments, the demand for "sufficient" built-in knowledge is particularly critical as sometimes it is impossible for the agent's designer to know beforehand what the exact optimal behavior is in *some* situation, let alone design reward mechanisms that are flexible enough to be applied in a variety of different situations (Sorg et al., 2010a);

- In the context of computational bounds mitigation the word "weak" might be misleading—the challenge of designing domain-independent reward functions that are able to alleviate some of the problems associated with learning under partial observability must at least be as important as discovering the optimal one. In fact, the agent relies in these reward functions to learn the desired task and as such without "good" reward functions or "good" rewards, strong mitigation can never occur;

- Furthermore, strong mitigation implies a natural trade-off between the computational limitations of the agent and the *quality* of the obtained solutions (Bratman et al., 2012);

- On the other hand, we also have to balance the time costs involved in optimizing a generic and domain-independent reward function offline versus the time spent in manually adjusting the learning algorithm's parameters and reward functions for a specific domain.

Generally speaking, throughout this thesis we acknowledge the importance of strong mitigation, *i.e.*, we take into consideration the time necessary to discover the optimal reward functions and possible computational limitations associated with practical learning settings when testing our approach. Nevertheless, we note that our focus is mainly on the design of reward functions that are *independent* of the learning algorithm, state representation or exploration strategy adopted. Therefore, the quality and the specific characteristics of the optimized reward functions are of more importance in the context of our discussions than the processes by which they are optimized. Furthermore, there is nothing that prevents our proposal for intrinsic rewards from being integrated within the approaches to strong mitigation (*e.g.*, within the NORC architecture of Bratman et al., 2012) discussed in Section 2.4.4 if further limitations have to be taken into account.

2.7 Our Approach

To tackle with the research problem described above, in this thesis we focus on solutions for reward design that are in line with the ecological and evolutionary principles behind the IMRL framework. As such, it is necessary to better understand the meaning of *intrinsic motivation* within IMRL, and how can it be used to address the aforementioned challenge.

2.7.1 Ecological Perspective

To understand the role of intrinsic motivation within IMRL, first it is important to address the relationship between reward functions in RL in general and the kinds of rewards in nature that motivate the behavior of biological organisms. As described throughout this chapter, RL agents' behavior is contingent on the reward function that it uses to learn the intended task/optimal policy. From a computational perspective, the *critic* evaluating the agent's performance during its lifespan provides the only source of motivation for its behavior (see Figure 2.1). IMRL approaches reward design by formulating the ORP from an *ecological* and *evolutionary perspective*(Singh et al., 2009, 2010). Within this perspective, any rewards in RL provided to the agent are considered as *primary rewards*, *i.e.*, innate, phylogenetically significant rewards that, throughout evolution, provided reproductive benefit in a particular habitat (Singh et al., 2009).

In relation to the *nature* of the rewards and motivation provided to the agent, two types are considered:

- *extrinsic rewards* characterize activities related to attaining some specific goal, like reducing the value of some biological drive[11](Oudeyer et al., 2007; Ryan and Deci, 2000; Singh et al., 2010). Making a parallel with our running example, this corresponds to the direct rewards provided by eating a prey;

- in contrast, by means of intrinsic motivation, *intrinsic rewards* foster behaviors devoid of a specific goal, such as play or exploration, that despite not having a direct biological significance for the organism are inherently enjoyable and provide adaptive advantages (Deci and Ryan, 1985; Oudeyer et al., 2007; Ryan and Deci, 2000; Schmidhuber, 2010; Singh et al., 2010).

Several parallels have been drawn between intrinsic motivation systems in the psychology literature and some active learning and experimental design techniques from machine learning (Kaplan and Oudeyer, 2007; Oudeyer et al., 2007; Schmidhuber, 2010). Also, several intrinsic motivation systems have been proposed for artificial systems in areas such as developmental robotics. Examples include the hierarchical acquisition of skills using intrinsically motivated reinforcement

[11] In works that consider the nature of the motivation provided by the rewards, the fitness-based reward function $r^{\mathcal{F}}$ is also referred to as the *extrinsic reward function* as it explicitly rewards goal-related behaviors (Singh et al., 2009, 2010).

learning (Barto and Şimşek, 2005; Schembri et al., 2007; Stout et al., 2005), the acquisition of a visual-attention system from motivation variables (Kaplan and Oudeyer, 2003), and others (Şimşek and Barto, 2006; Konidaris and Barto, 2006; Oudeyer et al., 2007; Schmidhuber, 2010).

Intrinsic rewards come to be by means of evolutionary processes that reward behaviors which have an adaptive purpose and enhance the organism's fitness in some environments. Similarly, within IMRL, intrinsic rewards can stem from an optimization procedure that can be "paralleled" with the evolutionary pressures for adaptation that animals are subject to in nature (Singh et al., 2010). Intrinsic reward functions should then provide intrinsic motivation to the agent, *i.e.*, useful rewards that may not be directly related to fitness improvement but that guide the agent throughout learning and ultimately improve its adaptive potential in a set of environments of interest (Singh et al., 2010). As illustrated in Figure 2.10, the use of intrinsic rewards within IMRL has facilitated both the design of reward functions (Niekum et al., 2010; Sequeira et al., 2011a) and also mitigated computational limitation of learning agents (Bratman et al., 2012; Sorg et al., 2010a,b).

2.7.2 Sources of Intrinsic Motivation

We have seen that intrinsic rewards are useful for the design of more flexible and robust learning agents within IMRL. But what are the sources of intrinsic motivation in nature? Oudeyer et al. (2007) analyzed some activities providing intrinsic motivation in humans according to research in the psychology field. Basically, several theories have tried to explain the existence of motivation that encourages *exploration* and *manipulation* in the environment. From a physiological point of view, such motivations are not explained by theories of homeostasis, *i.e.*, these observed behavioral tendencies do not address any specific tissue deficit like hunger or thirst do. Instead, theories of cognitive dissonance assert that organisms are motivated to *reduce the incompatibility* between perceived situations and cognitive structures built from past experience. However, such theories can not account for the "search for uncertainty" observed in human behavior. More recent research has formulated theories where both exploratory and *incongruence reduction* behaviors are rewarded. According to these theories, people seem to find an equilibrium between the search for *novel stimuli* through exploration and the comfort of familiar situations that provide an idea of *control* or *competence* over the external environment (Ryan and Deci, 2000).

2.7.3 Socio-Emotional Reward Design

Provided this ecological perspective over the possible sources of intrinsic motivation, our approach for the design of intrinsic reward mechanisms is inspired by the role of emotions in nature in motivating behavior and aiding decision-making and also by the way humans and other animals take advantage of their social context in order to thrive. Throughout the following chapters we show that reward features based on appraisal theories of emotion indeed simplify the design of IMRL agents and also alleviate common perceptual challenges. Moreover, we show that emotions emerge as natural sources of information within IMRL agents and that the motivation provided by means of the emotion-like rewards fits the above-mentioned notion of intrinsic motivation. Finally, we propose social intrinsic reward features that evaluate the agent's actions in benefiting its social group, thus extending IMRL into multiagent scenarios.

2.8 Summary

In this chapter we provided the technical background on RL and IMRL, necessary to set up nomenclature and notation used throughout the thesis. We started by presenting the general problem faced by an RL agent and also common methods proposed in the literature for the agent to learn the task intended by its designer. We overviewed some limitations and design challenges associated with classical approaches within RL. We then introduced the framework of IMRL and formulated the ORP that provide a structured way to alleviate some of the previously identified limitations. Finally, we briefly surveyed related work within these areas and identified how our approach can also contribute to mitigate challenges faced by both agents and their designers in complex domains.

The next chapter provides theoretical background and related work on emotions and the field of affective computing, making the bridge between the technical problem identified above and our approach and contributions in the following chapters.

CHAPTER 3

Emotions and Affective Computing

A large part of the contributions of this thesis is based on the adaptation of emotional signals to design reward functions for IMRL. As such, in this chapter we provide theoretical background on emotions to understand their role on biological organisms and how can they contribute to build more robust and flexible IMRL agents, alleviating some of the problems identified in the previous chapter. The area of *affective computing* (AC) has long been a research field concerned at building more interactive, natural and proficient artificial agents based on computational models of emotions (Picard, 2000). Therefore, we also show the beneficial impact that emotion-based design within AC has had in the performance of artificial learning agents.

3.1 Introduction

Over the past 50 years, research within the fields of psychology, biology, ethology, neuroscience and others have shed light over the origins, causes and consequences and the underlying brain mechanisms behind one of the most common behavioral phenomenon observed in nature: *emotions*. Emotions have often been considered as detrimental to rational and sound decision-making (Isen, 2008; Oatley and Jenkins, 2006). However, as the research about the influence of emotions on human and other animals grew, emotions started being regarded as a beneficial adaptive mechanism for problem solving, enhancing perception, memory, attention and other cognitive skills (Cardinal et al., 2002; Isen, 2008; Kensinger, 2004; LeDoux, 2007; Naqvi et al., 2006; Phelps and LeDoux, 2005). The need for an attention-focusing interrupting mechanism for artificial agents having the properties of emotions has been advocated for a long time (Minsky, 1986; Simon, 1967). Other recent works established the importance of including emotion-based components within the design of artificial agents (Lisetti and Gmytrasiewicz, 2002; Rumbell et al., 2011; Scheutz, 2004). However, apart from some exceptions focusing on discrete modal emotions, only a few computational systems have considered the potential of emotions as a basic, general-purpose, universal, evolutionary survival mechanism and its integrative role on learning, self-development and adaptation.

3.2 The Role of Emotions in Nature

We now review in greater depth what research in different areas indicates to be the purpose of emotions in biological organisms. More specifically, we emphasize the importance of an emotional mechanism to process perceptual information and guide an individual to adapt to its environment, a key idea of our approach. As mentioned above, as the body of knowledge about the influences of emotions on humans cognitive and behavioral systems grew, and more clues about the neural circuitry of emotional events were discovered, emotions have increasingly been regarded as a beneficial adaptive mechanism for decision-making (Cardinal et al., 2002; Isen, 2008; Naqvi et al., 2006; Phelps and LeDoux, 2005).

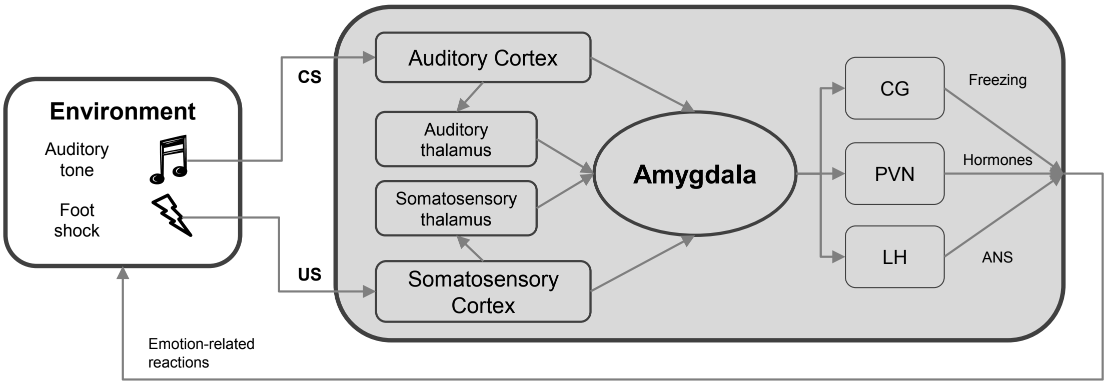


Figure 3.1: Fear conditioning processing within the amygdala (see text for details). CG—central gray, LH—lateral hypothalamus, PVN—paraventricular nucleus of the hypothalamus, ANS—autonomic nervous system. Adapted from (LeDoux, 2007).

3.2.1 The Neurological Level

At a neurological level, the limbic system is a set of evolutionarily primitive brain structures responsible for the first animal surviving skills, *e.g.*, fight or flight responses (LeDoux, 2000). Recent studies have implicated the amygdala, a region within the limbic system, has having a major role in several cognitive and behavioral processes, such as memory enhancement, sensory plasticity, attention facilitation and regulation of social behavior (Kensinger, 2004; LeDoux, 2000; Phelps and LeDoux, 2005). At the same time, the amygdala is responsible for the emotional processing of events, "imprinting" emotional content into memories, mediating learning and regulating emotional responses (Kensinger, 2004; LeDoux, 2000). At its essence, emotions thus have an essential *functional* and *adaptive* role in natural agents.

An example of the role of the amygdala in emotional learning is illustrated in Figure 3.1. In the paradigm of fear conditioning, a *conditioned stimulus* (CS) such as an auditory tone gains access to primary, species-specific behavioral responses like freezing by being associated to an *unconditioned stimulus* (US) such as a foot shock. Through *fear*, the animal increases its fitness by avoiding in the future hazardous situations like the foot shock when in the presence of learned, initially neutral stimuli (Dawkins, 2000; LeDoux, 2000, 2007).

3.2.2 The Absence of Emotions

The importance of emotions and its impact on decision-making and other cognitive skills has been demonstrated by studies that analyze the effects of lesions in (or absence of) certain regions of the brain associated with emotional processing of events both in humans and other animals. The most notorious case in that respect is that of Phineas Gage, a railroad worker which in the summer of 1848 suffered an accident while setting explosive charges in a rock which resulted in an iron bar penetrating his skull just below the left eyebrow (Damasio, 1994; Oatley and Jenkins, 2006). While Gage physically recovered from the accident, some of its cognitive capabilities were altered, turning the once efficient and reliable worker into an unstable person incapable of making correct daily-life decisions, *e.g.*, planning some work task (Damasio, 1994).

Recent technology allowed the identification of Gage's brain region damaged in the accident as being the ventromedial prefrontal cortex (vmPFC), an area responsible for assessing the relevance of events and also the modulation of emotional-eliciting behavior from the lower regions like the amygdala (Bechara et al., 2000; Oatley and Jenkins, 2006). Without this controlling mechanism, individuals are unable to undertake proper planning or advantageous decisions relating their personal well-being (Bechara et al., 2000; Damasio, 1994; Naqvi et al., 2006; Oatley and Jenkins, 2006). Such observations lead to appearance of the *somatic marker hypothesis* Bechara et al. (2000); Damasio (1994) that proposes the involvement of emotions, particularly the bodily signals (the so-called *somatic markers*) arousing with them, in proper reasoning and planning by "guiding" the subject towards advantageous choices.

Besides accidental lesions in humans, such as the one observed in Phineas Gage, lesions deliberately inflicted to the same brain regions in animals also establish the importance of emotions in providing adaptive behavior. As mentioned before, fear conditioning is an experimental procedure in which an animal learns to fear (through avoidance or freezing) a previously neutral stimulus when paired several times with a biologically significant aversive stimulus (LeDoux, 2000, 2007) (see Figure 3.1). Within this paradigm, several studies have reported that lesions to areas of the amygdala and the vmPFC lead to a general inability to learn the conditioned response that lead to the (advantageous) exhibition of fear (Amorapanth et al., 2000; Killcross et al., 1997), deficits in attentional and perceptual mechanisms (Holland and Gallagher, 1999), memory impairment (Morgan et al., 2003; Nader et al., 2000), inappropriate behavioral reactions (Morgan and LeDoux, 1995; Walker and Davis, 2002), and other unfavorable effects (LeDoux, 2000).

Together, these studies show that lesions in certain areas of the limbic system lead to poor decision making (Damasio, 1994; Naqvi et al., 2006), lack of the ability to learn the significance of stimuli through association (LeDoux, 2000, 2007), or impoverished retention of emotional stimuli in memory (LeDoux, 2000; Phelps and LeDoux, 2005). At the same time, these damages also impair "normal" emotional responses to situations and a general inability to behave in a healthy social manner, further implicating the role of emotions in cognitive processing mechanisms involving learning (Damasio, 1994; Naqvi et al., 2006; Oatley and Jenkins, 2006).

3.2.3 Emotions, Evolution and Learning

It is not only in human beings that emotions seem to play a major role in the processing of external events. In fact, emotional processing of events seems to involve primitive circuits within the limbic system that were conserved throughout mammalian evolution (LeDoux, 2000). Throughout evolution, emotions have provided animals with the ability to better obtain food resources and avoid predators and other dangers (Cardinal et al., 2002; Darwin, 1872; Dawkins, 2000). As such, emotions provide an adaptive tool for natural agents, enhancing their fitness and behavior potential in a dynamic and sometimes unpredictable environment (Cardinal et al., 2002; Dawkins, 2000; LeDoux, 2000). Emotions, like other information-processing natural mechanisms, develop at different levels:

- by means of evolutionary processes, emotions develop at a *phylogenetic*, adaptive level. Unlike simple reflexive responses, emotions provide a multitude of complex interactions that allow for more adaptive behaviors in both the social and physical environment (Darwin, 1872; Dawkins, 2000; Ellsworth and Scherer, 2003; Griffiths, 2004; Lazarus, 2001; Leventhal and Scherer, 1987; Scherer, 2001);

- there is also evidence for the non-stationarity of emotions during an organism's lifetime. At an *ontogenetic* level, biological reinforcement learning, by means of associative learning processes, seems to rely on emotional states to indicate whether a situation is perceived as advantageous or undesirable (Cardinal et al., 2002; Dawkins, 2000; Naqvi et al., 2006).

Biological reinforcement learning processes found in nature thus seem to rely on *emotional cues* to indicate the pleasantness or adversity of events and identify advantageous acting opportunities or harmful behaviors (Cardinal et al., 2002; Dawkins, 2000; Leventhal and Scherer, 1987). Moreover, as we have seen, the absence of such evaluative mechanism impairs a good judgment of the significance of situations for the individual's well-being, leading to maladaptive behaviors.

◇

As a consequence, all the aforementioned studies show that rational, fitness-focused decision-making relies on emotional states—the feelings—to indicate whether a situation is perceived as beneficial or undesirable, the lack of which leads to disruptive and maladaptive behaviors (Bechara et al., 2000; Cardinal et al., 2002; Damasio, 1994; Naqvi et al., 2006).

3.3 Appraisal Theories of Emotions

As stated earlier in the beginning of this chapter, throughout this thesis we approach reward design within IMRL by considering rewards based on emotions. We have seen that emotions is a crucial skill in natural agents to correctly adapt to their environment. However, so far we lack a formal way to describe the process of emotion elicitation. In this section we provide theoretical background about the generation of emotions according to events perceived in the environment according to appraisal theories of emotion.

Appraisal theories of emotions were pioneered by the works of Arnold (1960), who proposed that the elicitation of some emotional state is preceded by an *appraisal* of the situation in relation to its significance for the individuals's well-being or goals. Later, Lazarus (1966) introduced a

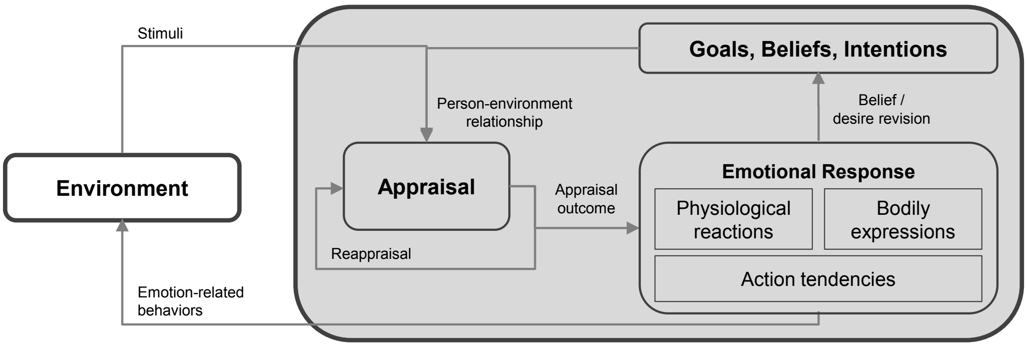


Figure 3.2: The elicitation of emotional responses as the result of an evaluation of the situation (the stimuli) in relation to the individual's goals, beliefs and intentions, from the perspective of appraisal theories of emotion.

distinction between *primary* and *secondary appraisals* to distinguish the processes of *evaluating* and *coping with* a situation, respectively (Ellsworth and Scherer, 2003; Leventhal and Scherer, 1987; Roseman and Smith, 2001; Schorr, 2001). Appraisal theories contrast with other theories of emotion elicitation which do not consider such an evaluative and relational process. Specifically, they contrast with categorical theories of emotions targeted at explaining the emergence of only a limited number of basic emotions (Ellsworth and Scherer, 2003; Frijda and Mesquita, 1998; Roseman and Smith, 2001). They also contrast with stimulus-response and other physiological and expressive theories which focus on the subjective experience of emotions while ignoring the link between the situation and the individual as proposed by appraisal theorists (Ellsworth and Scherer, 2003; Roseman and Smith, 2001).

3.3.1 The Process of Appraisal

Appraisal theories of emotions are more concerned about how the result of the appraisals affects behavior and decision-making. Figure 3.2 shows a diagram of the process of emotional elicitation by means of appraisal, which combines information from external stimuli and the individual's goals, beliefs and norms to perform an evaluation of the situation. The outcome of the appraisal is a set of responses, including the physiological signals and bodily expressions responsible for the *subjective feelings* of emotions. They are also responsible for the *functional aspect of emotions*, *i.e.*, the behavioral and cognitive responses to deal with the situation at hand which direct the individual's attention to the significant aspects of its environment (Frijda and Mesquita, 1998; Lazarus, 2001; Leventhal and Scherer, 1987; Smith and Kirby, 2000).

Many of the theories of appraisal stressed in the literature (Frijda and Mesquita, 1998; Lazarus, 2001; Reisenzein, 2009; Roseman, 2001; Scherer, 2001) propose *structural models* in which emotions are elicited by evaluations of events through a set of appraisal *variables*. Each variable is usually conceptualized as a *dimension* along which appraisal outcomes may vary continuously (Roseman and Smith, 2001). The several dimensions define the criteria used to evaluate a situation and ascribe the *structure* or the contents of the appraisal (Ellsworth and Scherer, 2003; Roseman and Smith, 2001). Moreover, through a process of *reappraisal*, the subject evaluates a situation by considering the outcomes of previous appraisals in order to cope with a significant event, usually by requiring a more thorough processing (Ellsworth and Scherer, 2003; Lazarus, 1966, 2001).

An important feature of dimensional appraisal theories is that they contrast with discrete appraisal theories of emotions which explain the situation by selecting one of a set of qualitatively distinct emotions (Ellsworth and Scherer, 2003; Frijda and Mesquita, 1998; Oatley and Jenkins, 2006; Roseman and Smith, 2001). This difference is a relevant aspect of the dimensional models because they are capable of accounting for the subtleties between the different emotional states as well as individual differences in the appraisal of events and cultural variance in terms of the specific emotion labeling (Ellsworth and Scherer, 2003; Frijda and Mesquita, 1998; Oatley and Jenkins, 2006; Roseman, 2001; Scherer, 2009).

By defining appraisal as a tight connection between the organism and the particular situations eliciting the emotions, what Lazarus (2001) calls the *relational meaning* (corresponding to the person-environment relationship in Figure 3.2), appraisal theories support the fact that different individuals will appraise external events in a distinct manner according to their own experience and cultural background (Frijda and Mesquita, 1998; Roseman and Smith, 2001; Smith and Kirby, 2009). As such, most of the appraisal dimensions proposed in the literature deal with universal, culturally-independent evaluations of the personal significance of events. It is by combining specific values or outcomes of the dimensions that these theories can model discrete emotions (*e.g.*, joy, sadness, fear, etc.) and predict the particular physiological responses and action tendencies associated with each of them (Ellsworth and Scherer, 2003; Frijda and Mesquita, 1998; Roseman and Smith, 2001).

3.3.2 Major Dimensions of Appraisal

While the several appraisal theories have some differences, *e.g.*, which emotions are supported by the model or which particular appraisals contribute to the elicitation of some emotion, most of them largely overlap in the kinds of dimensions that are necessary for the evaluation of a given situation (Ellsworth and Scherer, 2003; Frijda and Mesquita, 1998; Roseman and Smith, 2001). A study by Ellsworth and Scherer (2003) compared the most common appraisal theories and identified a set of five *major dimensions* or *groups of appraisal* proposed by most appraisal theorists (Frijda and Mesquita, 1998; Lazarus, 2001; Leventhal and Scherer, 1987; Roseman and Smith, 2001; Scherer, 2001). Table 3.1 depicts the major dimensions and the kinds of appraisal variables usually related with each of them.

Table 3.1: Major appraisal dimensions and the common appraisal variables associated with them.

Major dimensions	**Associated appraisal variables**
Familiarity	novelty, suddenness, matching
Pleasantness	intrinsic pleasantness, valence
Motivational relevance	goal/need significance, expectation and conduciveness
Coping potential	causation, control, power, adjustment
Social significance	norm/self compatibility

3.3.3 Levels of appraisal

A relevant concern when considering rewards for IMRL agents based on emotions has to do with the level at which appraisal occurs— for example, the more basic fight-or-flight kind of evaluation observed in humans and mostly in other animals when facing a dangerous situation is different from the more cognitive assessments that we make after the death of a close family member.

Some of the above-mentioned appraisal dimensions are usually defined using high-level cognitive concepts and mental representations such as one's beliefs and desires, or even more abstract concepts such as cultural norms or social standards (Ellsworth and Scherer, 2003; Lazarus, 2001; Leventhal and Scherer, 1987; Scherer, 2001). However, our learning agents only have access to rather low-level properties of their history of interaction with the environment. As such, one of the main challenges of our emotion-based reward design is to map the evaluations made by the appraisal components into low-level scalar values that can be used as intrinsic reward features.

In this thesis we follow the perspective conceived by several appraisal theorists that many appraisals, especially in the case of young children and nonhuman animals, require little cognitive processing and simple judgments of the event (Frijda and Mesquita, 1998; Leventhal and Scherer, 1987; Scherer, 2001; Smith and Kirby, 2000). For example, Leventhal and Scherer (1987) proposed a framework where the events are evaluated at different levels of processing: a *sensory-motor* level, a *schematic* or learned level and finally at a *conceptual* level. Such multilevel model of emotional appraisal allows to explain emotions as an adaptive mechanism developing from simple, reflex-like innate responses into more complex cognitive patterns throughout time (Leventhal and Scherer, 1987). Moreover, it is by interacting with the environment and learning that the organisms evaluate more complex appraisal variables, which in turn leads to more complex emotions. In our approach, by considering this multilevel perspective, the agents can leverage the low-level information they have access to and perform evaluations similar to those made by the several appraisal dimensions.

Table 3.2: The levels of processing in appraisal. Adapted from (Leventhal and Scherer, 1987).

Dimensions	**Processing Levels of Appraisal**		
	Sensory-motor	***Schematic***	***Conceptual***
Familiarity	sudden, intense stimulation	schemata matching	cause/effect, probability estimates
Pleasantness	innate preferences/aversions	learned preferences/aversions	recalled, anticipated or derived positive / negative evaluations
Motivational relevance	basic needs	acquired needs and motives	conscious goals and plans
Coping potential	available energy	body schemata	problem solving ability
Social significance	—	self/social schemata	self ideal and moral evaluation

3.4 Affective Computing

As discussed earlier, emotions were not always considered as a beneficial adaptive mechanism, essential in cognition for problem solving and decision-making. Because of this, a large volume of early research within AI did not consider emotions as a potential mechanism to be adapted into intelligent systems (Marsella et al., 2010; Picard, 2000). Nevertheless, a few early models were proposed asserting the importance of emotions as an attention-focusing, task prioritizing mechanism, crucial to any system that wants be regarded itself as *intelligent* (Minsky, 1986; Simon, 1967).

3.4.1 Computational Models of Emotion

As researchers recognized this major role of emotions in cognition, especially after the results by Damasio (1994), they started building computational models of human emotions.[1] Computational models of emotions are usually based on appraisal theories of emotions (Marsella et al., 2010), *e.g.*, (Becker-Asano and Wachsmuth, 2008; Dias and Paiva, 2005; El-Nasr et al., 2000; Gebhard, 2005; Marinier, 2008; Marsella and Gratch, 2009; Ortony et al., 1988). A generic architecture for appraisal-based models is depicted in Figure 3.3 and can be related to Figure 3.2 outlining the process of appraisal. As we have seen previously in Section 3.3.1, a central tenet of appraisal theory is the notion of a *person-environment relationship*. As such, computational models based on appraisal develop ways to represent this relationship and how it influences the appraisals— the *appraisal derivation*, specify the role of perception, memory and inference in the *appraisal variables*, and how appraisal and derived emotions affect coping responses— the *affect consequent*

[1] We refer to (Marsella et al., 2010) for a recent overview of computational models of emotions.

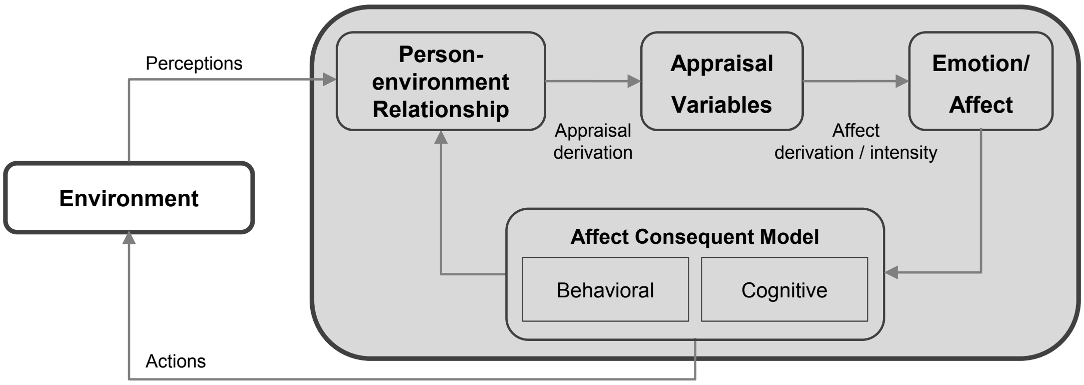


Figure 3.3: Idealized architecture for an appraisal-based computational model of emotions. Adapted from (Marsella et al., 2010).

model (Marsella and Gratch, 2009).

Over the years, computational models of emotions have enabled the construction of more intelligent and robust artificial agents (Marsella et al., 2010; Picard, 2000).[2] As a consequence, the appearance of AC allowed the inclusion of emotions in a large number of applications which allowed the construction of richer and more believable virtual characters (Dias and Paiva, 2005; El-Nasr et al., 2000; Reilly and Bates, 1992; Velásquez, 1997), more interactive systems that are able to recognize the user's feelings and express its "own" emotions (Caridakis et al., 2007; Hudlicka et al., 2009; Pantic and Bartlett, 2007), more efficient agents and robots (Ahn and Picard, 2006; Broekens et al., 2007; Cañamero, 1997; Gadanho and Hallam, 2001; Salichs and Malfaz, 2012), or even pure experimental simulations (Armony et al., 1997; Elliott, 1994; Marsella and Gratch, 2009).

3.5 Emotion-based Agent Learning

While the aforementioned use of emotion-based systems provides more engaging interactive experiences, there are only a few systems that, like our approach, explicitly leverage emotions to improve the adaptive capacity of autonomous agents. Some works bare some similarities in their use of emotion-like signals to control or influence learning. For example, in the *FLAME* model (El-Nasr et al., 2000), RL is used to form emotion-object associations and to predict the user's actions. Armony et al. (1997) used a connectionist learning approach to simulate some effects associated with the fear-conditioning paradigm. The artificial creatures developed by Cañamero (1997) are an example of the use of "low-level" emotional signals as a motivation drive to behavior selection. Based on the somatic marker hypothesis described earlier, Ventura and Pinto-Ferreira (2009) proposed an agent architecture where stimuli are processed at two different levels, thus allowing for a more cognitive assessment and response of each situation. In an approach similar to our own concerning appraisal, Si et al. (2010) propose a computational model of emotional appraisal in

[2] We refer to (Rumbell et al., 2011) for a recent comparative overview of several emotion-based systems for autonomous agents.

a multiagent framework using POMDP agents. Five key appraisal dimensions are derived to aid decision-making in a look-ahead process that calculates the next action depending on the agent's emotional state.

3.5.1 Emotion-based RL

In this section we provide a comparative analysis between some recent works within adaptive behavior research into the role of emotions in providing reward and influence decision-making within RL, which is more related to the approach proposed in this thesis.

In the field of robotics, Gadanho (1999); Gadanho and Hallam (2001) proposed a bottom-up approach to emotion elicitation. The system uses artificial neural networks combining values of sensations, feelings and hormones to determine a dominant *emotional state* from a set of four basic emotions, namely *happiness*, *sadness*, *fear* and *anger*. A traditional RL mechanism was used to reinforce state-behavior associations, where the values of the rewards and their respective sign were provided by the intensity of the current dominant emotion and whether that emotion was *positively* or *negatively* valued.

Salichs and Malfaz (2006) proposed three basic emotions to control the behavior of an agent in an RL task: *happiness*, *sadness* and *fear*. The reward provided during learning is calculated according to a temporal difference of a measure of the agent's *well-being*, which is defined by a linear combination of the levels of four internal drives. Positive and negative differences make the agent "happy" or "sad", respectively. The behavior selection mechanism uses a predefined level of *dare* to select conservative (high-valued) actions and prevent the agent from choosing bad (low-valued) actions due to fear.

The idea behind the approach of Broekens (2007); Broekens et al. (2007) is that associating *positive* affective states with *exploitation* and *negative* affect with *exploration* strategies provides adaptive benefits for the agent in some RL scenarios.[3] The agent's reward (used to learn) and its affective state at each time step are computed based on the relation between the short and long-term running averages of the reinforcement signal provided to the agent in the past. The agent's affective state further influences the behavior selection within the RL algorithm by increasing or decreasing the probability of choosing a random (exploration) or greedy (exploitation) action.

Marinier (2008) proposed an intrinsic reward signal based on appraisal theories for the SOAR architecture (Nason and Laird, 2005). The signal is based on the appraisal of *conduciveness*, which basically states how "good" or "bad" the situation is with respect to the current goal. The conduciveness appraisal's valence (positive/negative) determines the sign of the reward value, while the intensity of the agent's current *feeling* determines the magnitude of the signal. An experiment conducted in a grid-world scenario showed that intermediate, emotion-based rewards lead to

[3] Interestingly, this is in contrast with approaches relying on optimism in the face of uncertainty such as the E^3 (Kearns and Singh, 2002) and R-MAX (Brafman, 2003) algorithms.

Table 3.3: Comparative overview between our approach and other systems using emotions to influence learning and decision-making within RL.

Criteria	Gadanho and Hallam (2001)	Salichs and Malfaz (2006)	Broekens et al. (2007)	Marinier (2008)	Ahn and Picard (2006)	**Our approach**
Emotions/ appraisal variables	happiness, sadness, fear, anger	happiness, sadness, fear	valence	several	valence, arousal	novelty, goal rel., control, valence
Intrinsic/ extrinsic distinction	no	no	no	yes	yes	yes
Reward generation	emotions intensity and sign	temporal difference of internal drive levels	averages of extrinsic reward	feelings intensity x conduciv. sign	intrinsic reward + changes on value functions	linear combination of appraisal values and extrinsic reward
Emotional components contribution	fixed, equal	fixed, personality dependent	fixed	fixed	fixed	optimized weights (env. dependent)
Influence on action selection	indirect, via rewards	direct, through fear	direct, through valence	indirect, via rewards	indirect, via rewards	indirect, via rewards

learning the task faster.

Perhaps the work that most resembles our approach is the one in (Ahn and Picard, 2006). Like in our approach, the authors also consider the use of both extrinsic and intrinsic rewards to improve the learning speed and also influence decision-making. Apart from the extrinsic reward provided according to the external goal or cost, they propose a model for *affective anticipatory (intrinsic) reward* based on *valence* and *arousal* levels. Valence is related to the expected reward for some action in relation to the average reward, and arousal is calculated according to an uncertainty measure. The relative influence between immediate intrinsic rewards and long-run extrinsic rewards is given by a constant scale factor.

To better evaluate the differences between the aforementioned systems and our own approach, Table 3.3 makes a comparative overview according to some criteria. Specifically, we are interested in knowing what *emotions* or *dimensions of appraisal* are modeled in each system, the *influence* that emotions have on the agent's rewards and on action selection, the *contribution* of the different emotional components on the reward and also whether there is a *distinction* between extrinsic and intrinsic components of the rewards signal.

3.6 Summary

In this section we provided theoretical background on emotions, appraisal theories and related work involving emotion-based agent learning. We showed the importance of emotions as a basic but powerful survival mechanism for biological agents that, throughout evolution, has helped humans and other animals adapting to their environment. We showed how emotions and all the behavioral phenomena that come with them can be elicited, from the perspective of appraisal theories of emotions. We introduced the research field of AC and showed how appraisal theories have led to the development of computational models of emotions. We also performed a brief comparative analysis about some applications within RL that, just like our approach in this thesis, have used emotions to enhance the behavior of artificial learning agents.

In the next two sections we will present our approach for emotion-based design based on appraisal theories of emotions, and also analyze the role of emotions as an optimal information processing mechanism for artificial agents.

CHAPTER 4

Emotion-based Intrinsic Reward Design

In this chapter we propose a set of reward features based on four common dimensions of emotional appraisal that, similarly to what occurs in biological agents, evaluate the significance of several aspects of the agent's history of interaction with its environment. Therefore, the work included in this chapter bridges the research problem of building reward functions for IMRL, described in Section 2.6, with research on appraisal theories of emotions, described in Section 3.3. Figure 4.1 outlines our approach, validation and contributions provided in this chapter.[1]

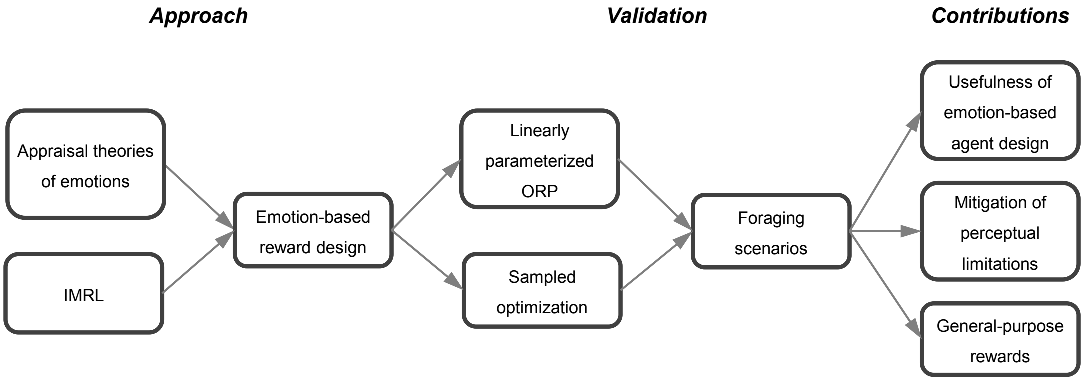

Approach
Validation
Contributions
Appraisal theories of emotions
IMRL
Emotion-based reward design
Linearly parameterized ORP
Sampled optimization
Foraging scenarios
Usefulness of emotion-based agent design
Mitigation of perceptual limitations
General-purpose rewards

Figure 4.1: Outline of our approach for emotion-based intrinsic reward design.

4.1 Introduction

In this first set of contributions we follow the notion of emotions as an evolutionary adaptive mechanism in nature and address the ORP formulation within the IMRL framework described in Section 2.4.3. We consider an intrinsic reward mechanism for autonomous learning agents inspired by appraisal theories of emotions. The main technical contribution is a set of four domain-independent emotion-based reward features, namely *novelty*, *valence*, *goal relevance* and *control*. The proposed features are based on dimensions of *appraisal of the emotional significance of events*, commonly found in the psychology literature (Ellsworth and Scherer, 2003; Frijda and Mesquita, 1998; Lazarus, 2001; Leventhal and Scherer, 1987; Reisenzein, 2009; Roseman, 2001; Roseman and Smith, 2001; Scherer, 2001, 2009; Smith and Kirby, 2000, 2009). We also focus on emotions as a plausible source of general-purpose, domain-independent intrinsic reward and discuss possible alternatives for each reward feature.

We design a set of experiments in several foraging scenarios and show that, by optimizing the relative contributions of each feature for a set of environments of interest, emotion-based reward functions enable better performances when compared to more standard goal-oriented reward functions, particularly in the presence of agent limitations. The results support our claim that reward functions inspired on biological evolutionary adaptive mechanisms (as emotions are) have the potential to provide more autonomy to the agents and a greater flexibility in the reward design,

[1] Part of the contributions within this chapter can be found in (Sequeira et al., 2011a, 2012).

while alleviating some limitations inherent to artificial agents, as depicted in Figure 4.1.

4.2 Emotion-based Reward Design

As mentioned earlier, in this chapter we focus on the agent design challenge regarding the possible sources of intrinsic rewards to be used within IRML, discussed in Section 2.5.2. We also look at possible agent limitations that can be mitigated following our approach. In this respect we propose looking at a natural adaptive mechanism present in almost all forms of biological organisms since the origins of evolution: emotions. Based on emotion literature, specifically the one presented in Section 3.3 on appraisal theories of emotions, we propose the use of reward features that, in some manner, *relate to* the way emotions provide motivation in natural organisms. The hypothesis for our approach is that *like emotions do in natural organisms, emotion-like processes that evaluate the relevance of a given situation in the environment should provide artificial learning agents with a simple but powerful adaptive mechanism.*

4.2.1 Rationale for our Approach

In Section 3.3.1 we showed that one way of explaining the generation of emotions is through the description of a process known as *appraisal*. As we have seen, appraisal theories of emotions usually define a set of appraisal dimensions through which an individual evaluates situations perceived from its environment. In our approach we follow the five *major appraisal dimensions* from (Ellsworth and Scherer, 2003; Leventhal and Scherer, 1987) discussed in Section 3.3.2 and propose four *emotion-based intrinsic reward features* based on four of those dimensions, namely *novelty*, *valence*, *goal relevance* and *control*. We restrict to single agent settings and as such do not consider the normative/social dimensions as they are responsible for the emergence of more complex emotions like shame or guilt requiring a formal social, multiagent framework.[2]

We follow the perspective that each appraisal dimension prescribes a *criterion* to evaluate the significance of a specific aspect of the individual's relationship with its environment (Ellsworth and Scherer, 2003; Scherer, 2001). Also, we explore the idea that the dimensions vary continuously, defining a multidimensional space of emotional experience in which some point represents a distinct experience (Roseman and Smith, 2001). The rationale of such proposal stems from the fact that, just like the corresponding appraisal dimension does in biological agents, each reward feature indicates the *significance* of the current situation for the agent's well-being according to specific aspects of the agent's history of interaction with its environment (Singh et al., 2009, 2010).

We note, however, that by following this approach we *are not assuming* that in nature, emotions (or the appraisal process) correspond merely to a complex form of reward—we rather inspire in the role of emotions in *motivating* behavior towards the achievement of an individual's goals and needs

[2] We refer to Chapter 6 where we propose a set of intrinsic reward features based on signals exchanged between individuals to communicate approval/reprehension of socially-aware/unaware behaviors in multiagent scenarios.

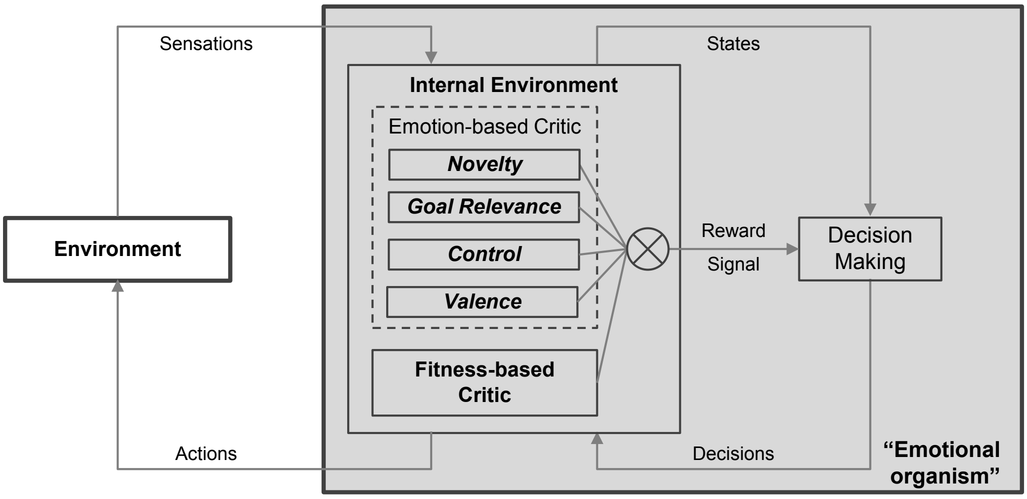


Figure 4.2: Proposed framework for emotion-based intrinsic rewards. Adapted from (Sequeira et al., 2011b; Singh et al., 2009).

or dealing with some problem at hand. We incorporate that aspect within IMRL through the design of a combined emotion-based reward signal, independently of the type of learning algorithm, state representation or exploration strategy used as discussed in Section 2.7. As will become apparent, whether the agent should pay attention to each of these emotional aspects (the features) or not will depend on the specific challenges offered by the environment.

4.2.2 Learning Model

Figure 4.2 shows our approach for an emotion-based evaluative mechanism as part of an intrinsic reward component for learning agents, extending the IMRL model illustrated in Figure 2.7. We adopt the ORP formulation within the evolutionary interpretation of IMRL described in Section 2.4.3. For the reward functions providing the intrinsic rewards that guide the agent throughout learning, we propose a linear combination of reward features originating from both an "emotional" and a "fitness-based" critic, as indicated in Figure 4.2. The *fitness-based critic* corresponds to the "standard" RL critic, evaluating the agent's behavior according to an external fitness-based evaluation signal, ρ. This critic uses a fitness-based reward function $r^{\mathcal{F}}$ as defined in (2.7). As we have seen in Section 2.4.3, this function is also referred to as the *objective* or *extrinsic* reward function, as it explicitly rewards fitness-enhancing behaviors, *i.e.*, actions that directly relate to the task being learned (Singh et al., 2010).

The *emotion-based critic* generates four domain-independent reward features, each one evaluating a certain aspect of the agent-environment relationship. These signals map the result of each appraisal into scalar values that somehow indicate the *degree of activation/significance* of each dimension. In our approach we consider emotion-based reward features depending on properties of a *fitness-based adaptive critic* (residing within the RL agent in Figure 4.2), according to the notion in

(Barto et al., 1983). In particular, we denote by $V^{\mathcal{F}}(s)$ and $Q^{\mathcal{F}}(s,a)$ the fitness-based state-value and action-value functions analyzing the agent's behavior according to $r^{\mathcal{F}}(s,a,h)$, respectively.[3]

4.2.3 Emotion-based Reward Features

Formally, we consider a set of five reward features, $\Phi = \{\phi_{\mathfrak{n}}, \phi_{\mathfrak{gr}}, \phi_{\mathfrak{c}}, \phi_{\mathfrak{v}}, \phi_{\mathtt{fit}}\}$, where

- $\phi_{\mathfrak{n}}(z,a,h)$ denotes the *novelty* dimension associated with trying action a in observation z, given the history h;
- $\phi_{\mathfrak{gr}}(z,a,h)$ denotes the *goal relevance* of executing action a when observing z, given h;
- $\phi_{\mathfrak{c}}(z,a,h)$ denotes the degree of *control* over observation-action pair z,a given history h;
- $\phi_{\mathfrak{v}}(z,a,h)$ denotes the dimension of *valence* resulting from executing a after observing z according to history h;
- $\phi_{\mathtt{fit}}(z,a,h) = \rho(z,a,h)$ is the fitness-based reward for performing action a after observing z during h.

To define the space $\mathcal{R}$ of possible agent reward functions, we follow the *linearly parameterized approach to ORP* described in Section 2.5.1. We thus define $\mathcal{R}$ as the linear span of Φ,*i.e.*, the set of all reward functions r in the form

$$r(z,a,h) = \sum_{\phi_i \in \Phi} \phi_i(z,a,h)\theta_i = \boldsymbol{\phi}^\top(z,a,h)\boldsymbol{\theta},$$

where

$$\boldsymbol{\phi}(z,a,h) = [\phi_{\mathfrak{n}}(z,a,h), \phi_{\mathfrak{gr}}(z,a,h), \phi_{\mathfrak{c}}(z,a,h), \phi_{\mathfrak{v}}(z,a,h), \phi_{\mathtt{fit}}(z,a)]^\top$$

and

$$\boldsymbol{\theta} = [\theta_{\mathfrak{n}}, \theta_{\mathfrak{gr}}, \theta_{\mathfrak{c}}, \theta_{\mathfrak{v}}, \theta_{\mathtt{fit}}]^\top.$$

Each parameter $\theta^i \in [-1.0, 1.0]$ determines the contribution of the corresponding reward-feature ϕ^i to the overall reward.

By defining the reward function space in this manner, we allow for the discovery of reward functions providing the agent with different ways of appraising its environment, according to the particular parameter vector $\boldsymbol{\theta}$. For example, the parameter vector $\boldsymbol{\theta}^{\mathtt{fit}} = [0,0,0,0,1]^\top$, originating the fitness-based reward function $r^{\mathcal{F}}$, indicates that the agent is *predisposed* to focus only in *fitness-inducing behaviors* while completely ignoring the emotional appraisal of the events.

[3] We note that the fitness-based value functions only *evaluate* the agent's behavior according to the fitness-based reward function $r^{\mathcal{F}}$. As such, one must not confound them with the value functions calculated by means of the agent's reward function r, which are used to *guide* the agent throughout learning. We refer to Section 2.4.3 for further details.

We consider the configuration of each reward function $r(\boldsymbol{\theta})$ to be fixed throughout learning, *i.e.*, the parameters $\boldsymbol{\theta}$ are initially set for the agent and remain that way during its lifetime, following the learning scheme described in Section 2.4.3. This idea fits the notion of intrinsic rewards as part of a primary, hard-wired mechanism in natural organisms (Singh et al., 2010).[4]

Having defined the general framework for emotion-based reward design, we now describe our proposal for the emotion-based reward features mapping from the agent's history of interaction with its environment to scalar values. We note that this corresponds to a *possible interpretation* for the corresponding appraisal dimensions in the context of our approach and scenarios. In that manner, for each reward feature we refer to the corresponding theoretical support provided by appraisal theories of emotions and also discuss alternatives and related features already proposed within the RL literature.

Importantly, we also note that "high/low" or "negative/positive" feature values *do not correspond* necessarily to "good" or "bad" states or actions for the agent. Similar to the emotional appraisal occurring in biological organisms, the emotion-based features only *signal* specific characteristics of the agent's relationship with its environment that *may be* pertinent to its fitness and to which the agent *can* pay attention to. As occurs in nature, it will depend on the agent's experience and specific situation whether some states and actions are considered beneficial or detrimental to its well-being and also the amount of attention it should pay to each reward-feature's value.

Novelty

Novelty is one of the most basic and low-level dimensions of emotional appraisal of events, usually eliciting focus of attention to important changes occurring in the environment (Ellsworth and Scherer, 2003; Frijda and Mesquita, 1998; Reisenzein, 2009). As indicated in Table 3.1, there are several factors which contribute to the evaluation of an event's novelty, such as the level of habituation to a stimulus, the individual's motivation state or the perception of *predictability* or *expectedness* of a situation (Ellsworth and Scherer, 2003; Roseman, 2001). From Table 3.2, we see that at perception or schematic level, novelty usually refers to the degree of *familiarity* or matching between the perceived stimuli and the agent's knowledge structures built so far (Frijda and Mesquita, 1998; Leventhal and Scherer, 1987; Reisenzein, 2009).

In the RL framework, familiarity about states and actions is directly related to the number of visits to state-action pairs. Let us denote by $n_t(z)$ the number of times z was observed up to time-step t, and by $n_t(z,a)$ the number of times that action a has been executed after observing z so far. Therefore, one can quantify the dimension of novelty as the reward-feature

$$\phi_{\mathfrak{n}}(z,a,h) = \frac{\lambda_{\mathfrak{n}}^{-n_t(z,a)} + \lambda_{\mathfrak{n}}^{-n_t(z)}}{2}, \tag{4.1}$$

[4] We refer to (Sorg et al., 2010b) for a solution in which the parameter vector is modified online according to the gradient of the utility function.

where $\lambda_\mathfrak{n}$ is a positive constant such that $\lambda_\mathfrak{n} < 1$. $\lambda_\mathfrak{n}$ can be seen as a "novelty rate" determining how the novelty dimension decays with experience. This feature thus signals the amount of familiarity of states and actions experienced so far by the agent. If associated with a positive weight, novelty can be useful in scenarios where the environment is very dynamic or the reward opportunities change throughout time by indicating that states and actions experienced long ago are preferable. On the contrary, a negative novelty weight can be useful in scenarios where well-known states and actions are better and thus where familiar behavior "routines" are preferable[5].

For simplicity reasons, we chose to include only statistical variables in the calculation of the novelty reward-feature which somehow evaluate the amount of past experience with some observations and actions. However, one can envisage expressions that evaluate the *predictability* of stimuli or the *probability* of actions outcomes, all characteristics of the novelty dimension at a higher level in the appraisal processes occurring in natural agents (Ellsworth and Scherer, 2003; Frijda and Mesquita, 1998; Leventhal and Scherer, 1987). Such expressions could, for example, assess prediction errors in $V^\mathcal{F}$ and $Q^\mathcal{F}$, or discrepancies detected from the fitness-based reward received in relation to previous interactions.

Another alternative is to consider recency-based features, rewarding interactions with (z, a) pairs (not) visited recently (Singh et al., 2010; Sorg et al., 2010a,b). However, we prefer frequency-based features in favor of recency-based ones as they better capture the essence of novelty. Having observations and actions that are not experienced for a certain amount of time does not imply that they are novel, although they can also encourage exploratory behaviors (Singh et al., 2010). The expression proposed for novelty also resembles the inverse-frequency reward feature in (Bratman et al., 2012). However, instead of a linear decaying rate, we consider an exponential decaying rate that is dependent, for example, on the total number of observations and actions that can be experienced or the duration of the agent's lifespan. Hester and Stone (2012) present a novelty reward for domains with factored state by measuring the distance in feature space between perceived state-actions to indicate situations that are most different from previously experienced ones.

Goal Relevance

As its name indicates, this dimension asserts the *relevance* of a perceived event in terms of the attainment of the agent's long-term goals or satisfaction of its needs (Ellsworth and Scherer, 2003; Lazarus, 2001; Leventhal and Scherer, 1987). Also related to the notions of *desire-congruence* (Reisenzein, 2009) or *motive-consistence* (Roseman, 2001), goal relevance is essential for the survival and adaptation of an individual to its environment as it evaluates the further consequences of

[5] We again reinforce that the emotion-based features are independent of the action-selection/exploration strategy used during learning. In the case of novelty, if taken into account by the agent it can lead to non-stationary policies as the rewards for some state or state-action pair are always changing and consequently adjusting the respective V and Q-value functions. In fact, in situations where the environment is itself non-stationary such policies are indeed desirable, which is fundamentally different from changing the exploration strategy used, *i.e.*, in such cases, the agent is following less familiar states and actions because it is "good" to do it so, not because it is still exploring the environment.

a given situation in relation to its goals, needs or desires (Ellsworth and Scherer, 2003; Reisenzein, 2009).

As indicated in Table 3.1, goal relevance has a *motivational basis* and is influenced by the importance of the event and the consistency of its outcomes in relation to the goals or needs being concerned (Roseman, 2001). Broadly speaking, we can say that the goal relevance of an event increases if such event is consistent with an individual's goals, *i.e.*, the individual *approaches* its objectives, and decreases when the consequences of the event are *obstructive* to reaching those goals (Ellsworth and Scherer, 2003; Reisenzein, 2009).

At a low-level, the goal of an organism is to attain the maximum fitness to its environment throughout its lifetime. Recall from Section 2.4.3 that this is exactly the long-term goal of an artificial learning agent according to the IMRL framework (Singh et al., 2010). As such, we consider that the observations for which the expected fitness-based return (as indicated by the state-value function $V^{\mathcal{F}}$) is high should lead to a greater degree of fitness than those with a low fitness return expectancy.

For the purposes of our model, we assume that, at each time-step t, the agent has access to a distance, $\hat{d}(s_t, s^*)$, that corresponds to an estimate of the number of actions needed to move from its current state s_t to a so-called *goal-state*, denoted by s^*. The distance, *i.e.*, the number of steps, is estimated based on the state transition model $\hat{\mathsf{P}}(s' \mid s, a)$ learned by the agent. A state s is considered to be the goal-state s^* when its current expected fitness return provided by $V_t^{\mathcal{F}}(s)$ is maximal in relation to all other known states when the agent last visited it, *i.e.*, $s^* = \operatorname{argmax}_s V_t^{\mathcal{F}}(s)$. If there are multiple maximal states, it is considered the one that was discovered first. In this manner, s^* is not fixed beforehand, rather it is discovered whenever the agent perceives a state with expected value that is maximal at some point in time. We also note that this distance estimate needs not to be accurate, but should be coherent with the true values, *i.e.*, if $d(s_1, s^*) > d(s_2, s^*)$ then $\hat{d}(s_1, s^*) > \hat{d}(s_2, s^*)$, where $d(\cdot, \cdot)$ denotes the actual distance.

In our framework, goal relevance is thus translated in terms of the numerical value

$$\phi_{\mathfrak{gr}}(z, a, h) = \frac{1}{1 + \hat{d}(s, s^*)}, \tag{4.2}$$

where s is the state underlying observation z.

The behavior of this expression seems to be coherent with the role of the goal relevance appraisal dimension in natural organisms. It makes the relevance of some observation decrease as its perceived distance to the goal state increases, and have the value of 1 when the agent gets the closest to its goal, *i.e.*, only when the agent achieves a state capable of fostering maximal fitness. Moreover, because the goal-state is not fixed beforehand it allows that different states can become the "goal" overtime whenever the associated $V^{\mathcal{F}}$ is maximal. This means that when taken positively, the goal relevance feature allows the agent of better adapting to environments where the "source of fitness" is altered throughout time, *e.g.*, the value of reward changes or the goal-state becomes unreachable. On the other hand, a negative weight associated with goal relevance can be useful in situations where relying in lower fitness-based rewards is beneficial compared to aiming at high-valued but possibly more irregular states[6].

Alternatives to this formulation can account for example the Manhattan distance to goal states as proposed in (Bratman et al., 2012), but with the disadvantage of having to specify such states beforehand.

Control

Control is part of a group of appraisal dimensions associated with the *coping potential* of an individual (Ellsworth and Scherer, 2003; Smith and Kirby, 2009), as denoted in Table 3.1. This dimension is usually defined involving a more proactive assessment of the ability of the individual to deal with a particular situation, *i.e.*, the potential of an organism to cope with the situation being evaluated (Ellsworth and Scherer, 2003; Frijda and Mesquita, 1998; Lazarus, 2001; Leventhal and Scherer, 1987). Because of that, this dimension is considered to be part of a "secondary" level of appraisal as they require an individual to determine an appropriate response to the event being evaluated (Lazarus, 2001).

Table 3.2 describes the several levels at which the assessment of the coping potential can occur. At a higher level of cognitive appraisal, this evaluation implicates the attribution of *causal agency* for the event, influencing the significance and kind of response to the situation at hand (Ellsworth and Scherer, 2003; Roseman, 2001), or cognitively adjusting one's goals in order to *fit* the outcomes of the situation (Lazarus, 2001; Smith and Kirby, 2009). At a rather lower level of information processing, coping potential estimates the extent to which an event or its outcomes can or cannot be *controllable*, and whether the organism has the sufficient *power* to change the situation to its benefit (Frijda and Mesquita, 1998; Roseman, 2001; Scherer, 2009).

In our framework, we follow the perspective that often the control over a situation involves determining the degree of *predictability* or *probability* of the events being considered (Ellsworth and Scherer, 2003; Leventhal and Scherer, 1987; Roseman, 2001; Scherer, 2009). By following this approach, an agent can determine the degree of *correctness/clarity* of the world-model that it has built of its own environment which directly influences its ability to control it. The more clear and correct the model is, the greater the control and power the agent has over its environment and thus the greater its potential is in coping with a particular situation.

As we have seen in Section 2.2.3, the action-value function $Q(s, a)$ for a certain policy can be estimated from the rewards received in the several states encountered through time by following such policy in the environment. Given an observed transition (z, a, r, z'), the *Bellman error* associated

[6] We refer to the Poisoned Prey Scenario in Section 4.3.5 for an example of an environment with such dynamics.

with $Q_t^{\mathcal{F}}$ is given by

$$\Delta Q_t^{\mathcal{F}}(z, a) = \hat{r}_t^{\mathcal{F}}(z, a) + \gamma \max_{b \in \mathcal{A}} Q_t^{\mathcal{F}}(z', b) - Q_t^{\mathcal{F}}(z, a). \tag{4.3}$$

We can therefore calculate the value of control by considering the expected Bellman error associated with $Q_t^{\mathcal{F}}$ at (z, a) through the numerical value

$$\phi_{\mathfrak{c}}(z, a, h) = 1 - \mathbb{E}\left[\Delta Q_t^{\mathcal{F}}(z, a)\right]. \tag{4.4}$$

Because the total reward depends itself on the control value, we consider the fitness-based reward function $r^{\mathcal{F}}$ to calculate the prediction error. From the above expression we can see that the lower the prediction error, the greater the agent's coping potential will be, and hence the more accurate the current world-model is for the given action and observation. Observations which associated action-values often change will have higher prediction errors and therefore denote irregularities in the environment that the agent does not fully control. As noted with the previous features, the way the control feature is weighted will depend on the environment's dynamics. If taken positively, control favors more predictable states and actions thus allowing a faster learning in environments where more uncertain situations are detrimental when compared to more controllable ones. On contrary, in scenarios where fitness enhancement may come from states and actions changing very often, a "negative control" value can be advantageous.

Other factors contributing to the prediction errors can account for discrepancies in the state transition model perceived throughout time, as proposed by the *quality of model* feature in (Sorg et al., 2010a). The *variance* reward in (Hester and Stone, 2012) follows a similar approach by measuring the variance in the predictions of decision trees used to model the dynamics of the learning domain. In (Lopes et al., 2012), a measure for *model accuracy* and *learning progress* related to control is presented where the likelihood of recently-visited state-action data-sets are evaluated over the complete (experienced) data in order to determine whether particular state-actions need further exploration or not. Within the area of emotion-based RL, the *uncertainty model* proposed in (Ahn and Picard, 2006), calculates the level of emotional *arousal* by considering discrepancies between the current state's value and expected extrinsic rewards from the same action in other states by means of a standard deviation measure. Other alternatives for the control feature could for example account for the difference between the global (across states and actions) maximum and minimum Q-values, meaning that large differences would indicate a greater control over the environment.

Valence

Valence measures how *intrinsically pleasant* a given situation is (Ellsworth and Scherer, 2003). It is either generated from a set of innate detectors or learned preferences/aversions that basically indicate whether a stimulus is "positive" or "negative" in terms of biological significance for the organism (Leventhal and Scherer, 1987). Valence is considered a very low-level and automatic appraisal dimension capable of evoking approaching behaviors to pleasant stimuli and aversion to unpleasant situations. Moreover, unlike other dimensions, valence is considered to be mainly a feature of the stimulus itself, independent of the momentary situation of the organism (Ellsworth and Scherer, 2003).

In our framework we consider the fitness-based reward function $r^{\mathcal{F}}$ to be directly related with the fitness utility-function $f(h)$ evaluating the performance of the agent in some environment, as described in Section 2.4.3. The external signal $\rho(s,a,h)$ received by the agent directly corresponds to the biological significance attained by executing action a in state s. Therefore, the fitness-based reward feature could itself be used to provide the evaluation of valence, *i.e.*, we could denote valence as $\phi_{\mathfrak{v}}(z,a,h) = \phi_{\mathfrak{fit}}(z,a,h)$.

However, in our approach we follow the perspective that the *implicit value* of things can change throughout time according to experience (Cardinal et al., 2002; Ellsworth and Scherer, 2003; Leventhal and Scherer, 1987). While some preferences are phylogenetically hardwired, other evaluative processes are acquired to signal preferences or dislikes over never-before-experienced stimuli (Ellsworth and Scherer, 2003). Furthermore, by means of associative processes, stimuli that were considered preferable or neutral can become unpleasant through the association with other biologically-relevant aversive stimuli (Cardinal et al., 2002).

Bearing this idea in mind, we can consider the fitness-based reward function as a source of *innate preferences* over some states and actions of the environment. In order to account for the adaptive nature of valence with experience, we consider the following numerical value to evaluate the *pleasantness* of a certain situation:

$$\phi_{\mathfrak{v}}(z,a,h) = \frac{V_t^{\mathcal{F}}(z) - \min_{z'} V_t^{\mathcal{F}}(z')}{2(\max_{z'} V_t^{\mathcal{F}}(z') - \min_{z'} V_t^{\mathcal{F}}(z'))} + \frac{Q_t^{\mathcal{F}}(z,a)}{2V_t^{\mathcal{F}}(z)}, \tag{4.5}$$

where $\min_{z'} V_t^{\mathcal{F}}(z')$ and $\max_{z'} V_t^{\mathcal{F}}(z')$ are the current minimal and maximal values of the fitness-based state-value function, respectively.

This expression basically indicates how "good" observation z is when comparing its current expected fitness return with all other observation-values, and also how "good" it is to execute action a after observing z when compared with other already-experienced actions given the same observation. As such, valence will have a value of 1 only when the agent executes the action with highest action-value after the observation with the highest observation-value, thus denoting a learned preference by the agent towards a behavior which it currently believes will lead to a high degree of fitness in the environment. As should be apparent, valence is useful to evaluate states that, although providing a low fitness-based reward on the immediate, may foster greater degrees of fitness on the long-run, *i.e.*, when adopting a long term perspective is better than focusing on instant reward.

A possible alternative for this expression can consider the reward proposed in (Ahn and Picard, 2006), where a sensation of feeling good or bad is given as a measure of the expected reward given the current state and an action in relation to the average reward received in that state. However, such expression evaluates only immediate reward, and we believe our feature captures the essence of valence as an evaluation towards the agent's long-term goals. A better alternative for this feature can be found in (Broekens et al., 2007), where the window-limited short-term running average of the (fitness-based) reinforcement is measured against its long-term running average to provide the reward (and valence) with which the agent learns.

4.3 Experiments and Results

In Figure 4.1 we outline the method to validate our approach for emotion-based reward design. In order to assess the potential of using an emotion-based mechanism to provide intrinsic reward we designed a set of experiments in foraging environments inspired by those in (Singh et al., 2010).

4.3.1 Objectives

The choice of foraging scenarios in our approach is tightly connected with the objectives of the experiments. First of all, foraging scenarios enable an easy evaluation of the agent's behavior as prescribed by the designer's objective. As such, the fitness-based reward functions for our scenarios are related to feeding behaviors, which in natural environments directly enhance most biological organisms' fitness. The foraging scenarios also enable us to test whether the different reward-features lead to distinct behaviors, depending on the environments. We observe the emergence of different strategies to attain fitness in different environments, but also maladaptive behaviors leading to poor performances.

Foraging scenarios also fit our purpose of evaluating whether an adaptation of an emotional appraisal-based mechanism brings advantages when designing artificial learning agents. In most scenarios, the agent will have some perceptual limitations that prevent it from attaining maximal fitness if it engages in fitness-inducing behaviors only, or, loosely speaking, if it only cares about eating. As such, we note that the majority of the environments proposed in the experiments do not hold the Markov property from the perspective of the agent's observations, since the information about the location of the preys or other predators cannot be directly determined from the current observation (Sutton and Barto, 1998). Therefore, just like biological agents encounter challenges in nature, our agents will have to discover particularities of the environments that allow them to achieve a better fitness. Because the agents are "guided" by appraisal-based reward features, the proposed foraging scenarios enable us to assert the usefulness of an emotion-based implementation in IMRL agents.

4.3.2 Methodology

In our experiments, the agent is modeled as predator moving in the environment, trying to eat preys and, in some scenarios, avoiding encounters with other predators.

Agent Description

In all scenarios, at each time-step, the agent is able to observe its *position* in the environment and whether it is collocated with a *prey*. Also for all scenarios the agent has available at least four possible actions, *i.e.*, $\mathcal{A} = \{Up, Down, Left, Right\}$, each deterministically moving the agent to the adjacent cell in the corresponding direction.

We use *prioritized sweeping* (Moore and Atkeson, 1993) to learn a memoryless policy that treats the observations of the agent as states.[7] Prioritized sweeping uses a model of the environment (namely of the state transitions and reward function) in order to back-propagate action-values to state-action pairs that have trajectories going to the current state perceived by the agent. We note that, in our approach, both the transition function $\hat{\mathsf{P}}$ the reward function $\hat{r}$ are learned online, *i.e.*, they are not known by the agent beforehand, following the learning scheme presented earlier in Section 2.2.3 which is depicted in Figure 2.4.

The agent updates a transition $\hat{\mathsf{P}}(s'|s, a)$ from state s to state s' after executing action a based on observed transitions according to

$$\hat{\mathsf{P}}(s'|s, a) = \frac{n_{s,a,s'}}{n_{s,a}},$$

where $n_{s,a}$ is the number of times action a was taken in s and $n_{s,a,s'}$ is the number of times s' was observed after executing a in s. The agent models a reward $\hat{r}(s, a, h)$ received after executing action a in state s by averaging the rewards received by means of the agent reward function in history h, *i.e.*, $\hat{r}(s, a, h) = r_{\text{ave}}(s, a, h)$.

In terms of the learning parameters, we use a learning rate $\alpha = 0.3$, a discount factor of $\gamma = 0.9$, a backup limit of 10 state-action pairs and a minimum priority threshold of 10^{-4}. The agent follows an ε-greedy policy with decaying exploration parameter $\varepsilon_t = \lambda_\epsilon^t$ with an exploration rate $\lambda_\epsilon = 0.9999$. We also use a novelty rate $\lambda_\mathsf{n} = 1.001$ required for the calculation of the novelty reward-feature in (4.1).

Computing Agent Fitness

Regarding the fitness function, we consider utility as measuring the total cumulative fitness-based reward received by the agent throughout its lifetime as defined by (2.8) for each history h^i sampled. We follow the reward function evaluation scheme outlined in Figure 2.8.

[7] We note that the fitness-based adaptive critic mentioned in Section 4.2.2 uses the same learning algorithm and parameters that the agent's emotion-based adaptive critic does, in order to calculate the fitness-based value functions $V_t^{\mathcal{F}}(s)$ and $Q_t^{\mathcal{F}}(s, a)$.

In our foraging scenarios, the fitness of the agent is then measured according to the total number of preys eaten and the particular fitness-based reward provided by each prey. Recall that we define a parameter vector $\boldsymbol{\theta} = [\theta_{\mathfrak{n}}, \theta_{\mathfrak{gr}}, \theta_{\mathfrak{c}}, \theta_{\mathfrak{v}}, \theta_{\mathfrak{fit}}]^\top$ weighting the contribution of each reward-feature to the total reward received by the agent at each time step. Therefore, different parameter vectors will yield different degrees of fitness, depending on the set of environments of interest, $\mathcal{E}$.

As seen in Section 2.4.3, the ORP is precisely the problem of, given $\mathcal{E}$, recovering the optimal reward function, $r^*(s, a, h)$, that provides the maximal degree of fitness. In this respect we adopt the optimization approach described in (Singh et al., 2010). In particular, we sample a total of $M = 14,003$ parameter vectors $\boldsymbol{\theta}_m \in [-1, 1]^5, m = 1, \ldots, M$, such that $\|\boldsymbol{\theta}_m\|_1 = 1$, and select the optimal vector $\boldsymbol{\theta}^*$ according to (2.12).

In our experiments, for each scenario, we generate a total of $N = 200$ histories as independent Monte-Carlo trials for each sampled reward function $r_m = r(\boldsymbol{\theta}_m), m = 1, \ldots, M$. For each history h^i, we simulate the agent for $T = 100,000$ learning steps in an environment E stochastically sampled according to $p_E(\mathcal{E})$. The number of environments in $\mathcal{E}$ depends on the specific scenario tested. For all scenarios we use a uniform distribution, *i.e.*, $p_E(e^i) = \frac{1}{|\mathcal{E}|}$ for all $e^i \in \mathcal{E}$, thus sampling each possible environment with equal probability.

We now provide a detailed analysis of each scenario, including the description of the environment's dynamics and the challenges each set of environments presents to the learning agent. For each scenario we provide the results of the optimization procedure, comparing the fitness attained by the agent using the optimal parameter vector $\boldsymbol{\theta}^*$ versus an agent using the fitness-based reward function using $\boldsymbol{\theta}^{\mathfrak{fit}} = [0, 0, 0, 0, 1]^\top$ corresponding to a traditional RL agent, and an agent receiving no reward, *i.e.*, $\boldsymbol{\theta}^0 = [0, 0, 0, 0, 0]^\top$ corresponding to a random-behavior agent. The objective is to assess the usefulness of the proposed emotion-based features when compared to an agent receiving the designer's fitness-based rewards and an agent behaving according to chance.[8]

4.3.3 IMRL scenarios

We start our experiments with three scenarios inspired by early work within the IMRL framework (Singh et al., 2009, 2010), namely the Hungry-Thirsty Scenario, the Lairs Scenario and the Moving Preys Scenario. These scenarios have the purpose of assessing the performance of our approach in scenarios already proposed in the literature to test the efficacy of using rewards different than those defined by the agent designer's objective. We note that we do not intend to perform a comparative analysis between our approach and others, but rather determine the usefulness of emotion-based rewards as a general-purpose learning mechanism that does not rely on any specific domain or task.

[8] Illustrative videos containing a comparison between the behaviors of the optimal emotion-based agent versus the fitness-based agent for each scenario are available online at http://gaips.inesc-id.pt/~psequeira/emot-design/.

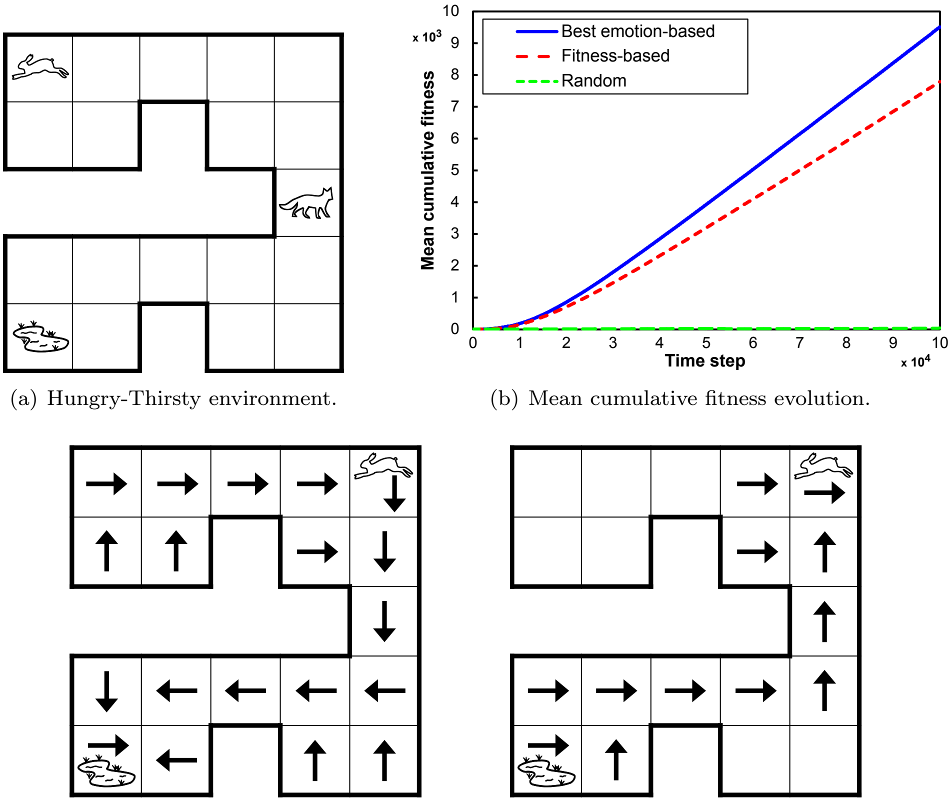
Hungry-Thirsty Experiment

This experiment is inspired by the Hungry-Thirsty domain proposed in Singh et al. (2009) and used in another work within IRML (Niekum et al., 2010). For reasons of simplicity, we have reduced the size of the environment to a 5 × 5 grid-world, consisting of four subspaces of 2 × 2 cells, as depicted in Figure 4.3(a).

> **Hungry-Thirsty Scenario**: In this scenario the agent has available two types of inexhaustible resources: a prey that the agent can eat, represented by a hare, and a pond from which it can drink water. These two resources are placed randomly in two different positions out of the 4 corners of the environment and remain fixed throughout the simulation. Figure 4.3(a) shows a possible configuration of the environment and also the agent's start position, where the agent is represented by a fox. As such, the set $\mathcal{E}$ of environments of interest for this scenario consists of a uniform distribution over the 12 different corner positioning configurations for the hare and the water.
>
> In terms of the dynamics of the scenario, at each time step the agent can be either thirsty or not-thirsty, depending on whether it drank water from the pond or not, respectively. After drinking, in successive time steps it becomes thirsty with a probability of 0.2. Also, eating the hare is the only behavior increasing the agent's fitness by providing it a fitness-based reward $\rho = 1$. This only happens however if the agent is currently not-thirsty, otherwise its fitness does not change. As we can see, the fitness function $f(h)$ for this scenario measures the total number of hares eaten by the agent (while not-thirsty) during learning. The state for this scenario is two-dimensional: besides its position the agent also observes its thirst status. The agent only has available the aforementioned movement actions in $\mathcal{A}$. For reasons of simplicity, we assume that the agent eats and drinks when collocated with the hare and the pond, respectively. $\Diamond$

We note that the assumption of the agent automatically eating and drinking while collocated with the resources does not impair the essence of the Hungry-Thirsty Scenario, which is to provide the agent the challenge of having to drink water before attaining fitness through food. In our scenario, the agent still needs to go to the pond when thirsty in order to be able to later eat the hare. We also note that this environment holds the Markov property, as the agent's state has access to whether it is thirsty and also its position, the only information needed to attain fitness and perform optimally within the scenario.

The comparative results of the experiments in the Hungry-Thirsty Scenario are described in Table 4.1. As we can see, the performance of the emotion-based agent using r^* surpassed the standard RL agent using $r^{\mathcal{F}}$ receiving fitness-based reward only after eating a prey. The high standard deviation of the mean cumulative fitness is related to the fact that different configurations of food and water provide very different fitness. In particular, the configuration in which the hare



(a) Hungry-Thirsty environment.

(b) Mean cumulative fitness evolution.

(c) Policy when *thirsty*.

(d) Policy when *not-thirsty*.

Figure 4.3: (a) The environment for the Hungry-Thirsty Scenario, inspired by the environment in (Singh et al., 2009). Each marked square corresponds to a possible position of the agent in the environment, non-crossable walls are denoted by bold lines; (b) The evolution of the mean cumulative fitness attained in the experiment over 100, 000 learning steps. We compare the optimal emotion-based agent, an agent receiving only fitness-based reward and a random-behavior agent. The results correspond to averages over 200 independent Monte-Carlo trials; (c)-(d) Policy learned by a single agent using the best reward function r^* within a particular environment configuration of the Hungry-Thirsty Scenario. Squares with no arrow refer to states seldom visited, thus denoting very random policies. See text for more details.

Table 4.1: Mean cumulative fitness for the Hungry-Thirsty Scenario, for the optimal emotion-based agent using parameter vector $\boldsymbol{\theta}^*$, an agent receiving only fitness-based reward using $\boldsymbol{\theta}^{\mathrm{fit}}$ and a random agent, using $\boldsymbol{\theta}^0$.

Parameter Vector	$\boldsymbol{\theta}$	$= [$	$\theta_{\mathfrak{n}},$	$\theta_{\mathfrak{gr}},$	$\theta_{\mathfrak{c}},$	$\theta_{\mathfrak{v}},$	$\theta_{\mathfrak{fit}}]^\top$	**Mean Fitness**
Emotion-based opt.	$\boldsymbol{\theta}^*$	$= [$	$-0.4,$	$0.0,$	$0.0,$	$0.5,$	$0.1]^\top$	$9,505.6 \pm 7,303.6$
Fitness-based	$\boldsymbol{\theta}^{\mathrm{fit}}$	$= [$	$0.0,$	$0.0,$	$0.0,$	$0.0,$	$1.0]^\top$	$7,783.7 \pm 6,930.1$
Random	$\boldsymbol{\theta}^0$	$= [$	$0.0,$	$0.0,$	$0.0,$	$0.0,$	$0.0]^\top$	35.6 ± 40.6

and the pond are in the left side corners of the environment provides the toughest challenge for the agent. In this case, the agent has to go all around the environment to eat, and come all the way back when it is thirsty, causing a smaller total amount of fitness attained when compared to other configurations. Nevertheless, this difference is statistically significant when compared to the fitness-based agent for $p < 0.02$, and Figure 4.3(b) further supports the view that the optimized emotion-based reward function r^* provided a faster learning for the agent by considering other sources of reward besides the fitness-based one.

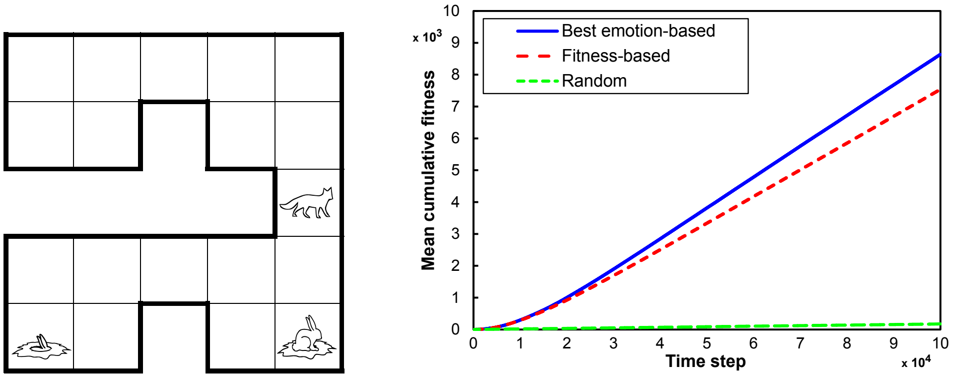
In Figures 4.3(c) and 4.3(d) we can see the policy learned by a single agent using the optimal emotion-based reward function, for a particular configuration of food and water when they are at opposite corners in the environment. As we can see, the emotion-based agent learned the optimal policy, going to the water when it is thirsty, and approaching the hare when not-thirsty. These results also confirm the idea in (Singh et al., 2009) about the emergence of a secondary, learned *preference* towards water, although water itself does not increase the agent's fitness. Furthermore, we note that to learn the optimal behavior, the emotion-based agent evaluated generic features of its relationship with the environment that had nothing to do with being hungry or thirsty, neither with the presence of food or water in its current location. This contrasts with the previous approach in a similar scenario which considered combinations of internal features of the agent to provide the reward, namely its *hunger* and *thirst* status (Singh et al., 2009).[9]

Lairs Experiment

This scenario is inspired by the Boxes domain presented in (Singh et al., 2009, 2010), where instead of boxes with food inside we consider lairs containing rabbits in a foraging environment. The structure of the environment is the same as in the Hungry-Thirsty Scenario, as can be seen from Figure 4.4(a), depicting a possible initial configuration for the scenario.

> **Lairs Scenario**: There are two lairs positioned in different corners of the environment. Each lair can be in one of three possible states: a lair can be empty of rabbits, occupied if there is a rabbit inside it, or the rabbit can be outside of the lair. As such, in this experiment there is a total of 6 different environments of interest based on the different configurations one can get from positioning the two lairs in the corners of the environment. As with the previous experiment, eating a prey, in any of the lairs, is the only source of fitness for the agent.
>
> The dynamics of the environment are as follows. Whenever a lair is occupied by a rabbit, the agent can pull it outside by means of a *Pull* action, in which case the lair's state goes to rabbit outside. When the rabbit is outside of a lair, the agent has exactly one time step to eat it by means of an *Eat* action, in which case it receives a fitness-based reward $\rho = 1$ and the lair transitions to an empty state. If, on the contrary, the agent does not eat the rabbit within one time step, the rabbit will run away, leaving the lair also in the empty state. In each successive time step, there is a chance of 0.1 of another rabbit appearing on the empty lair, again putting its state to occupied. ◊

[9] Despite acknowledging the (different) purpose of the Hungry-Thirsty experiment presented in (Singh et al., 2009), we are only emphasizing the generality of our method by not using domain-dependent reward features.



(a) Lairs environment.

(b) Mean cumulative fitness evolution.

Figure 4.4: (a) The environment for the Lairs Scenario, inspired by the environment in (Singh et al., 2010). In this particular configuration of the environment, the left lair is *occupied* by a rabbit, while in the right lair the rabbit is *outside*; (b) The evolution of the mean cumulative fitness attained in the scenario. The experimental procedure is the same as the above experiment. See text for more details.

Table 4.2: Mean cumulative fitness for the Lairs Scenario.

Parameter Vector	$\boldsymbol{\theta}$	$= [$	$\theta_{\mathfrak{n}}$,	$\theta_{\mathfrak{gr}}$,	$\theta_{\mathfrak{c}}$,	$\theta_{\mathfrak{v}}$,	$\theta_{\mathfrak{fit}}]^\top$	**Mean Fitness**
Emotion-based opt.	$\boldsymbol{\theta}^*$	$= [$	0.1,	0.0,	−0.2,	0.0,	$0.7]^\top$	8,635.8 ± 1,133.3
Fitness-based	$\boldsymbol{\theta}^{\text{fit}}$	$= [$	0.0,	0.0,	0.0,	0.0,	$1.0]^\top$	7,536.7 ± 944.8
Random	$\boldsymbol{\theta}^0$	$= [$	0.0,	0.0,	0.0,	0.0,	$0.0]^\top$	173.3 ± 13.5

The action space for this scenario thus corresponds to the set $\mathcal{A} = \{Up, Down, Left, Right, Pull, Eat\}$, explicitly including an action *Eat* which is necessary to distinguish situations in which the agent "captures" the rabbit or lets it run away. This environment is also Markovian as the agent's state comprises its position in the environment and the current state of both lairs, independently of the agent's position, and also because occupied lairs always contain a prey that can be eaten (Singh et al., 2010).

Table 4.2 includes the results of the parameter vector optimization procedure for the Lairs Scenario. Again, the results show a statistically significant ($p < 10^{-4}$) better performance and a faster learning of the agent using the optimal emotion-based reward function r^* when compared to the fitness-based agent. A visual analysis of the performance of the optimal emotion-based agent shows that the learned policy is to go from lair to lair, pulling and eating rabbits at a time when a lair becomes empty.

In comparison, by analyzing the performance of the fitness-based agent one sees that it concentrates in eating rabbits from only one of the two lairs. Because there is a small chance of the lairs becoming occupied after being empty, waiting for a rabbit in the same lair proves not to be the best strategy. Such optimal behavior policy is consistent with the one found in the Boxes experiment in (Singh et al., 2010). Again, the difference between the approaches lies in the fact

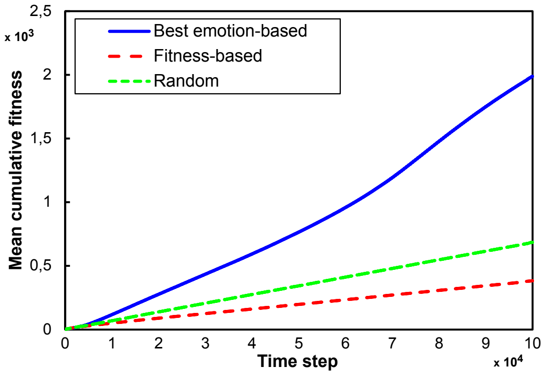


Figure 4.5: The evolution of the mean cumulative fitness attained in the Moving Preys Scenario. See text for more details.

that our emotion-based agent is not rewarded according to combinations of features from its direct observations.

Furthermore, by looking at the configuration of the optimal parameter vector $\boldsymbol{\theta}^*$ in Table 4.2, it seems that by rewarding less experienced states (through a positive *novelty* weight) and less controllable situations (by means of the negative *control* weight), the agent acquires behavior preferences towards changing of eating places after capturing a rabbit. This kind of "nomad" behavior observed in this experiment has itself nothing to do with fitness enhancement. Rather it can be considered as a primary intrinsic mechanism that, within the set of environments in this scenario, proved to be the one providing the best adaptive surviving strategy to the agent, which is exactly what we intend with our approach.

Moving Preys Experiment

In this experiment we resort to the Moving Preys Scenario defined in Section 2.2.1. Just as in the previous scenarios, in this experiment we assume that the agent automatically consumes the prey when collocated with it, and as such the set $\mathcal{A}$ of possible actions contains only the movement actions. The set $\mathcal{E}$ of environments for this scenario is composed of all environments of as depicted in Figure 2.2, the difference between them being the specific order of locations in which a new hare appears after one is consumed. Recall that in this scenario the agent receives a reward $\rho = 1$ whenever it eats a hare, which as with the previous experiments is the only behavior enhancing the agent's fitness.

In this scenario the agent does not have access to the current location of the prey, it can only observe its own location and whether a hare is present in its cell. This makes the environment non-Markovian in the agent's perspective because it does not have access to the necessary information to take the optimal decision from its current observation. As we have discussed in Section 2.3.2, partial observability poses a challenge to traditional RL algorithms that usually consider optimal stationary memoryless policies (Littman, 1994). Because the source of fitness-based reward is

Table 4.3: Mean cumulative fitness for the Moving Preys Scenario.

Parameter Vector	$\boldsymbol{\theta}$	$= [\ \theta_{\mathfrak{n}},$	$\theta_{\mathfrak{gr}},$	$\theta_{\mathfrak{c}},$	$\theta_{\mathfrak{v}},$	$\theta_{\mathfrak{fit}}]^{\top}$	**Mean Fitness**
Emotion-based opt.	$\boldsymbol{\theta}^*$	$= [\ 0.4,$	$0.0,$	$-0.1,$	$0.2,$	$-0.3]^{\top}$	$1,986.9 \pm 110.0$
Random	$\boldsymbol{\theta}^0$	$= [\ 0.0,$	$0.0,$	$0.0,$	$0.0,$	$0.0]^{\top}$	683.1 ± 25.7
Fitness-based	$\boldsymbol{\theta}^{\mathfrak{fit}}$	$= [\ 0.0,$	$0.0,$	$0.0,$	$0.0,$	$1.0]^{\top}$	381.3 ± 17.2

always “moving” in the environment, a traditional algorithm will never converge to an optimal policy that considers only the current observed state.[10] The objective of this experiment is then to provide the agent challenges similar to those encountered by natural agents that have to *search* and *explore* their environment to look for food resources.

The results in Table 4.3 confirm the idea that in this scenario, using only the fitness-based reward to learn a policy can lead to an extremely maladaptive strategy in the environment. In fact, in relation to such strategy the agent is better off behaving according to chance, *i.e.*, by not being *guided* at all by any reward. Moreover, the fitness-based parameter $\theta_{\mathfrak{fit}} = -0.3$ in the best parameter vector $\boldsymbol{\theta}^*$ confirms the hypothesis of fitness-based reward being in fact detrimental to fitness by providing a negative reward to the agent whenever it eats a prey.

The results also confirm the ones in (Singh et al., 2010) showing that providing intrinsic reward to the agent can motivate exploratory behaviors. In this case, such behaviors are beneficial for the agent in attaining a greater degree of fitness, especially when compared with a strategy that focuses only on fitness-inducing behaviors. This effect can be seen in Figure 4.5 portraying the evolution of the cumulative fitness throughout time.

By further looking at $\boldsymbol{\theta}^*$ we once again see that this kind of exploratory intrinsic motivation in our approach is fostered mainly by rewarding less experienced states and actions through *novelty* ($\theta_{\mathfrak{n}} = 0.4$), and slightly punishing *controlled* situations by means of the negative parameter $\theta_{\mathfrak{c}} = -0.1$. Together, the results of the Moving Preys Scenario further reinforce the usefulness and generality of our approach by mitigating a limitation so common to practical learning agents such as partial observability.

4.3.4 Persistence Experiment

In this experiment we wanted to test the possibility of an agent adopting a long term view of the world as opposed to focusing in immediate reward in order to achieve better performances in an environment.

> **Persistence Scenario**: The environment for this scenario is depicted in Figure 4.6(a), where two kinds of prey are always present, namely a hare in the top-right corner with

[10] We can envisage a policy that for example stores the previous locations of food encountered so far. However, the agent would still not know in which one the prey is. As such, to behave optimally the agent would have to observe the current location of the prey, which intentionally is not the case in this scenario.

Table 4.4: Mean cumulative fitness for the Persistence Scenario.

Parameter Vector	$\boldsymbol{\theta} \; = [$	$\theta_{\mathfrak{n}},$	$\theta_{\mathfrak{gr}},$	$\theta_{\mathfrak{c}},$	$\theta_{\mathfrak{v}},$	$\theta_{\mathfrak{fit}}]^\top$	**Mean Fitness**
Emotion-based opt.	$\boldsymbol{\theta}^* = [$	$-0.1,$	$0.1,$	$-0.1,$	$0.1,$	$0.6]^\top$	$1,879.8 \pm 11.2$
Fitness-based	$\boldsymbol{\theta}^{\mathsf{fit}} = [$	$0.0,$	$0.0,$	$0.0,$	$0.0,$	$1.0]^\top$	136.3 ± 1.4
Random	$\boldsymbol{\theta}^0 = [$	$0.0,$	$0.0,$	$0.0,$	$0.0,$	$0.0]^\top$	17.1 ± 0.7

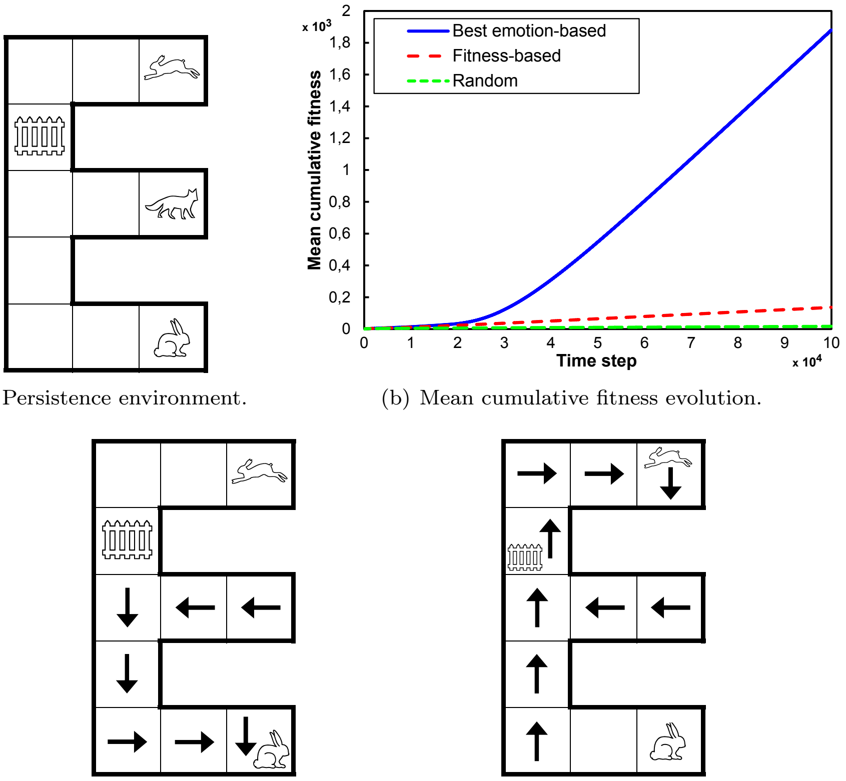
a value of $\rho = 1$ when eaten, and a rabbit in the lower-right corner providing a reward of $\rho = 0.01$. However, to get to the hare the agent must pass a fence as indicated in the figure. Initially, this fence is very weak, and the agent can pass it by performing only one *Up* action. After crossing the fence, the fence is reinforced so that the next time the agent will need to perform one more *Up* action to pass it up to a maximum of 30 actions, *i.e.*, $n_{Up} = \min\{n_{Up} + 1, 30\}$ whenever the agent crosses the fence, where n_{Up} is the number of *Up* actions needed to pass it.[11] In this scenario, we consider the environment just described as the only environment of interest.[12] The state of the agent is two-dimensional as it only observes its position and whether there is food in its location. ◊

Importantly, in the Persistence Scenario the agent does not known how many *Up* actions are needed to cross the fence. The lack of this information makes the environment non-Markovian in the agent's perspective as it is needed for optimal performance. This scenario is also episodic: the agent returns to its initial position after eating one of the preys. As mentioned in Section 3.3, one of the roles of emotions is to evaluate the current situation of the environment against the agent's goals and desires. In this case, our goal is for the agent to accumulate as much fitness-based reward as possible throughout time. Due to the differences of reward provided by the two preys in the environment, a "persistent" strategy directed towards the bigger future reward provided by the hare will naturally beat one that focuses on a smaller but immediate and easier reward given by the rabbit.

The results from this experiment are shown in Table 4.4, where the optimal emotion-based agent clearly outperformed the agent focusing only on fitness-based reward. This result can be better explained by looking at the policies learned by the two agents. As expected, the agent using the fitness-based reward function $r^{\mathcal{F}}$ focused on eating the rabbit providing a more immediate reward, as can be seen in Figure 4.6(c). The ε-greedy policy followed by the agent during learning allowed it to experience eating the hare in the initial phase of the simulation. However, as the fence got more difficult to cross, the agent ignored the bigger reward in favor of the more immediate

[11] We note that this obstacle only works when the agent is moving upwards across the fence, and not in the top-down direction, in which case the fence does not have any effect.

[12] In this experiment we are only concerned at finding a reward function enabling the agent of "learning to cross the fence" and as such it was not relevant to test other environments. Nevertheless, we could consider several configurations in which the fence was positioned in the environment so that the agent would have to cross it to get to the hare.



(a) Persistence environment.

(b) Mean cumulative fitness evolution.

(c) Fitness-based policy.

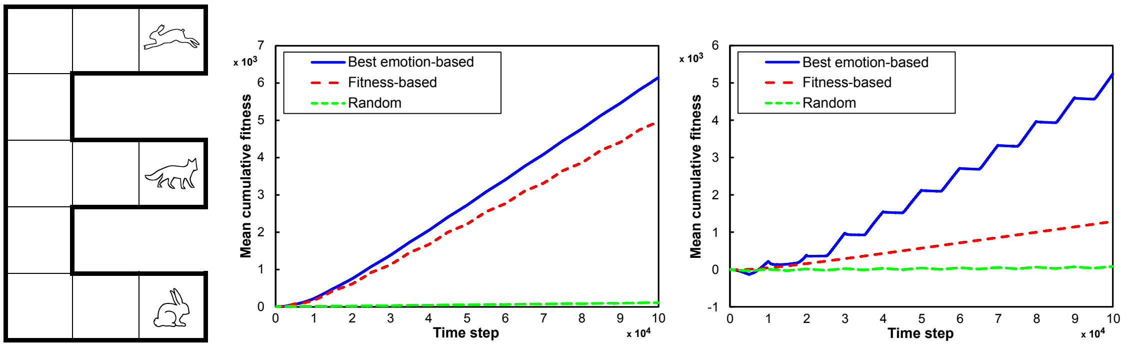
(d) Emotion-based optimal policy.

Figure 4.6: (a) The environment for the Persistence Scenario. The fence represents an obstacle the agent has to pass to access the hare. Each time it passes the fence, more difficult it will be for it to cross it the next time. The agent always starts in the indicated position after eating a prey; (b) The evolution of the mean cumulative fitness attained by each type of agent in the experiment; (c) Policy learned by a single agent using the fitness-based reward function $r^{\mathcal{F}}$; (d) Policy of an agent using the optimal emotion-based reward function r^{*}. Squares with no arrow refer to states seldom visited, thus denoting very random policies. See text for more details.

although smaller reward from eating the rabbits. This analysis is also supported by Figure 4.6(b), where we can see that the emotion-based agent only started to gain advantage over the other agents at about 20% of the simulation,[13] when the maximum number of actions was already required to cross the fence. On the contrary, the optimal emotion-based agent, experimenting both types of prey, preferred to continue on aiming at the hare prey in favor of the rabbit, as observable in Figure 4.6(d) showing the learned policy by a single agent using r^{*}.

Furthermore, by looking at the optimal weight vector $\boldsymbol{\theta}^{*}$ in Table 4.4, we see that besides the fitness-based reward, *goal relevance* and *valence* positively motivate the agent in crossing the fence to go to the hare. Recall from Section 4.2.3 that the goal relevance feature rewards the agent in approaching states with maximal expected fitness return. In this case, this corresponds to the hare location, where the agent always receives the bigger fitness-based reward of $\rho = 1$. Also, the valence feature rewards actions that lead to good fitness in the long-run, thus motivating the use

[13] This value was confirmed experimentally.



(a) Prey seasons environment.

(b) Different seasons results.

(c) Poisoned season results.

Figure 4.7: (a) Descriptive environment for the Prey Seasons Scenario and Poisoned Prey Scenario, in which two types of prey are available: hares and rabbits. In the Poisoned Prey Scenario environment there are always two preys available as depicted while in the Prey Seasons Scenario environment only one kind of prey is available at a time according to the season. In all scenarios the agent starts from the depicted position after eating. See text for more details; (b)-(c) Comparative results of the mean cumulative fitness evolution for the prey season scenarios.

of the *Up* action just before the fence.

Overall, in the Persistence Scenario a balanced combination of emotion-based rewards provided the best strategy for this scenario by motivating the agent in trying to cross the fence, although such behavior by itself has nothing to do with fitness enhancement.

4.3.5 Prey Season Experiments

The next set of scenarios present environments that not only change quickly as occurred in the Moving Preys Scenario, but also present different situations throughout time. The configuration of the environments changes cyclically according to "seasons" of 5,000 time steps. Each season presents food resources providing different degrees of fitness and/or appearing in different locations within the environment.

In this experiment all the scenarios are episodic—whenever the agent eats a prey it returns to the initial position in the environment. As with the Moving Preys Scenario and Persistence Scenario, the agent only has access to its location in the environment and the presence of food in its cell, and only has available the four movement actions. From the agent's perspective, all the season environments are non-Markovian because the agent does not know which season is currently "active", thus preventing the agent of acting optimally. As mentioned earlier, one of the advantages of emotions in nature is to provide biological organisms a mechanism to cope with the changes occurring in their environment. As such, this set of scenarios tests our hypothesis of emotion-based rewards guiding agents to quickly adapt to changing and ambiguous environments.

Table 4.5: Mean cumulative fitness for the Prey Seasons Scenario.

Parameter Vector	$\boldsymbol{\theta}$ = [	$\theta_{\mathfrak{n}}$,	$\theta_{\mathfrak{gr}}$,	$\theta_{\mathfrak{c}}$,	$\theta_{\mathfrak{v}}$,	$\theta_{\mathfrak{fit}}]^\top$	Mean Fitness
Emotion-based opt.	$\boldsymbol{\theta}^*$ = [	0.0,	0.1,	0.6,	0.0,	$0.3]^\top$	$6,142.3 \pm 1,336.3$
Fitness-based	$\boldsymbol{\theta}^{\mathfrak{fit}}$= [	0.0,	0.0,	0.0,	0.0,	$1.0]^\top$	$4,959.3 \pm 1,862.4$
Random	$\boldsymbol{\theta}^{\mathfrak{fit}}$= [	0.0,	0.0,	0.0,	0.0,	$0.0]^\top$	105.7 ± 24.4

Prey Seasons Experiment

Prey Seasons Scenario: In this scenario there are two types of preys, each one available to the agent during its own season. During the hare season, a hare appears on the top-right corner of the environment and during the rabbit season, a rabbit appears in the lower-right corner, as depicted in Figure 4.7(a). When eaten, a hare enhances the agent's fitness by $\rho = 1$ and a rabbit provides a reward $\rho = 0.1$. After eating the agent returns to its initial position as indicated in the figure.

In this scenario, hares "live in the wild" and as such provide the agent with an inexhaustible source of food. On the other hand, rabbits are "raised by a breeder". As such, during the rabbit season, the agent is only allowed to eat a maximum of 9 rabbits. If it tries to eat other rabbits after that, the breeder "shoots" at the agent which results in a loss of fitness in the amount of $\rho = -1$ (the agent still returns to its initial position). There are two environments of interest in the set $\mathcal{E}$ for this scenario, depending on which season starts first. ◊

The prey season at the start of the simulation significantly impacts the learned policy as the agent spends more time exploring the environment in the beginning of the simulation, thus influencing the type of prey it encounters more during that initial phase. Moreover, this scenario poses two kinds of challenges for the learning agent: it is *constantly changing* by means of the seasons and it is *ambiguous* by providing different rewards when eating a rabbit depending on the number of rabbits eaten so far—which the agent does not observe.

As we can see from from Table 4.5, the optimal emotion-based agent outperformed the fitness-based agent ($p < 10^{-4}$). By analyzing the performance of the two agents we see that both agents learn the same *good* policy for this scenario, which is to eat only during the hare season, avoiding being punished for eating too much rabbits during the rabbit season.[14] (see also Figure 4.7(b)). The difference of total cumulative fitness between the two agents is explainable by the relatively high positive *control* parameter $\theta_{\mathfrak{c}} = 0.6$ in the optimal parameter vector $\boldsymbol{\theta}^*$. This scenario shows an example in which following *safe* and *controlled* behaviors leads to a better adaptation to the environment, unlike with the previous experiments in which exploratory behaviors seeking novelty

[14] We note that the *optimal* policy would be to eat as much hares as possible during the hare season and eat only 9 rabbits during the rabbit season. Given the information the agent has access to from the environment we consider the learned policy to be a *good* policy for this scenario.

were required to gather more fitness-based reward. Because eating rabbits provides an ambiguous reward in the agent's perspective, it is preferable to avoid *uncertainty* and wait for the more *predictable* reward provided by eating hares. By not having this emotion-like mechanism, the fitness-based agent initially spends more time trying to eat rabbits, which at the end proved to be a disadvantage in terms of cumulative fitness.

Poisoned Prey Experiment

This scenario is an extension of the previous one and has the objective of reinforcing the idea of using emotions for coping in an changing and ambiguous world.

> **Poisoned Prey Scenario**: In this environment the two types of prey—hares and rabbits—are always available in the locations indicated in Figure 4.7(a). As before, the rabbit, when eaten, provides a fitness-based reward $\rho = 0.1$. During the normal season, the agent receives a reward $\rho = 1$ whenever it eats a hare. However, in the poisoned season eating hares is harmful for the agent as the hares are "poisoned due to toxic waste" being released during this period of time. Whenever this occurs, the agent's fitness is reduced by $\rho = -1$ by eating the toxic food. ◊

As with the Prey Seasons Scenario, the agent does not know which season is currently occurring. This scenario simulates the kinds of cyclic and drastic environmental changes that natural agents suffer in their environments. In such situations, a balanced strategy may provide the best outcome. On one hand, healthy hares are worth much more than the rabbits. However, the agent must quickly change its feeding behavior whenever it perceives that the hares are no longer the best choice of food during the poisoned season.

The results for this scenario depicted in Figure 4.7(c) show that in the end, both agents engaged in fitness-enhancing behavior strategies. However, in comparison, the total cumulative fitness attained shows that the optimal emotion-based agent largely outperformed the fitness-based agent. This is due to the fact the fitness-based agent, relying only on reward provided by the food, preferred to eat only the rabbits throughout time. In fact, such behavior makes sense from a fitness-only point of view. In this scenario, on average, eating the same amount of hares on the two seasons results in a fitness-based reward of $\rho = 0$. In other words, on average it does not compensate to go to the upper part of the environment to eat the hares.

In Section 3.2.3 we discussed that emotions are a mechanism that enable an individual in taking advantage of its environment, seeking food and avoiding harm by quickly adapting to the current situation. Therefore, the results from this experiment further support our claim that emotion-based rewards enable learning agents of adapting to an environment with low-predictability conditions.

Moreover, we note that the emotion-based agent learned a memoryless policy acting solely based on the current observation. By observing the resulting behavior learned by our agents, we can see

Table 4.6: Mean cumulative fitness for the Poisoned Prey Scenario.

Parameter Vector	$\boldsymbol{\theta}$	$= [$	$\theta_{\mathfrak{n}}$,	$\theta_{\mathfrak{gr}}$,	$\theta_{\mathfrak{c}}$,	$\theta_{\mathfrak{v}}$,	$\theta_{\mathfrak{fit}}]^\top$	**Mean Fitness**
Emotion-based opt.	$\boldsymbol{\theta}^*$	$= [$	0.1,	-0.2,	0.1,	0.0,	$0.6]^\top$	$5,237.6 \pm 77.2$
Fitness-based	$\boldsymbol{\theta}^{\mathrm{fit}}$	$= [$	0.0,	0.0,	0.0,	0.0,	$1.0]^\top$	$1,284.3 \pm 4.3$
Random	$\boldsymbol{\theta}^{\mathrm{fit}}$	$= [$	0.0,	0.0,	0.0,	0.0,	$0.0]^\top$	80.6 ± 24.9

the benefits of having a mechanism providing rewards that *appraise* the history of interaction with the environment. In fact, the emotion-based agent is intrinsically motivated to cope with drastic changes by preferring more *secure* behaviors—eating the rabbits, but also to take *risk* and take advantage when the hares are healthy.

As opposed to what occurred in the Persistence Scenario, going to the upper part of the environment by itself can lead to a poor performance on average, so other aspects had to be considered in this case. Table 4.6 presents a comparative analysis for this experiment. As we can see, the best strategy found for this scenario is a balanced consideration of different aspects of the agent-environment relationship. On one hand, positive *novelty* ($\theta_{\mathfrak{n}} = 0.1$) encouraged exploration in the environment required for when seasons change. On the other hand, predictable states by means of positive *control* ($\theta_{\mathfrak{c}} = 0.1$) motivate behaviors such as eating rabbits. More significantly, relying on goal states in this experiment has a negative impact, as denoted by the negative *goal relevance* parameter $\theta_{\mathfrak{gr}} = -0.2$.

4.3.6 Universality of Parameter Vectors

Thus far we have been silent about the universality of the optimal parameter vectors $\boldsymbol{\theta}^*$, *i.e.*, about what would happen if we took the optimal emotion-based agents discovered by the optimization procedure for each scenario and tested them in the other scenarios. As an example, in this section we present a brief comparative analysis of the performance of all the optimal emotion-based agents attained when acting in the Persistence Scenario environment described in Section 4.3.4. For each scenario we took the optimal parameter vector $\boldsymbol{\theta}^*$ providing the highest fitness for each scenario and tested it in the Persistence Scenario.

The results of this experiment can be seen in Table 4.7. As we can observe there is a large difference in the cumulative fitness attained by the optimal emotion-based agent in this scenario when compared with the optimal agents of the other scenarios. Moreover, all other agents performed worse than the fitness-based agent in the Persistence Scenario, and two of them even worse than the random agent.

Recall from the results of the Persistence Scenario in Section 4.3.4 that a strategy correctly balancing all the reward-feature weights allowed the optimal emotion-based agent to learn to "cross the fence" in order to get to the higher fitness-based reward. By not considering this balance, the

Table 4.7: Mean cumulative fitness attained by the optimal parameter vector $\boldsymbol{\theta}^*$ discovered in each experiment in the Persistence Scenario, sorted descendingly. We also include the results for the fitness-based agent $\boldsymbol{\theta}^{\mathfrak{fit}}$ and a random agent using $\boldsymbol{\theta}^0$.

Scenario	Parameter Vector	Mean Fitness
	$\boldsymbol{\theta} = [\ \theta_{\mathfrak{n}},\ \theta_{\mathfrak{gr}},\ \theta_{\mathfrak{c}},\ \theta_{\mathfrak{v}},\ \theta_{\mathfrak{fit}}]^\top$	
Persistence Scenario	$\boldsymbol{\theta}^* = [-0.1,\ 0.1, -0.1,\ 0.1,\ 0.6]^\top$	$1,879.8 \pm 11.2$
Fitness-based	$\boldsymbol{\theta}^{\mathfrak{fit}} = [\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 1.0]^\top$	136.3 ± 1.4
Lairs Scenario	$\boldsymbol{\theta}^* = [\ 0.1,\ 0.0, -0.2,\ 0.0,\ 0.7]^\top$	67.3 ± 2.1
Poisoned Prey Scenario	$\boldsymbol{\theta}^* = [\ 0.1, -0.2,\ 0.1,\ 0.0,\ 0.6]^\top$	60.5 ± 1.5
Moving Preys Scenario	$\boldsymbol{\theta}^* = [\ 0.4,\ 0.0, -0.1,\ 0.2, -0.3]^\top$	47.3 ± 4.2
Random	$\boldsymbol{\theta}^0 = [\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0]^\top$	17.1 ± 0.7
Hungry-Thirsty Scenario	$\boldsymbol{\theta}^* = [-0.4,\ 0.0,\ 0.0,\ 0.5,\ 0.1]^\top$	14.5 ± 5.8
Prey Seasons Scenario	$\boldsymbol{\theta}^* = [\ 0.0,\ 0.1,\ 0.6,\ 0.0,\ 0.3]^\top$	11.5 ± 1.5

other agents, which strategies allowed them to thrive in their respective environments, performed very poorly in this scenario and did not learn the optimal behavior. This example serves to illustrate the *non-universality* of the optimal parameter vectors for each scenario, as each is tuned for a specific environment. However, as will be discussed further ahead, this fact does not invalidate the *generality* of our approach, as the reward-features used by the agents are the same in all experiments.

On the other hand, it is also important to assess whether there is a universal or "good enough" parameter configuration, *i.e.*, one that is better on average than the fitness-based agent on all scenarios. For that purpose we made another experiment where we measured the average "rank" of each parameter vector across all the foraging scenarios. The rank position for a specific scenario is taken by ordering the several parameter vectors according to the mean cumulative fitness attained by the respective agent in that scenario. This means that the optimal weight vector $\boldsymbol{\theta}^*$ for a scenario is the highest ranked parameter vector, *i.e.*, $rank(\boldsymbol{\theta}^*) = 0$. We then averaged the rankings of all the tested parameter vectors to determine the one performing the best on average across all scenarios. We denote by $\boldsymbol{\theta}^U$ the universal parameter-vector. Table 4.8 presents the comparison in ranking of the universal emotion-based agent versus the fitness-based and random agents. The difference in averaged performance between the the universal and the fitness-based agent was found to be statistically significant ($p = 8 \times 10^{-4}$).

As the results of this experiment indicate, there are in fact parameter vectors that behave well, on average, in all the foraging scenarios when compared to an agent receiving only fitness-based reward. As $\boldsymbol{\theta}^*$ indicates, a combination of fitness-based reward and negative control allows the universal agent of attaining a better performance on average when compared to the standard agent. The negative weight on control is justified by the non-stationarity of most of the foraging scenarios, thus favoring exploratory strategies in the environments.

Table 4.8: Mean rank positions for the "universal" emotion-based, fitness-based and random agents across all the foraging scenarios. See text for details.

Parameter Vector	$\boldsymbol{\theta}$	$= [$	$\theta_{\mathfrak{n}},$	$\theta_{\mathfrak{gr}},$	$\theta_{\mathfrak{c}},$	$\theta_{\mathfrak{v}},$	$\theta_{\mathfrak{fit}}]^\top$	**Mean Rank**
Universal	$\boldsymbol{\theta}^U$	$= [$	0.0,	0.0,	−0.3,	0.0,	$0.7]^\top$	522.7 ± 460.2
Fitness-based	$\boldsymbol{\theta}^{\text{fit}}$	$= [$	0.0,	0.0,	0.0,	0.0,	$1.0]^\top$	779.8 ± 602.0
Random	$\boldsymbol{\theta}^0$	$= [$	0.0,	0.0,	0.0,	0.0,	$0.0]^\top$	6243.0 ± 2996.7

Table 4.9: Comparison of the performance of the universal emotion-based agent and the optimal and fitness-based agents in each foraging scenario.

Scenario	**Mean Fitness**		
	Universal	Optimal	Fitness-based
Hungry-Thirsty Scenario	8,297.8 ± 5,933.5	9,505.6 ± 7,303.6	7,783.7 ± 6,930.1
Lairs Scenario	8,798.0 ± 1,576.6	8,635.8 ± 1,133.3	7,536.7 ± 944.8
Moving Preys Scenario	460.8 ± 49.2	1,986.9 ± 110.0	381.3 ± 17.2
Persistence Scenario	470.8 ± 59.0	1,879.8 ± 11.2	136.3 ± 1.4
Prey Seasons Scenario	4,912.0 ± 2,606.3	6,142.3 ± 1,336.3	4,959.3 ± 1,862.4
Poisoned Prey Scenario	1,279.7 ± 5.2	5,237.6 ± 77.2	1,284.3 ± 4.3

However, as one would expect given that $rank(\boldsymbol{\theta}^U) = 522.7$, when the performance of this "universal" parameter vector is taken in each scenario individually, the results are only marginal in terms of the fitness attained compared to the optimal parameter vectors, as indicated in Table 4.9. Nevertheless, when compared to the fitness-based agent, in all but the Hungry-Thirsty Scenario, Prey Seasons Scenario and Poisoned Prey Scenario scenarios the universal agent performed significantly better ($p < 10^{-4}$). In the referred scenarios the performance of the universal agent was in line with that of the fitness-based agent.

The results of this experiment further show the existence of a parameter vector that, despite not being "specialized" in any particular environment, is "good enough" across all scenarios, especially when compared to an agent learning only with the external task reward. In the context of our study this result thus point towards the general-purpose usefulness of emotion-based rewards in solving complex learning tasks. We note however that such universal configuration is still *dependent* on the particular set of foraging environments in which learning took place. It is therefore expected that, in scenarios that have dynamics and challenges quite distinct from those presented by our foraging scenarios, the discovered universal agent performs worse than the agent best adapted to such environments or even the standard fitness-based agent.[15] A more critical discussion on this subject of universality is provided in the following section.

[15] We refer to the experiments in Section B.1 where we test the performance of the discovered "universal" parameter vector in different scenarios with varying complexity in the presented challenges.

4.4 Discussion

Now that the we provided both our approach and the validation for the design of rewards based on emotions, it is important to analyze our results in light of the several areas of research that contributed to them, namely emotion-based research in the field of RL and psychological work on emotions.

4.4.1 Emotions and Motivation

Recall from Section 3.2 that emotions play a fundamental role in several aspects of humans' cognitive and behavioral processes, such as such as memory enhancement, sensory plasticity, attention focusing, regulation of social behavior, etc. In this chapter we propose the use of emotion-based rewards that are able to complement the perceptions of the agent and guide it throughout its interaction with the environment. The way the agent *appraises* its environment by means of the intrinsic reward features together with the parameter vector used makes it focus on different aspects of the interaction and act accordingly. As discussed earlier, our focus is on the *motivational* aspect of emotions. Just as occurs with natural organisms, different evaluations—the rewards—provide distinct motivations—as indicated by the Q-values that they influence—to deal with the problems at hand.

We acknowledge the fact that emotions could, in parallel to what occurs in nature, also influence other aspects of a RL agent. We could consider for example an emotional mechanism focusing on particular stimuli from the environment, or influencing the rate at which the RL agent learns according to the positive/negative emotional tone associated with some state, or even adjusting the amount of exploration according to the level of control over the environment. In Section 3.5 we discussed other approaches to emotion-based RL that had some of these aspects of emotions into consideration for the design of the whole learning mechanism.

Nevertheless, we note that this is not a limitation of our approach as we *do not propose* a computational model of emotions that attempts to incorporate *all* the aspects of the emotional process for the design of the agent. Moreover, such emotional mechanisms within the RL framework would have to undergo independent processes of optimization (or manual tuning) in order to discover the best parameterization for some scenario. By focusing in the motivational role of emotions through the design of emotion-based reward functions, we concentrate on optimizing only the particular parameter configuration—hence the particular combined reward signal—that best guides the agent in a specific domain.

4.4.2 Emotions as Intrinsic Reward

In this thesis, particularly in this chapter, we explore whether emotions, from an ecological point of view, fit the idea of intrinsic rewards as primary and biologically-relevant rewards to natural organisms (Singh et al., 2010), as discussed in Section 2.7.1. We follow the perspective in which intrinsic rewards encourage behaviors that, throughout evolution, proved to contribute to an organism's reproductive success, *i.e.*, its fitness. Intrinsic rewards foster courses of action that are not directly related to enhancing the animal's fitness, *i.e.*, behaviors unrelated to eating, drinking or sex that directly reduce some internal biological drive.

The motivation behind our approach comes precisely from the fact that emotions are a hard-wired, phylogenetic adaptive mechanism that, throughout evolution, allowed animals to avoid dangers and gain fitness in the environment. Although emotions are tied to an individual's desires and goals, emotions may motivate behavior not directly related to fitness enhancement. By assessing the value or the significance of stimuli in the environment, emotions promote behaviors directed towards *dealing* with a situation so to improve the individual's well-being.

One might argue that some reactions to emotion-inducing events are extrinsically motivated, such as when in a fear conditioning experiment (see Section 3.2.1) a rat responds with immobility, as a result of fear, in the presence of a stimulus previously associated with an electrical shock. However, as the emotional processing mechanism becomes more complex, emotions elicit behaviors unrelated to extrinsic pain or pleasure stimuli. For example, one might feel *curious* and explore unfamiliar situations "just for the fun of it", or *sad* due to an unfavorable outcome without the need to be *fearful* of some extrinsic punishment. One can even feel *ashamed* of having done some action which is not acceptable within its social circle.

In essence, humans and other animals seem to respond emotionally to situations because evolution and individual experience dictates so. Together, all these ideas seem to support our claim of emotions being a possible and natural evaluative mechanism for autonomous learning agents in the general case.

4.4.3 Differences in Appraisal

Recall also from Section 3.3.1 that phylogenetic characteristics together with experience allow for individual and cross-cultural differences in emotional experience and emotion-related behaviors. This is a relevant phenomenon which is demonstrated by our model. In our approach, the emotion-based evaluative mechanism depends both on *individual characteristics* of the agent, *i.e.*, by means of the particular parameter vector providing the intrinsic reward, and also on *experience* as provided by the course of behavior performed by the agent and the rewards received during learning.

Additionally, because the agent uses a reward function based on emotion-like signals, our model follows the perspective that affective states may encode useful information that *guide* an agent during learning and decision-making (Isen, 2008; Naqvi et al., 2006). The proposed emotion-based features consider properties of the agent's history of interaction with its environment and not solely properties of the current state and action. Although we do not explicitly consider the values of previous features for the calculation of the rewards in each time step, the proposed evaluative mechanism resembles the process of reappraisal (Lazarus, 1966, 2001) in the sense that past information contributes for the current and future appraisals.

4.4.4 Connection with Computational Models of Emotions

As stated earlier, in this thesis we do not propose a new computational model of emotions. As described in Section 3.4.1, computational models based on appraisal theory specify emotional components that interact with entities such as memory, inference, reasoning or the agent's beliefs, as illustrated in the agent architecture of Figure 3.3. Although our IMRL-based agents do not have access to such components, we can still discuss how our model fits the above-mentioned architecture.

Recall from Section 3.3.1 that the person-environment relationship is the basis of appraisal, representing a connection between what is perceived from the environment and past interactions with it, the individuals beliefs, norms and cultural background (Frijda and Mesquita, 1998; Roseman and Smith, 2001; Smith and Kirby, 2009). In our framework, this entity is represented by everything the learning agent has access to in order to make its decisions, *i.e.*, its history h of interaction with the environment. In our emotion-based agent model, depicted in Figure 4.2, we see that the reward features based on appraisal dimensions correspond to the *appraisal variables* proposed for computational appraisal models (Marsella et al., 2010).[16] Like the appraisal dimensions in which they were inspired, our emotion-based reward features make universal, domain-independent evaluations of the significance of events for the agent's welfare, *i.e.*, its fitness. Therefore, our emotion-based critic relates to the *appraisal derivation model* proposed for appraisal-based computational models. Moreover, because the values of the several features are combined into a reward signal to guide the agent through learning, our RL decision-making mechanism corresponds to the *affect consequent model* of the emotional architecture that, at each step of time, takes a decision based on the outcome of previous appraisals (Marsella et al., 2010).

Finally, we note that in our approach, due to the nature of the evaluation and scenarios implemented, we are not concerned in explaining the current emotional state of the agent with labels such as *happy*, *sad* or *angry* corresponding to the *emotion/affect* component of Figure 3.3. Nevertheless, one can envisage a *labeling mechanism* that, at each time step, evaluates a point representing the agent's current emotional experience in the 4-dimensional space created by the actual values of the emotion-based reward features. Based on predefined *emotion profiles* or patterns associating a *label* with a specific point in the emotional space, one could find the emotion label which respective point was closest to the agent's current experience point. Such label would then be the one describing the agent's current emotional state.

[16] A similar perspective of the internal critic as an appraisal process is provided in (Marinier, 2008).

4.4.5 Comparison with Other Approaches

In Section 3.5 we discussed some works that, like our approach, are inspired by emotions for the creation of more robust and efficient autonomous agents. As we can see from Table 3.3, all the approaches rely either on a set of discrete emotions or positive/negative evaluations of the emotional state of the agent. This contrasts with our approach, since we propose a set of domain-independent reward features based on major dimensions of emotional appraisal which, together, create a multidimensional emotional experience space capable of generating a multitude of distinct emotional states.

While some systems use changes in the internal drives of the agent as the reward that guides the agent through learning (Gadanho and Hallam, 2001; Salichs and Malfaz, 2006), others (Ahn and Picard, 2006; Broekens et al., 2007), like our approach, use (combinations of) intrinsic and extrinsic reward. We note however that in our approach we do not modify the learning architecture as the intrinsic reward provided to the agent is considered a "normal", primary reward in any traditional RL algorithm (we refer to the discussion on Section 2.7.1). Also, as described earlier, we consider an optimization procedure to search for the optimal contribution of each reward feature for a particular set of environments. Finally, in our approach, instead of using predefined rules relating particular emotion states and action strategies (exploration vs. exploitation), emotions influence the choice of actions indirectly, *i.e.*, actions are chosen so as to maximize the *emotional benefit* of the current situation, as ascribed by the appraisal-based reward mechanism.

4.4.6 Plausibility and Universality of our Approach

We conclude this section with some important remarks regarding our approach. First, we *do not claim* the reward features proposed in this paper to be *universal*, *biologically plausible*, or *the only ones* reasonable to represent the discussed appraisal dimensions. Instead, we propose a *possible interpretation* for a set of low-level features that, within our framework, provide evaluations similar to those emerging from the respective appraisal dimensions in nature.[17] Like other approaches within IMRL (Niekum et al., 2010; Singh et al., 2009, 2010; Sorg et al., 2010a), we could have used observation/state data to perform the appraisal-like evaluation, but we wanted to provide features that, like emotions, provide general-purpose evaluations of events according to one's particular situation, independently of the particular domain being considered.

Finally, we also *do not claim* that emotion-based agents are able to cope with *all* kinds of challenges in *all* types of scenarios. Indeed, it is only expected that mechanisms that are inspired by biological systems are not perfect or flawless. As occurs in nature, *maladaptive behaviors* may lead to poor performances (see the discussion on Section 3.2.2). The results from our experimental procedure seem to support the claim that there may be "universal" parameter vectors that behave

[17] We propose *domain-independent* features that quantitatively evaluate some aspects of the agent's history of interaction with its environment, including statistical information about reward, action and state-value functions.

well across a given set of scenarios when compared to a fitness-based agent, despite not being "specialized" in any particular environment. However, the results from the experiment in Section 4.3.6 show that parameter vectors that were optimized for some scenarios lead to maladaptive behaviors by the agent when acting in a different environment. In fact, the optimization procedure for each scenario makes the agent value different aspects of its relationship with the environment, *i.e.*, it makes the agent to adapt to a specific habitat. This is to be expected in natural agents as well. Animals accustomed to living in one environment seldom thrive in environments which have major distinct characteristics.

4.5 Contributions

We now detail the contributions within this chapter resulting from our experiments in Section 4.3 and the discussion in Section 4.4. We also denote the implications of our approach for IMRL and the area of AC. Generally speaking, we advocate the idea that an emotion-based mechanism may be useful for both agent designers and for autonomous agents in general.

4.5.1 Implications for RL/IMRL

The main technical contribution for IMRL is a set of *four domain-independent emotion-based reward features*, namely *novelty*, *valence*, *goal relevance* and *control*, based on dimensions of *appraisal of the emotional significance of events*, commonly found in the psychology literature.

- The use of these reward features in the context of IMRL *alleviates the need for the agent designer to handcraft reward functions for a specific domain*, thus making the agents more *autonomous*;

- The advantage of our approach stems from the fact that, like emotions do to natural agents, emotion-based reward functions provide a general-purpose evaluative mechanism of the agent's history of interaction with its environment.

Furthermore, we demonstrate the potential of emotion-based learning through a set of foraging experiments targeted at evaluating how the proposed model can *mitigate perceptual limitations*.

- The scenarios also indicate a wider range of problem domains in which our features can improve the agents at being more *robust* and *flexible* by *requiring little tuning* of the reward functions for specific domains.

- Previous approaches within IMRL have also demonstrated the effectiveness of using intrinsic rewards to overcome computational limitations, but the solutions were usually based on frequency and domain-related features (Niekum et al., 2010; Singh et al., 2009, 2010; Sorg et al., 2010a). In this chapter we showed that emotions can be a very natural and efficient approach to reward design by mimicking the way humans and other animals evaluate and respond to events in their environment;

Our approach also *departs from reward shaping* techniques used with RL (Dorigo and Colombetti, 1994; Mataric, 1994; Ng et al., 1999; Randløv and Alstrøm, 1998). It is known that shaping-based algorithms require a great amount of knowledge about the domain and its effect is also temporary (Ng et al., 1999; Randløv and Alstrøm, 1998; Wiewiora, 2003). Also, in a large part of our experiments the agents do not have access to the necessary information to behave optimally and in consequence reward shaping techniques cannot be expected to overcome such limitations (Sorg et al., 2010a).

- By using combinations of intrinsic reward features evaluating the agent's history with its environment (unrelated directly with the intended task/fitness), our reward functions allow the agent to *mitigate its perceptual limitations* by actually changing the optimal policy, as opposed to reward shaping where the optimal policy is only approximated but is not modified, potentially leading to very poor performances;

- The results from our experiments show that our approach is also suitable to be used in non-stationary environments in which the elements change quickly or that due to limited perception provide ambiguous outcomes, reinforcing our claim of using emotions to *enhance the adaptive capabilities* of the learning agents.

4.5.2 Implications for AC

Our experiments and results reinforce other approaches within the field of AC that advocate the need for emotion-based mechanisms for autonomous agents capable of enhancing their perceptual and information-processing capabilities, similarly to what occurs in nature. However, as discussed earlier in Section 4.4.5, *our approach departs from previous works* in several ways:

- The proposed emotion-based reward features can be used in conjunction with any RL algorithm used to learn an optimal policy, *i.e.*, we do not modify the learning algorithms to include emotional information;

- We do not focus in a predefined set of basic emotions to influence learning. Our agents use a reward function that is the combination of the values of several appraisal dimension-based reward features. In this manner we are not limited by focusing on properties of a few modal emotions;

- We also do not use predefined rules to influence the agent's behavior based on emotions. Rather, the behavior emerges from the agent's interaction with its environment;

- Finally, we do not embed in the agent any domain knowledge relating the particular scenarios in which learning takes place. Our use of the appraisal-based features allows of their application in a variety of different scenarios.

4.6 Summary

In this chapter we analyzed the contributions of a novel mechanism for the design of intrinsic reward functions. We followed the ORP formulation within the IMRL framework in which the agent is provided with rewards motivating behaviors not directly related to its designer's objectives. For the nature of the rewards we proposed a set of four reward features each one evaluating a particular aspect of the agent's relationship with its environment. The proposed features relate to the kinds of evaluations that emotional appraisal dimensions provide to natural organisms. In a sense, each emotion-based reward feature makes the agent focus in aspects of the environment that are not themselves related to fitness enhancement but which motivate useful, advantageous behaviors for a particular set of environments.

Together with a feature providing the "standard" fitness-based reward to the agent, these reward features form a space of emotion-based reward functions. Each function has a distinct parameter vector that weights the importance of each feature differently. We tested the potential of our approach in a set of foraging experiments, each one providing particular challenges to the learning agent that somehow simulate the kinds of limitations biological agents face in nature. For each scenario, an optimization procedure searched for the particular configuration of reward features providing the best overall cumulative fitness for the agent in that scenario.

◇

In conclusion, in this chapter we showed that emotions, adapted to an intrinsic reward mechanism, endows IMRL agents to overcome some of their perceptual deficits and at the same time provide agent designers with a general-purpose, domain-independent reward mechanism capable of providing leverage in a variety of different situations. In the next chapter, we complement the work within this chapter to discover whether, from a set of possible reward mechanisms, the ones providing maximal performance to learning agents have dynamical and structural properties that can be related to emotions, specifically as prescribed by appraisal theories.

CHAPTER 5

Emerging Emotions within IMRL

In Section 3.2 we have provided theoretical support for the importance of emotions in biological organisms. Chapter 4 accompanied other works within AI advocating the positive impact of emotion-based mechanisms in augmenting the perceptual information of autonomous agents. In this chapter we address the question of whether emotions are indeed the *best* candidate for the agents' information processing mechanism. Appraisal theories of emotions investigated within psychology provide support for this hypothesis in biological agents, as discussed in Section 3.3. In this chapter we look for similar support in artificial systems. In Figure 5.1 we outline the methodology used in this chapter in order to answer the aforementioned question and highlight the contributions arising therefrom.

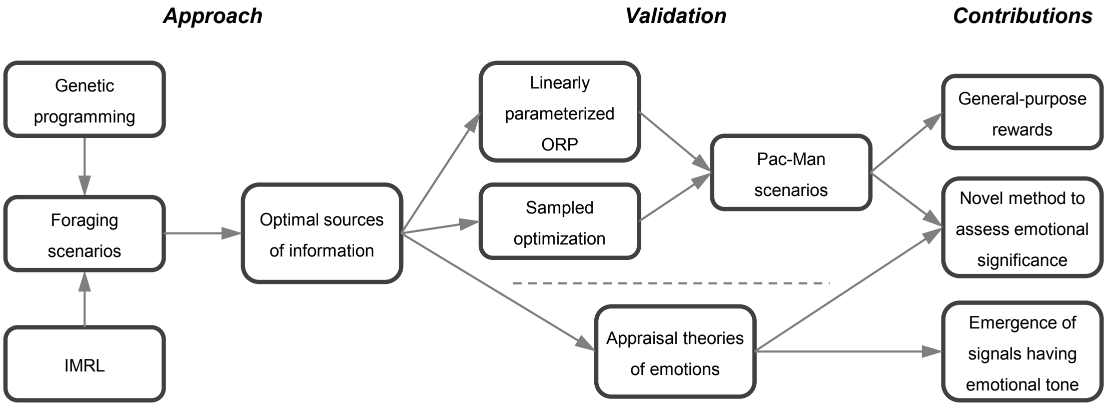

Approach
Validation
Contributions
Genetic programming
Foraging scenarios
IMRL
Optimal sources of information
Linearly parameterized ORP
Sampled optimization
Pac-Man scenarios
Appraisal theories of emotions
General-purpose rewards
Novel method to assess emotional significance
Emergence of signals having emotional tone

Figure 5.1: Outline of our approach for assessing the emergence of emotion-like signals within IMRL.

5.1 Introduction

As discussed in Section 3.2, research on psychology, neuroscience and other related areas established emotions as a powerful adaptive mechanism that influences cognitive and perceptual processing (Cardinal et al., 2002; Dawkins, 2000; Phelps and LeDoux, 2005). Emotions indirectly drive behaviors that lead individuals to achieve goals and satisfy needs. Damage to regions of the brain identified as responsible for emotional processing was shown to impact the human and animal ability to properly learn aversive stimuli, plan courses of action and, more generally, take decisions that are advantageous for their well-being (Bechara et al., 2000; Damasio, 1994; LeDoux, 2000).

On the other hand, the area of affective computing (AC) has also investigated the impact of emotional processing capabilities in the development of autonomous agents, as seen in Section 3.4. The embedded emotional processing mechanisms were shown to improve the performance of artificial agents as measured in several manners such as robustness, efficiency, believability and others (Marsella et al., 2010; Rumbell et al., 2011; Salichs and Malfaz, 2012; Sequeira et al., 2011b). As we have seen throughout this thesis, in very general terms, emotional architectures feature an *emotional processing* module that, *together with* the perceptual information acquired by the agent,

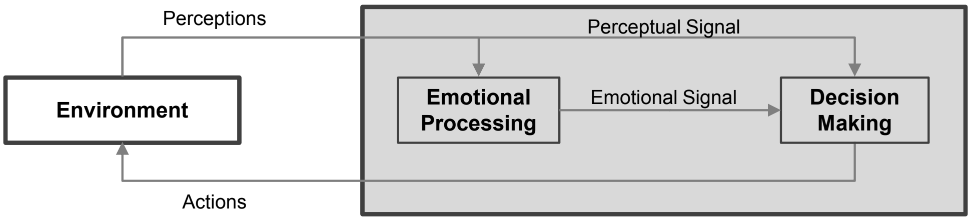


Figure 5.2: General architecture for an artificial agent with emotional processing. The *decision-making* process is driven by both *perceptual information* from the environment and also by some form of *emotional information*.

guides its decision process, as illustrated in Figure 5.2 (Becker-Asano and Wachsmuth, 2008; Dias and Paiva, 2005; El-Nasr et al., 2000; Marsella and Gratch, 2009; Marsella et al., 2010; Reilly and Bates, 1992; Sequeira et al., 2011a; Velásquez, 1997).

The *emotional signals* provided by such module "translate" information about the history of interaction of the agent with its environment that aid decision-making, complementing the perceptual information acquired through the agent's sensors. For example, our emotion-based learning architecture depicted in Figure 4.2, proposed in Chapter 4, fits this model. The decision-maker receives a reward signal as a linear combination of four emotion-based reward features—the emotional information— and a fitness-based reward feature— the external information.

Although one of the driving motivations for the use of emotional agent architectures is the creation of "better agents" (*e.g.*, agents able to successfully perform more complex tasks) one fundamental question remains mostly unaddressed in the literature: *in the search for information that may complement an agent's perceptual capabilities, are emotions the* best *candidate?*

In this chapter we contribute to this question by providing empirical evidence that emotion-like signals may arise as natural candidates when looking for sources of information to complement an agent's perceptual capabilities. Using an evolutionary approach, we show the *emergence* of domain-independent, general-purpose intrinsic sources of information to complement the (possibly limited) perception of artificial agents, thus endowing them with evolutionary advantages. We analyze the dynamical and structural properties of such sources of information and discover interesting connections with evaluative variables from appraisal theories of emotions proposed in the psychology literature (Ellsworth and Scherer, 2003; Frijda and Mesquita, 1998; Lazarus, 2001; Leventhal and Scherer, 1987; Roseman and Smith, 2001; Scherer, 2001). We thus contribute a computational parallel to the evidence observed in biological systems, where the organisms with the most complex emotional processing capabilities (humans) are arguably those most fit to their environment (LeDoux, 2000; Leventhal and Scherer, 1987; Oatley and Jenkins, 2006).

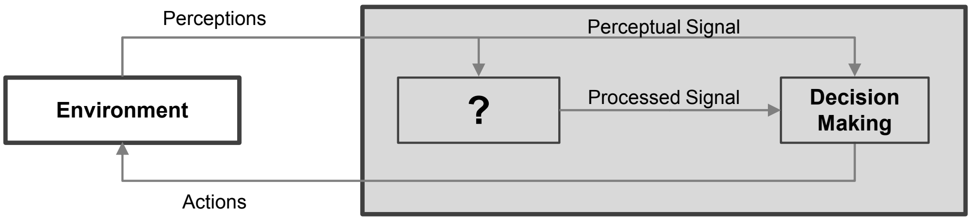


Figure 5.3: General architecture for an artificial agent which decisions are influenced by the perceptions from the environment and also other sources of information from some (unknown) processing mechanism.

5.2 Identification of Optimal Sources of Information

We begin our study in this chapter by addressing the following question: *which information is (potentially) most useful to complement the perceptual capabilities of an autonomous learning agent*? In other words, we abstract from any type of information processing mechanism for a learning agent and investigate possible sources of information that may most significantly impact the agent's performance, as illustrated in Figure 5.3.

5.2.1 Background

Before detailing our approach for the emergence of the optimal sources of information, we briefly describe the purpose behind the discipline of *evolutionary computation* (EC) and the use of *genetic programming* (GP) within the IMRL framework.

Evolutionary Computation

Within machine learning, the discipline of EC (or evolutionary modeling) is motivated by evolution in biological systems, which is known to be a successful adaptation process (Mitchell, 1997). The *neo-Darwinian* paradigm is a collection of evolutionary theories that stresses the importance of four physical processes, operating within *populations of species*, that determine their history throughout time (Fogel, 1994):

Reproduction, or the lack of it, allows for changes in the size of population throughout time. Through reproduction, part of an individual's genetic program is passed on to its descendants;

Mutation is the effect which allows for replication errors to occur when information is being transfered during reproduction;

Competition occurs within expanding populations with access to limited resources or by decreasing the amount of such resources;

Selection is a consequence of competition and determines which individuals within a population will pass on its genetic code to future generations.

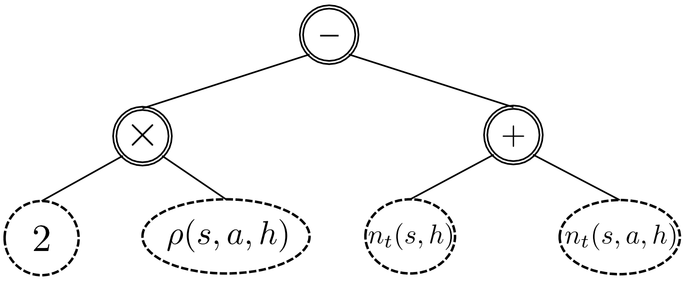


Figure 5.4: Example of a GP tree representing the reward function $r(s,a,h) = 2\rho(s,a,h) - (n_t(s,h) + n_t(s,a,h))$ (see text for details). Double circles represent operator nodes, dashed circles represent terminal nodes.

Genetic Programming within IMRL

As discussed in Section 2.5.1, the ORP is the formulated as finding a primary reward function $r(s,a,h)$ providing maximal utility as defined by a fitness function $f(h)$ in a set of environments of interest $\mathcal{E}$ (Bratman et al., 2012; Sorg et al., 2010a). In that respect, Niekum et al. (2010) proposed the use of GP in the context of IMRL as a possible approach to identify optimal rewards for RL agents.

In that particular work, GP was used to search for reward functions—represented by *genetic programs*—that combine different elements of the learning domain, such as the agent's position in the environment or its hunger status. For example, a reward function expressed by the program $2\rho(s,a,h) - (n_t(s,h) + n_t(s,a,h))$, represented in tree form in Figure 5.4, rewards the agent for fitness-inducing behaviors and punishes the agent as it becomes more and more "familiarized" with s and a. The GP algorithm finds interesting combinations of these informative elements to search for the ones providing maximal fitness in some environments.

5.2.2 Objectives

In this study we will again adopt the IMRL framework (Singh et al., 2009, 2010) introduced in Section 2.4.3. Recall that IMRL provides a principled manner to integrate multiple sources of information in the process of learning and decision-making of artificial agents (Singh et al., 2009). As such, it is a framework naturally suited to our investigation in this chapter. To address the aforestated question, we consider the foraging scenarios presented in Chapter 4, where the learning agent acts as a predator in the environment. As discussed therein, the perceptual limitations of the agent in the different environments pose challenges that directly impact its ability to capture its prey and, consequently, its *fitness*.

Our objective in this first set of experiments is to determine which reward functions—and, consequently, which sources of information—best complement the agent's perceptions. Because the different environments provide distinct challenges for our learning agents, there is no general heuristic for us to determine good reward functions for the set of environments of interest. As mentioned above, the GP approach automatically discovers interesting relations between *variables* (like $n_t(s)$, $n_t(s,a)$, and $r^{\mathcal{F}}(s,a)$ in the example of Fig. 5.4) that account for the agent-environment history of interaction. In fact, evolutionary computation algorithms have proved to be well suited to be applied to problems where heuristic solutions are not available or produce unsatisfactory results (Fogel, 1994).

In order to identify possible sources of useful information to complement the agent's perceptual limitations, we depart from a primitive population of agents, each endowed with a reward function evaluating information about different aspects of the agent's history of interaction with its environment. The fittest agents, *i.e.*, those with the greatest ability to capture preys, are used to successively improve the preceding population. Upon convergence, we identify the set of agents able to attain the largest degrees of fitness in each scenario. The analysis of the corresponding reward function provides the required information about which signals are potentially most useful to complement the perceptual capabilities of our IMRL agents.

5.2.3 Methodology

We now provide a detailed description of the setup and procedure used in this first experiment. As mentioned earlier, we will use the foraging scenarios defined in the experiments in Section 4.3, *i.e.*, the Hungry-Thirsty Scenario, Lairs Scenario, Moving Preys Scenario, Persistence Scenario, Prey Seasons Scenario and Poisoned Prey Scenario. Also, we model our RL agents as learning in MDPs using prioritized sweeping (Moore and Atkeson, 1993) exactly as described in Section 4.3.2. Recapitulating, in our experiments, prioritized sweeping updates the Q-value of up to 10 state-action pairs in each iteration, using a learning rate of $\alpha = 0.3$ and a discount factor of $\gamma = 0.9$. The agent follows an ε-greedy exploration strategy with a decaying exploration parameter $\varepsilon_t = \lambda_\epsilon^t$, with $\lambda_\epsilon = 0.9999$.

Evolutionary Procedure

In general terms, GP aims to find a *program* that maximizes some measure of *fitness* (Koza, 1992). Programs are represented as syntax trees, where nodes correspond to either *operators* or *terminal nodes* (see Figure 5.4). Operators are selected from a set $\mathcal{O}$ of possible operators, and its arguments are represented as their descendants in the tree. Non-operator (terminal) nodes are selected from a set $\mathcal{T}$ of possible terminal nodes, and represent either numerical variables or constants.

GP iteratively explores possible solutions by maintaining a population of candidate programs, referred to as *hypothesis*, producing new generations of programs by means of several mechanisms that relate to the evolutionary physical processes described above. Specifically, within GP, *reproduction* is achieved by means of the *crossover* function that randomly replaces some subtree (a node and all of its descendants) of a parent program by another subtree from another parent. The

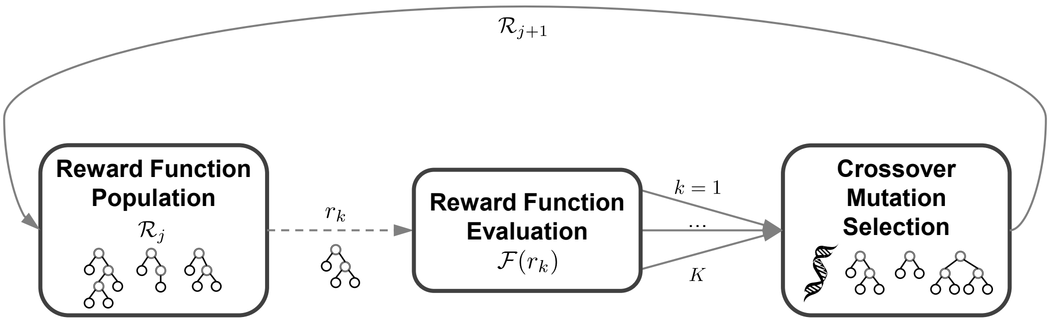

$\mathcal{R}_{j+1}$
Reward Function Population
$\mathcal{R}_j$
r_k
Reward Function Evaluation
$\mathcal{F}(r_k)$
$k = 1$
...
K
Crossover Mutation Selection

Figure 5.5: The GP approach to the ORP, as proposed in (Niekum et al., 2010). Populations of agents, each with one reward function, are evaluated according to some fitness function and evolve according to crossover, mutation and selection. See text for details.

mutation operator replaces some node by another randomly selected one.[1]

GP alleviates the need for specifying an explicit *parameterization* for the space of rewards, *e.g.*, as occurs in linearly parameterized ORP. Instead, we implicitly define the space of possible reward functions $\mathcal{R}$ as a population of genetic programs by specifying a set of *operators* and *terminal nodes*. In our case, we use as terminal nodes quantities that evaluate some aspects of the history h of interaction of the agent with its environment. The operators then combine these quantities, constructing richer, more complex and potentially more informative signals throughout "evolution". By following such offline optimization procedure we automatically aim at weak mitigation, as discussed in Section 2.4.4. We note however that in this case we are interested in the quality of the resulting GP expressions used as reward functions and that the evolutionary procedure is a requirement to discover interesting combinations of the basic information elements.

Figure 5.5 outlines the learning scheme for GP within IMRL. At each generation j, each reward function $r_k, k = 1, \ldots, K$ from a population of reward functions $\mathcal{R}_j$ of size K is evaluated according to the fitness function $\mathcal{F}(r_k)$ as it was defined in (2.9) and the scheme delineated in Figure 2.8. When all the reward functions have been evaluated, the evolutionary procedure takes place by applying the mutation and crossover operators defined above and applying selection over the population in order to produce the new generation of reward functions, corresponding to population $\mathcal{R}_{j+1}$.[2]

Genetic Programming Parameterization

Recall from Section 2.4.3 that $\rho_t(s, a, h)$ is an "external" evaluative signal that rewards the agent according to the increase/decrease of fitness caused by executing each action a in each state s during history h. Recall also the definition given by (2.7) of the fitness-based reward function $r^{\mathcal{F}}$ that uses solely this external evaluative signal to provide the reward to the agent. In our experiments, we used a terminal set $\mathcal{T} = \mathcal{C} \cup \mathcal{V}$, with $\mathcal{C}$ corresponding to the set of *constants*,

[1] More details can be found in (Koza, 1992, 1994).

[2] The first generation, corresponding to the population of reward functions $\mathcal{R}_0$, is randomly generated.

$\mathcal{C} = \{0, 1, 2, 3, 5\}$, and $\mathcal{V}$ to the set of *variables*, $\mathcal{V} = \{r_{za}, n_z, n_{za}, v_z, q_{za}, d_z, e_{za}, p_{zaz'}\}$, where

- $r_{za} = \rho(z, a, h)$ is the agent's fitness-based reward for performing action a after observing z. It is a function of z, a, and the agent's history h.

- $n_z = n(z, h)$ is the number of times that z was observed so far during history h.

- $n_{za} = n(z, a, h)$ is the number of times the agent executed action a after observing z during history h.

- $v_z = V^{\mathcal{F}}(z, h)$ is the value function associated with the fitness-based reward function estimate $\hat{r}^{\mathcal{F}}(z, a, h)$. It is a function of z and the agent's history h.

- $q_{za} = Q^{\mathcal{F}}(z, a, h)$ is the Q-function associated with the fitness-based reward function estimate $\hat{r}^{\mathcal{F}}(z, a, h)$. It is a function of z, a, and the agent's history h.

- $d_z = \hat{d}(z)$ corresponds to an estimate of the number of actions needed to reach a *goal* after observing z. Goals correspond to those observations that maximize $\hat{r}^{\mathcal{F}}$. It is a function of z and the agent's history h.

- $e_{za} = \mathbb{E}\left[\Delta Q^{\mathcal{F}}(z, a, h)\right]$ is the expected *Bellman error* associated with $Q^{\mathcal{F}}$, where $\Delta Q^{\mathcal{F}}(z, a, h)$ is given by (4.3). This variable is a function of z, a, and the agent's history h.

- $p_{zaz'} = \hat{\mathsf{P}}(z' \mid z, a)$ corresponds to the estimated probability of observing z' when executing action a after observing z. Since the learning algorithm used by the agent averages the perceived reward function, $p_{zaz'}$ is actually equivalent to

  $$\mathbb{E}\left[\hat{\mathsf{P}}(z' \mid z, a)\right] = \sum_{z' \in \mathcal{Z}} \hat{\mathsf{P}}(z' \mid z, a)\mathbb{P}\left[z_{t+1} = z' \mid z_t = z, a_t = a\right].$$

  The latter is a function of z, a, and the agent's history h.

The variables above include all elements stored and/or computed by the learning agent, and therefore summarize the agent's history of interaction with its environment h. As for the operators used by the GP algorithm, we considered the set $\mathcal{O} = \{+, -, \times, /, \sqrt{\cdot}, \exp, \log\}$.

Computing the Optimal Reward Functions

We generate a total of 50 independent initial populations, each containing $K = 100$ elements, and ran the evolutionary procedure described earlier for 50 generations for each population. For the *selection* mechanism, we use a *steady-state procedure* that, in each generation, maintains the 10 most fit elements and generates 10 new random elements. The remaining 80 elements are generated by pairing elements of the previous population (through *reproduction*) according to a *rank selection* that chooses parents with a probability which is proportional to their fitness.

Table 5.1: Mean cumulative fitness and evolved reward functions for each scenario. The results correspond to averages over 200 independent Monte-Carlo trials.

Scenario	**Reward Function**	**Mean Fitness**
Hungry-Thirsty Scenario	$r = q_{za} - v_z - 2$	$10,252.1 \pm 6,773.1$
	$r = r_{za}$	$7,129.4 \pm 6,603.2$
Lairs Scenario	$r = q_{za} - v_z$	$8,136.5 \pm 1,457.5$
	$r = r_{za}$	$7,478.3 \pm 791.6$
Moving Preys Scenario	$r = -n_z^2$	$2,452.6 \pm 45.4$
	$r = r_{za}$	381.1 ± 18.0
Persistence Scenario	$r = q_{za} - v_z$	$1,877.4 \pm 11.6$
	$r = r_{za}$	136.1 ± 1.5
Prey Seasons Scenario	$r = r_{za} + q_{za} - p_{zaz'}$	$6,426.1 \pm 149.1$
	$r = r_{za}$	$4,936.4 \pm 1,900.9$
Poisoned Prey Scenario	$r = 5r_{za} - q_{za}$	$5,233.7 \pm 715.3$
	$r = r_{za}$	$1,284.3 \pm 4.1$

Also, as with the experiments of Chapter 4, in order to estimate the value $\mathcal{F}(r)$ for each reward function, we run $N = 200$ independent Monte-Carlo trials of $100,000$ time-steps each, where in each trial we simulate an RL agent driven by reward r in an environment selected randomly from the corresponding environment set, $\mathcal{E}$ according to the distribution $P(\mathcal{E})$.

When all 50 generations are run for a population, the best reward functions, *i.e.*, those providing the highest fitness to the agent according to $\mathcal{F}(r)$ are parsed for sub-expressions that may have no effect on the fitness. Such parsing consists in running an additional set of 200 simulations for each possible sub-expression of a top reward function. The different sub-expressions are evaluated in terms of their impact in the fitness, and useless sub-expressions are then eliminated from the corresponding GP tree.

5.2.4 Results

The results of the GP experiment are summarized in Table 5.1. We present the average fitness obtained by the best agent selected using GP in each of the test scenarios, as well as a simplified expression for the corresponding *evolved reward function* as described above. As a straightforward baseline for comparison, we also present the fitness obtained by an agent driven by the fitness-based reward function (for which $r = r_{za}$), corresponding to an agent receiving only fitness-based reward.

One first observation is that, in all scenarios, the best evolved reward function clearly outperforms the fitness-based reward function. Our results are in accordance with findings in previous works on the advantages of allowing additional sources of information to guide the agent decision-making (Bratman et al., 2012; Sequeira et al., 2011b; Singh et al., 2010; Sorg et al., 2010a).

Our results also confirm previous findings on the usefulness of an evolutionary approach to search for optimal reward functions (Niekum et al., 2010). There is, however, one key difference between our approach and that in (Niekum et al., 2010): we provide the evolutionary approach with *domain-independent* sources of information relating to the agent's history of interaction with the environment, which were detailed in Section 5.2.3. Therefore, we expect that the reward functions thus evolved can be applied in domains other than those used in this experiment which were described in Section 4.3.

5.2.5 Discussion

We recall that the goal of our first experiment was to identify possible sources of information that could improve the agent's performance if taken into consideration in the process of decision-making. In that respect we used GP to discover relations between genetic variables that provided maximal fitness in some foraging scenarios. Given the parsing process used to simplify the best reward functions' expressions evolved through GP, each reward function indicated in Table 5.1 can be interpreted as a possible "signal" that can drive the agent's decision process, allowing it to maximize its fitness.

Discarding additive and multiplicative constants, we can distill from Table 5.1 a set of five signals, $\Phi = \{\phi_{\mathfrak{fit}}, \phi_{\mathfrak{adv}}, \phi_{\mathfrak{rel}}, \phi_{\mathfrak{prd}}, \phi_{\mathfrak{frq}}\}$, given by

- $\phi_{\mathfrak{fit}} = r_{za}$ corresponds to the agent's estimate of the *fitness-based reward function*. It evaluates the immediate impact on *fitness* associated with performing action a after observing z.

- $\phi_{\mathfrak{rel}} = q_{za}$ corresponds to the estimated Q-function associated with r_{za}. This function assesses the *relevance* of executing action a after observing z in terms of long-term impact on fitness, corresponding to the long-run counterpart to $\phi_{\mathfrak{fit}}$.

- $\phi_{\mathfrak{adv}} = q_{za} - v_z$ corresponds to the estimated *advantage* function associated with r_{za} (Baird, 1993). This function evaluates how good action a is when executed after observing z relatively to the best action (thus its *advantage*). While $\phi_{\mathfrak{rel}}$ evaluates the absolute value of actions, $\phi_{\mathfrak{adv}}$ evaluates their relative value.

- $\phi_{\mathfrak{prd}} = p_{zaz'}$ corresponds to the agent's estimate of the transition probabilities. As discussed in Section 5.2.3, it provides a measure of how *predictable* the observation at time $t+1$ is given that the agent performed action a after observing z.

- Finally, $\phi_{\mathfrak{frq}} = -n_z^2$ provides a (negative) measure of how *frequently* the agent observed z.

The signals ϕ_k defined above were obtained by combining the variables in $\mathcal{V}$ using different GP operators. Each such signal is a function mapping observation-action-history triplets to a real-value, and will henceforth be used as a *source of information* guiding the decision process of

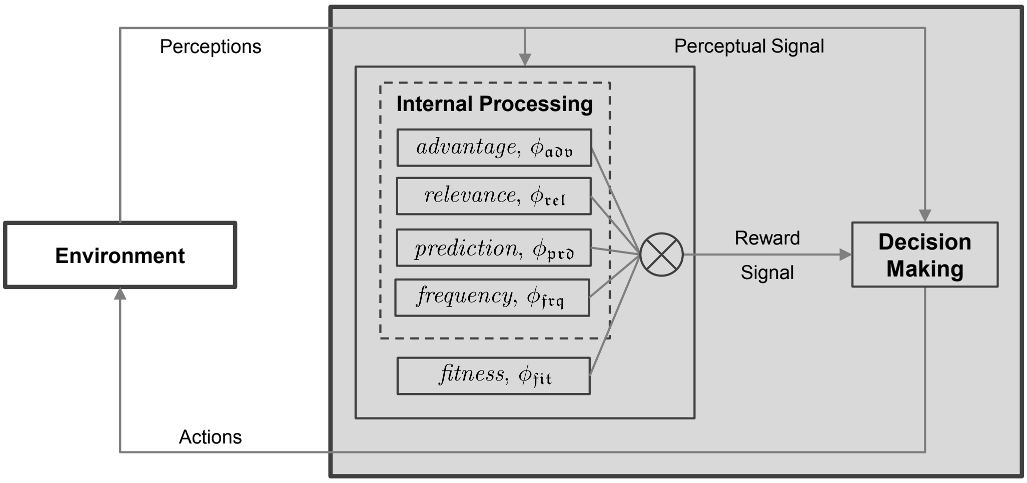

Perceptions
Perceptual Signal
Internal Processing
advantage, $\phi_{\mathfrak{adv}}$
relevance, $\phi_{\mathfrak{rel}}$
prediction, $\phi_{\mathfrak{prd}}$
frequency, $\phi_{\mathfrak{frq}}$
fitness, $\phi_{\mathfrak{fit}}$
Environment
Reward Signal
Decision Making
Actions

Figure 5.6: Agent framework resulting from the GP procedure, in which each signal ϕ_k can be used as a reward feature contributing to the overall reward provided to the agent. We note that the *fitness* component is directly perceived from the environment and thus is not internally processed.

the agent. The abstract agent architecture depicted in Figure 5.3 can then be realized by using the sources of information discovered in this experiment. The resulting model is illustrated in Figure 5.6.

5.3 Validation of Identified Sources

Section 5.2 focused on identifying general-purpose sources of information that can guide the decision process of an IMRL agent while positively impacting its performance. These different sources of information emerged from the interaction of agents with several different environments and, as such, should be applicable in different scenarios.

This section investigates whether this is indeed so, *i.e.*, whether the sources of information identified earlier can be used in a broader range of scenarios than those considered so far. In particular,

- We show that the set of "signals" $\Phi = \{\phi_{\mathfrak{fit}}, \phi_{\mathfrak{adv}}, \phi_{\mathfrak{rel}}, \phi_{\mathfrak{prd}}, \phi_{\mathfrak{frq}}\}$ can be used to construct reward functions other than those in Table 5.1, establishing them as *general-purpose* sources of information for IMRL agents;

- We show that it is generally advantageous to indeed combine one or more of the signals in Φ to formulate the reward function driving our IMRL agents, thus establishing them as *universal* sources of information for IMRL agents.

As we have seen, the agent architecture considered in this section specializes that in Figure 5.3, specifically accounting for the sources of information in Φ (see Figure 5.6). In this architecture, the reward signal driving the decision-making process is a *linear combination* of the different signals

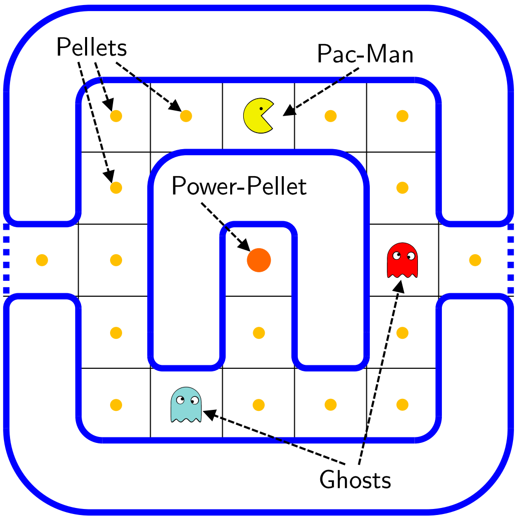


Figure 5.7: Structure and elements of the Pac-Man environment used in the second set of experiments. Each square represents a possible location for the agent. Bold lines represent walls which the agent cannot transpose.

in Φ. As such, this model can be used within the linearly parameterized approach to the ORP described in Section 2.5.1.

5.3.1 Objectives

To validate the applicability of the signals emerged by means of the GP procedure identified in Section 5.2.5, we conducted two sets of experiments. We start by performing an initial validation, again resorting to the foraging scenarios described in Section 4.3. This first set of experiments is essentially equivalent to those in Section 5.2, now using the linear formulation of the ORP instead of the GP approach. The goal is merely to replicate the results reported in Table 5.1 with the linear ORP formulation.

The second set of experiments is the central purpose of this section, and aims at providing a more extensive validation of the applicability of the signals in Φ as useful sources of information to complement the agent's perceptions. To this purpose, we consider several significantly harder domains inspired by the traditional computer game of Pac-Man. We use a total of four scenarios, each with different goals and posing different challenges to the agent.

5.3.2 Pac-Man Scenarios

Figure 5.7 illustrates the structure of the environment and the elements used in the Pac-Man scenarios. In this set of scenarios, we model our agent as the Pac-Man trying to eat as much *pellets* as possible throughout time while avoiding encounters with the *ghosts*.

Power-Pellet Scenario: The configuration of the environment for this scenario is outlined in Figure 5.7. In this scenario, our agent co-exists in the environment with two *ghosts*, the Smart Ghost and the Keeper Ghost. One Power-Pellet is available per episode, located in the central cell of the environment, as depicted. The Power-Pellet is *consumed* and removed from the environment as soon as Pac-Man reaches its position, contributing to its fitness with a value of $\rho = 0.8$.

In each episode, Pac-Man departs from the position depicted in Figure 5.7 and the two ghosts depart from the central position. As with the original game of Pac-Man, the Power-Pellet provides special powers to the agent. When one of the ghosts and Pac-Man stand in the same cell, the ghost *captures* Pac-Man if the latter has not yet consumed the Power-Pellet, and *is consumed* by Pac-Man otherwise. The episode terminates as soon as one of the following conditions is met:

- Pac-Man consumes *both* ghosts, which contributes to its fitness with $\rho = 1$;
- Pac-Man is "captured" by one of the ghosts, contributing to its fitness with $\rho = -1$;
- 20 time-steps have elapsed after the Power-Pellet was consumed.

When an episode terminates, the environment is reset to its initial configuration and a new episode starts. ◊

Eat-all-Pellets Scenario: In this scenario, our Pac-Man agent co-exists with only the Smart Ghost. The environment has available a total of 20 pellets, one in each cell, as depicted in Figure 5.7. The pellets are consumed and removed from the environment whenever Pac-Man visits the corresponding cell. However, consuming a pellet does not alter the agent's fitness. The Pac-Man and ghost initial configuration is the same as in the Power-Pellet Scenario. However, in this scenario, consuming the Power-Pellet contributes to the fitness of the agent with a value of $\rho = 0.5$, but does not enable Pac-Man to consume the ghost. Instead, episodes terminate as soon as one of the following conditions occurs:

- Pac-Man consumes *all* 20 pellets, which contributes to its fitness with $\rho = 1$;
- Pac-Man is captured 3 times by the ghost before all pellets are consumed, which contributes to its fitness with a value of $\rho = -0.5$.

When the ghost captures Pac-Man, their positions are reset. When an episode terminates, the whole environment (including existing pellets) is reset to its initial configuration and a new episode starts. ◊

Rewarding-Pellets Scenario: The general configuration for this scenario is outlined in Figure 5.7. In this scenario, our Pac-Man agent co-exists with both the Smart Ghost and the Keeper Ghost. The environment has available a total of 20 pellets (one in each cell), which are consumed and removed from the environment whenever Pac-Man visits the corresponding cell. Each consumed pellet contributes to the fitness of the agent with a value of $\rho = 0.1$, in the case of a regular pellet, or $\rho = 0.8$, in the case of the Power-Pellet. The Pac-Man and ghost initial configuration is the same as in the previous scenarios. As with the Eat-all-Pellets Scenario, consuming the Power-Pellet does not enable Pac-Man to consume the ghosts. Instead, an episode terminates as soon as one of the following conditions is met:

- Pac-Man consumes *all* 20 pellets, which contributes to its fitness with $\rho = 1$;
- Pac-Man is captured by a ghost before all pellets are consumed, contributing with a value of $\rho = -1$ to its fitness.

When an episode terminates, the whole environment (including existing pellets) is reset to its initial configuration and a new episode starts. ◊

Pac-Man Scenario: This scenario is a combination of all previous scenarios, and is the one closest to the original game of Pac-Man. In this scenario, our agent again corresponds to the Pac-Man, and co-exists with only the Smart Ghost. The environment has available a total of 20 pellets (one in each cell), which are consumed and removed from the environment whenever Pac-Man visits the corresponding cell. As occurred in the Eat-all-Pellets Scenario, eating a pellet does not contribute to the agent's fitness.

The Pac-Man and ghost initial configuration is the same as in the Power-Pellet Scenario. In this scenario, consuming the Power-Pellet does not contribute to the fitness of the agent, but does enable Pac-Man to consume the ghost if collocated with it. An episode terminates as soon as one of the following conditions is met:

- Pac-Man consumes *all* 20 pellets, which contributes to its fitness with $\rho = 1$;
- Pac-Man is captured 3 times before all pellets are consumed (with no impact in fitness).

When the Pac-Man is captured by the ghost, its fitness is decreased by a value of $\rho = -0.1$, and their positions are reset. When an episode terminates, the whole environment (including existing pellets) is reset to its initial configuration and a new episode starts. ◊

5.3.3 Methodology

As we have seen, the linear formulation of the ORP that we adopt in this section has been explored in the IMRL literature by different authors (Singh et al., 2009, 2010; Sorg et al., 2010a,b). It was also used in Section 4.3 to validate our approach for emotion-based design.

In the context of this section, the linear formulation has two appealing properties:

- First, by comparing the parameters associated with each source of information, we are able to perceive their relative importance in each scenario: signals for which the corresponding parameter has only a residual value have little weight in the agent's reward and, consequently, in the decision process of the agent. This is useful to assess whether the sources of information in Φ indeed provide useful information to guide the agent's decisions;

- A second appealing aspect of this formulation is that it allows a relatively general agent architecture, where all the signals in Φ are provided to the agent. The particular environment with which the agent interacts will condition *how* the agent uses these different signals, paralleling the evolutionary process by which natural organisms are conditioned to (see the discussion in Section 2.7.1.

Agent Description

We refer to Section 4.3 for a detailed description of the foraging scenarios, including each environment's dynamics and the description of the agent's actions.

In the Pac-Man scenarios, our agent has 4 actions available, $\mathcal{A} = \{Up, Down, Left, Right\}$, that deterministically move it in the corresponding direction. The regions delimited by solid blue lines in Figures 5.7 correspond to obstacles that cannot be traversed. There is a "magic door" in the environments that move the Pac-Man (and also the ghosts) from one side to the other. Moving $Right/Left$ in the leftmost/rightmost cell moves Pac-Man to the rightmost/leftmost cell, respectively.

The motion of the ghosts respects the same restrictions as the motion of Pac-Man (*e.g.*, they cannot traverse obstacles). At every time-step, the Smart Ghost moves towards the Pac-Man with probability 0.6. However, once the Pac-Man consumes the Power-Pellet, the Smart Ghost instead moves *away* from the Pac-Man (in the Power-Pellet Scenario and Pac-Man Scenario). With a probability 0.4 it moves in a random direction. The Keeper Ghost moves towards the Smart Ghost with probability 0.5 and towards one of the bottommost cells otherwise.[3]

In each scenario, our agent is modeled as a MDP whose state-dynamics follow from the description in Section 5.3.2. In this model, the Pac-Man agent is able to observe, at each time-step:

- Its current position $(x : y)$ in the environment;
- Whether a ghost exists in the same corridor as the agent, in each of the 4 possible directions;
- Whether a pellet exists in the same corridor as the agent, in each of the 4 possible directions;
- When co-located with the Smart Ghost;
- When co-located with the Keeper Ghost;
- When co-located with a pellet;

[3] The Keeper Ghost, when present, makes it difficult for Pac-Man to reach the central cell, essential for the completion of most scenarios.

- When co-located with the Power-Pellet.

As with the previous experiments, the experiments in this section use RL agents using prioritized sweeping (Moore and Atkeson, 1993) to learn a memoryless policy that treats observations as states, *i.e.*, thus ignoring partial observability by the agent. The algorithm's parameterization can be found in Section 5.2.3.

Computing Agent Fitness

Recall from Section 2.5.1 that the linearly parameterized approach to the ORP proposes the definition of a set of real-valued *reward features* to build the reward function space $\mathcal{R}$ in order to search for the optimal reward function r^*. In our experiments, we use as reward features the set of signals $\Phi = \{\phi_{\mathfrak{frq}}, \phi_{\mathfrak{rel}}, \phi_{\mathfrak{prd}}, \phi_{\mathfrak{adv}}, \phi_{\mathfrak{fit}}\}$ identified Section 5.2.5. By optimizing the associated parameters—henceforth denoted $\boldsymbol{\theta} = \{\theta_{\mathfrak{frq}}, \theta_{\mathfrak{rel}}, \theta_{\mathfrak{prd}}, \theta_{\mathfrak{adv}}, \theta_{\mathfrak{fit}}\}$—in a broad set of scenarios, we can analyze the relative importance of the different signals in each scenario and draw conclusions on their *general usefulness* and *applicability*.

Recall also that solving the ORP involves the discovery of $\boldsymbol{\theta}^*$, *i.e.*, the optimal parameter vector providing maximal fitness in a set of environments of interest $\mathcal{E}$ according to $\mathcal{F}(r(\boldsymbol{\theta}))$, as defined by (2.12). As discussed in Section 2.5.1, this optimization can be conducted using different techniques (Bratman et al., 2012; Singh et al., 2010; Sorg et al., 2010a,b). In the previous set of experiments, we used GP to search for the optimal reward function. In these experiments we follow the linearly parameterized approach to ORP and our focus is on the particular optimal parameter combination $\boldsymbol{\theta}^*$ associated with the emerged reward features. We therefore adopt the same sampling procedure and offline optimization method used in our experiments in Section 4.3.2. We generate a total of $M = 14,003$ parameter vectors $\boldsymbol{\theta}_m \in [-1, 1]^5, m = 1, \ldots, M$, such that $\|\boldsymbol{\theta}_m\|_1 = 1$. We measure the agent's fitness during a particular history interaction $f(h_t^i)$ as the cumulative fitness-based reward received during h_t^i as defined by (2.8).

As with the previous experiments, we evaluate $\mathcal{F}(r(\boldsymbol{\theta}))$ by running $N = 200$ independent Monte-Carlo trials of $100,000$ time-steps each, where in each trial we simulate an RL agent driven by reward $r(\boldsymbol{\theta})$ in an environment $e^i \in \mathcal{E}$ selected randomly from the corresponding environment set $\mathcal{E}$, according to $p_E(e^i)$. In our experiments we used a uniform distribution, *i.e.*, $p_E(e^i) = \frac{1}{|\mathcal{E}|}$. $\mathcal{F}(r(\boldsymbol{\theta}))$ is approximated as indicated by (2.9).

5.3.4 Results

Foraging Scenarios

The results corresponding to the foraging scenarios are summarized in Table 5.2. Comparing these results with those in Table 5.1, it is clear that the linear architecture is able to attain similar

Table 5.2: Mean cumulative fitness and obtained parameter vectors for each foraging scenario. The fitness results correspond to averages over 200 independent Monte-Carlo trials.

Scenario	**Parameter Vector**						**Mean Fitness**
	$\boldsymbol{\theta} = [$	$\theta_{\mathfrak{frq}},$	$\theta_{\mathfrak{rel}},$	$\theta_{\mathfrak{prd}},$	$\theta_{\mathfrak{adv}},$	$\theta_{\mathfrak{fit}}\ ]^\top$	
Hungry-Thirsty Scenario	$\boldsymbol{\theta}^* = [$	$0.0,$	$0.0,$	$-0.2,$	$0.5,$	$0.3\ ]^\top$	$10,718.8 \pm 7,226.5$
Lairs Scenario	$\boldsymbol{\theta}^* = [$	$0.1,$	$0.0,$	$0.3,$	$0.4,$	$0.2\ ]^\top$	$9,598.6 \pm 1,543.4$
Moving Preys Scenario	$\boldsymbol{\theta}^* = [$	$1.0,$	$0.0,$	$0.0,$	$0.0,$	$0.0\ ]^\top$	$2,394.5 \pm 48.8$
Persistence Scenario	$\boldsymbol{\theta}^* = [$	$-0.1,$	$0.0,$	$0.0,$	$0.5,$	$0.4\ ]^\top$	$1,879.9 \pm 10.7$
Prey Seasons Scenario	$\boldsymbol{\theta}^* = [$	$0.0,$	$0.2,$	$-0.5,$	$0.1,$	$0.2\ ]^\top$	$6,473.6 \pm 136.5$
Poisoned Prey Scenario	$\boldsymbol{\theta}^* = [$	$0.0,$	$0.2,$	$0.0,$	$-0.1,$	$0.7\ ]^\top$	$5,297.9 \pm 529.4$

Table 5.3: Mean cumulative fitness and parameter vector determined in each of the Pac-Man scenarios. The results correspond to averages over 200 independent Monte-Carlo trials.

Scenario	**Parameter vector**						**Mean Fitness**
	$\boldsymbol{\theta} = [$	$\theta_{\mathfrak{frq}},$	$\theta_{\mathfrak{rel}},$	$\theta_{\mathfrak{prd}},$	$\theta_{\mathfrak{adv}},$	$\theta_{\mathfrak{fit}}]^\top$	
Power-Pellet Scenario	$\boldsymbol{\theta}^* = [$	$-0.2,$	$0.2,$	$0.1,$	$0.5,$	$0.0]^\top$	$1,265.0 \pm 424.9$
	$\boldsymbol{\theta}^{\text{fit}} = [$	$0.0,$	$0.0,$	$0.0,$	$0.0,$	$1.0]^\top$	$-1,902.6 \pm 183.5$
Eat-all-Pellets Scenario	$\boldsymbol{\theta}^* = [$	$0.1,$	$0.1,$	$0.1,$	$0.6,$	$0.1]^\top$	$1,005.5 \pm 207.1$
	$\boldsymbol{\theta}^{\text{fit}} = [$	$0.0,$	$0.0,$	$0.0,$	$0.0,$	$1.0]^\top$	25.3 ± 215.5
Rewarding-Pellets Scenario	$\boldsymbol{\theta}^* = [$	$0.5,$	$0.0,$	$0.1,$	$0.2,$	$0.2]^\top$	$4,343.7 \pm 210.1$
	$\boldsymbol{\theta}^{\text{fit}} = [$	$0.0,$	$0.0,$	$0.0,$	$0.0,$	$1.0]^\top$	$3,060.8 \pm 208.6$
Pac-Man Scenario	$\boldsymbol{\theta}^* = [$	$0.2,$	$0.1,$	$0.2,$	$0.2,$	$0.3]^\top$	$1,223.6 \pm 117.5$
	$\boldsymbol{\theta}^{\text{fit}} = [$	$0.0,$	$0.0,$	$0.0,$	$0.0,$	$1.0]^\top$	862.2 ± 95.7

fitness values by combining the reward features in Φ. In some scenarios, the obtained fitness is, inclusively, slightly superior, although overall the differences are not statistically significant.

It is also interesting to compare the weights θ_k associated with each of the signals ϕ_k in Φ, denoting that the resulting signals generally match those found in Section 5.2.4. As such, the results from this experiment thus validate the signals in Φ as those responsible for the performance reported in Section 5.2.

Pac-Man Scenarios

The results of the Pac-Man experiments are summarized in Table 5.3. We present the average fitness obtained by the agent using the optimal reward function $r(\boldsymbol{\theta}^*)$ in each of the test scenarios, as well as the corresponding parameter vector $\boldsymbol{\theta}^*$. As a baseline for comparison, we also present the fitness obtained by an agent driven by the fitness-based reward function, corresponding to the parameter vector $\boldsymbol{\theta}^{\text{fit}} = [0, 0, 0, 0, 1]^\top$.

As with the scenarios in Section 5.2, our results show that the Pac-Man agents that use only the fitness-based reward function are clearly inferior to those agents that use additional sources of information. This settles the main issue addressed in this section, on the general applicability of the sources of information identified in Section 5.2.5: not only using these signals is advantageous in terms of the performance of the agent but also, as seen from the weights in Table 5.3, these signals are generally more informative than the fitness-based reward function in most scenarios.

5.3.5 Discussion

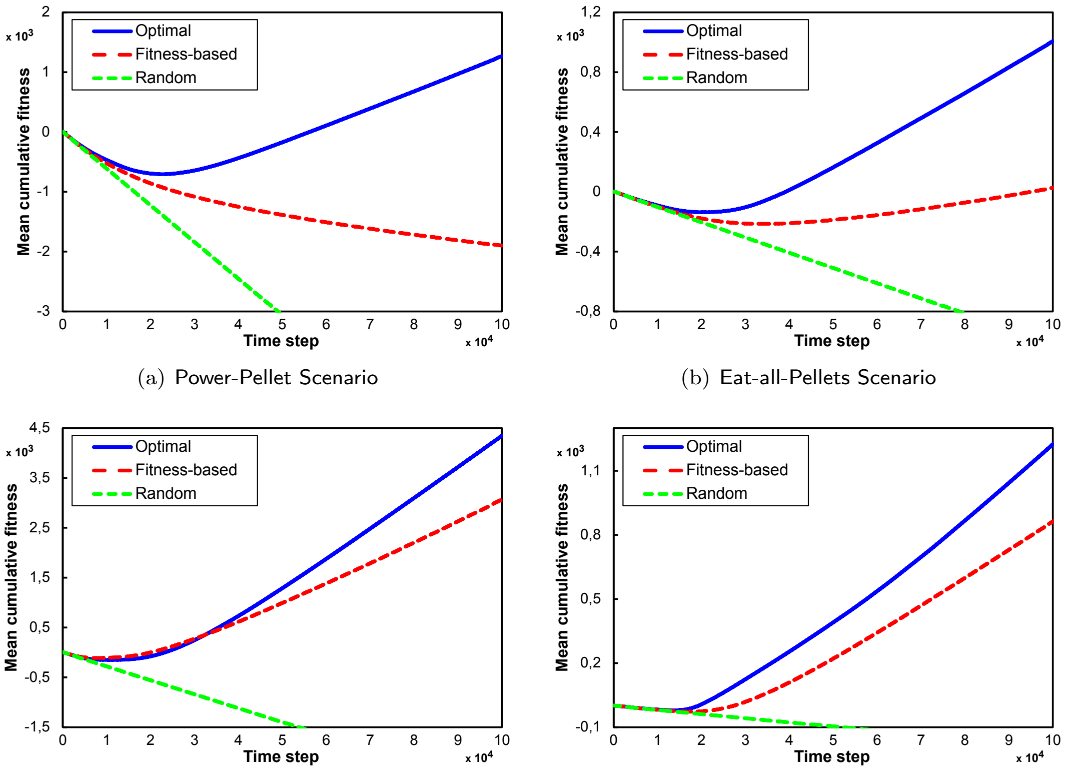
We conclude this section by analyzing in greater detail some of the main challenges that the Pac-Man agent must face in the scenarios used in the experiments. These challenges allow us to evaluate the benefit of the reward features in Φ in guiding the agent.[4] To aid in this analysis, Figure 5.8 shows the evolution of fitness throughout time, where we also included the performance of a "random" agent receiving no reward, corresponding to the parameter vector $\boldsymbol{\theta}^0 = [0, 0, 0, 0, 0]^\top$, to serve as a baseline.

First of all, the Pac-Man scenarios are significantly larger and more complex than the foraging scenarios used in Section 5.2. For example, the Rewarding-Pellets Scenario has over 8×10^9 states. Additionally, the Pac-Man agent is lacking information regarding significant elements of the game, necessary to act optimally in the Pac-Man scenarios. In all scenarios, the agent cannot tell the exact position of the ghosts unless it is collocated with one of them, which generally leads to the end of an episode.

Considering some scenarios in more detail, in the Power-Pellet Scenario, the observations of the agent were not sufficient to distinguish the different behavior of the ghosts before and after the Power-Pellet was consumed. As such, the fitness-based agent ended up suffering a significant number of captures, attaining significantly negative fitness, as depicted in Figure 5.8(a). Our Pac-Man agent instead learned to avoid the ghosts in a first instance, and then to wait for the ghosts just below the Power-Pellet, which allowed it to capture them both after consuming the Power-Pellet. We thus note the emergence of a behavior, "waiting below the Power-Pellet", that according to what the agent is able to observe in that situation, nothing has to do with fitness.

In the Eat-all-Pellets Scenario, the fitness-based agent learned a very conservative strategy, avoiding being eaten by the ghost and aiming only at the fitness increment provided by eating the Power-Pellet. Because the small pellets did not provide any direct fitness increment, it never got to eat all pellets and get the largest fitness increment. Our agent, guided by a balanced consideration of all available sources of information, was able to partially disregard the largest reward provided by the Power-Pellet and instead learned to avoid the Smart Ghost throughout the environment,

[4] Illustrative videos of the observed behaviors in different stages of the learning process are available online at http://gaips.inesc-id.pt/~psequeira/emot-emerg.



(a) Power-Pellet Scenario

(b) Eat-all-Pellets Scenario

(c) Rewarding-Pellets Scenario

(d) Pac-Man Scenario

Figure 5.8: Evolution of the agents' fitness in each of the four Pac-Man scenarios. Results are averages over 200 independent Monte-Carlo trials. "Optimal" corresponds to an agent learning using r^*, "Fitness-based" to an agent receiving only fitness-based reward using $r^{\mathcal{F}}$, and "Random" is an agent acting randomly, *i.e.*, receiving no reward.

eating all small pellets in the process and, when possible, eating the Power-Pellet.

In the Rewarding-Pellets Scenario however, the small pellets did provide an immediate, albeit small reward. Although the fitness-based agent also developed a "preference" towards the reward provided by the Power-Pellet, it developed an approach strategy to the small pellets, and as such the difference for the optimal agent was attenuated, as illustrated in Figure 5.8(c). This difference in the structure of the scenario brings up the advantage of considering sources of information other than those related to fitness when the agent's observations are not informative enough.

The Pac-Man Scenario, being a variation of the other scenarios, denotes some of the problems that the fitness-based agent has in the other Pac-Man scenarios. Particularly, the fitness-based agent performs poorly when it encounters inconsistent elements in the environment—provided by the different behavior of the ghosts before and after consuming the Power-Pellet— and by not being able to observe the number of pellets eaten so far—necessary for optimal performance. On the contrary, an agent guided by the emerged sources of information considers other aspects of its history of interaction with the environment, allowing it to overcome challenges that cannot be directly observed.

5.4 Overall Discussion

Thus far we have emerged informative signals relating the agent's relationship with its environment and tested the applicability of using such sources of information in several scenarios. In this section we provide evidence on the similarities between the characteristics and effects of the emerged sources of information and those of emotions in nature. We analyze the *nature* of the signals from the perspective of emotion theory— particularly that of *appraisal*— and also discuss the implications of our approach and results to IMRL and the field of AC.

5.4.1 The Perspective of Appraisal Theories of Emotion

Recall from Section 3.3 that appraisal theories propose that emotions are elicited by evaluations (appraisals) of the significance of a situation for an individual's well-being or goals (Ellsworth and Scherer, 2003). Also, the most common theories of appraisal proposed in the literature model the elicitation of emotions as the result of a set of *appraisal variables*, each evaluating a particular aspect of the individual-environment relationship (Ellsworth and Scherer, 2003; Roseman and Smith, 2001).

For the purposes of our work, we now analyze the structural and dynamical properties of each of the sources of information emerged from the GP experiments that were detailed in Section 5.2.5. For each emerged source of information we relate to the evaluative properties of commonly proposed appraisal variables in the psychology literature. As such, our objective is to examine whether the emerged optimal sources of information have any kind of an *emotional tone* associated with them.

Specifically, we base our analysis in the work of Ellsworth and Scherer (2003) discussed in Section 3.3.2 in which five major dimensions or groups of appraisal are identified. Table 3.1 depicts the major dimensions and the kinds of appraisal variables usually related with each of them. For each emerged source of information we analyze characteristics of the signals and compare them with the criteria defined by several appraisal variables for the emotional evaluation of events to see whether some connection can be made.

Fitness: The expression for this source of information, $\phi_{\text{fit}} = r_{za}$, signals behaviors that directly enhance or reduce fitness. As we have seen, computationally this feature corresponds to the agent designer's reward function that directly ascribes *preferences* over the behavior of the agent. As such, fitness does not correspond to a subjective evaluation of the situation by the agent, a condition necessary for the process of emotional appraisal. This feature rather corresponds to external, innate preferences over certain characteristics of the environment that, depending on the scenario, the agent may choose to ignore. The most flagrant example of this occurred in the Moving Preys Scenario, where the best strategy focused only on the frequency of occurrence of stimuli, independently of their impact on fitness (see Table 5.2);

Relevance: This source of information denotes the impact of executing actions in some states for the agent's fitness in the long-run, as given by the expression $\phi_{\mathfrak{rel}} = q_{za}$ indicating the expected return of executing actions in states according to the fitness-based action-value function. This signal has properties related to appraisal dimensions that assert the *motivational significance* or *conduciveness* of a situation in relation to the individual's long-term goals or the satisfaction of its needs (Ellsworth and Scherer, 2003; Lazarus, 2001; Leventhal and Scherer, 1987) (see Table 3.1). By ascribing preference over actions that seem to lead to higher degrees of fitness in the long-run in a particular situation, this source of information directly denotes the contribution and future consequences of the behavior for the agent's goal—which, as we have seen, is to maximize its fitness in some environments of interest.

In nature, an evaluation of the *goal relevance* of a situation is essential for the adaptation of an individual to its environment, favoring behaviors that seem to promote its goals and desires (Ellsworth and Scherer, 2003). In our experiments, the emerged feature of relevance fostered behaviors that the agent believed conducive for its goals, especially in scenarios where the environment changed constantly, as is the case of the Prey Seasons Scenario (see Table 5.2) or the Power-Pellet Scenario (see Table 5.3);

Advantage: This source of information, expressed by $\phi_{\mathfrak{adv}} = q_{za} - v_z$, denotes the disadvantage of executing actions in some states considering their future impact on fitness. It thus gives origin to learned, acquired *preferences* by the agent towards behaviors it currently believes will lead to future high degrees of fitness in the environment. This feature is in accordance with the perspective that the *implicit value* of things changes as a consequence of experience and associative learning processes (Cardinal et al., 2002; Ellsworth and Scherer, 2003; Leventhal and Scherer, 1987). Unlike the *static* external preferences attributed by the fitness signal, this signal indicates the emotional *valence* of stimuli acquired through experience (Cardinal et al., 2002).

In nature, valence is considered an evaluation of the *value* of stimuli according to what the individual currently believes is the impact of the situation for its fitness (Leventhal and Scherer, 1987). We believe that *advantage* captures the essence of valence by rewarding (and thus promoting approach to) fitness-inducing states, and punishing (which leads to aversion towards) fitness-hindering situations, and also by being a continuous process biased by learning. While relevance corresponds to an estimate of future fitness gain, the advantage feature denotes the relative loss of executing some actions in certain states, which is sometimes important in situations where the *value* of stimuli changes throughout time, as is the case of the ghosts in the Power-Pellet Scenario, which impact on fitness changed depending of whether the Power-Pellet had been consumed by the agent (see Table 5.3);

Prediction: As stated earlier, this source of information, expressed through $\phi_{\mathfrak{prd}} = p_{zaz'}$, indicates how *predictable* the transition to some state is after the execution of an action in a previous state. According to many appraisal theorists of emotions, one important group of appraisal variables, referred to as *coping potential* dimension in Table 3.1, assess the ability that an individual believes to possess in order to cope with some emotion-eliciting situation (Ellsworth and Scherer, 2003; Frijda and Mesquita, 1998; Lazarus, 2001; Leventhal and Scherer, 1987). Situations which outcome is more predictable are more easy to cope with than those with more uncertain results. According to appraisal theories, an assessment of *control* tells the subject whether some situation can be altered to favor its current objectives, which often involves determining the degree of *predictability* or *probability* of the events being considered (Ellsworth and Scherer, 2003; Leventhal and Scherer, 1987; Scherer, 2001). In our framework, the emerged prediction feature indicates the level of *correctness* of the agent's world model, disfavoring the execution of actions that do not provide guarantees in terms of the future state of environment and favoring behaviors leading to more certain, expected situations;

Frequency: As the expression $\phi_{\mathfrak{frq}} = -{n_z}^2$ indicates, this source of information punishes visits to states to which the agent is more accustomed to, somehow favoring states that have been visited less often. One of the most basic and low-level dimensions proposed by many appraisal theorists has precisely to do with the *novelty* or *matching* between the perceived stimuli and the agent's acquired knowledge, usually referred to as the dimension of *familiarity* (Frijda and Mesquita, 1998; Leventhal and Scherer, 1987), as informed in Table 3.1.

In nature, the objective of this dimension is to focus the organisms's attention to changes perceived within the environment that might be relevant to its survival (Ellsworth and Scherer, 2003; Frijda and Mesquita, 1998). Likewise, the information provided by the frequency signal mainly punishes visits to states regularly encountered. As such, it is a feature that fosters exploratory behaviors, necessary to deal with always changing, unpredictable environments, such as the Moving Preys Scenario in the GP experiments (see Table 5.2), or situations where the agent has to continuously traverse its environment to achieve optimal performance, as was the case of the Rewarding-Pellets Scenario, in which it was advantageous for the agent to eat *all* the pellets in the environment as opposed to eating only one small pellet or even the Power-Pellet (see Table 5.3).

5.4.2 Coverage of the Emerged Signals

In the previous section we saw that the optimal sources of information indeed share structural and dynamical characteristics with some appraisal variables described in the specialized literature. This section complements this analysis and examines whether the appraisal as whole can be described with our sources of information, *i.e.*, whether the emerged signals cover *all* the dimensions defined by common appraisal theories.

As can be seen from Table 3.1, not all appraisal dimensions (or variables) are covered by the sources of information emerged in the first experiment in Section5.2. Although our main objective was to inspect the emotional properties of the emerged signals, we believe there are two main reasons behind the absence of some of the appraisal variables:

- The main reason has, perhaps, to do with the particular characteristics of the environments used to search for the optimal sources of information in the GP experiments, and not the variables provided to the evolutionary genetic algorithm.

  On one hand, the dynamics of the environments promoted the appearance of strategies that favored some (combinations of) specific sources of information in order to tackle with the challenges offered by each environment. On the other hand, different environments would possibly favor different (combinations of) sources of information. Nevertheless, the results of both experiments demonstrate the general-purpose and domain-independent character of the emerged signals, an attribute commonly associated with emotions in nature, where such mechanism seems to be necessary for the adaptation of organisms to ever-changing and sometimes unpredictable habitats;

- Another reason for the non-emergence of some of the appraisals proposed in the literature has to do with the characteristics of the agents themselves, which make the sources of information they have access to to be of a very “statistical nature”, which in terms of appraisal corresponds to being made at a rather low-level, *i.e.*, requiring little cognitive processing (Ellsworth and Scherer, 2003; Leventhal and Scherer, 1987). For example, evaluations such as the *causality* in Table 3.1 determine the responsibility or agency for the occurrence of some event (Ellsworth and Scherer, 2003; Scherer, 2001), which is difficult to assess in our experiments, where each new state is the consequence of: (a) the agent’s own actions; (b) the environment’s dynamics which in turn are dependent, for example, upon time.

  Also, we assume that the agent is always capable of performing a given action in some state according to its decision-making. Because of that, an evaluation of the agent’s *power*, within the appraisal of the coping potential, does not make much sense in our framework, whereas the notion of control, depending as we have seen on an evaluation of the predictability of action execution, can be easily assessed by the kinds of learning agents we model.

  Furthermore, the absence of the *social significance* variables is easy to explain, as we test the agents in single-agent scenarios. Appraisals like evaluating the compatibility of some behavior against socially-defined norms or moral, individual values would require the interaction of several agents within the same environment and a notion of expected social norms or values to be accessible to the agent, for example through its reward function.[5]

[5] We refer to Chapter 6 for the multiagent case where we propose intrinsic reward functions regarding information about the social context of the agents.

Table 5.4: Parallel between the influence of evolution and learning for the development of emotions in nature and our approach for the emergence of emotion-like reward signals.

	Nature	**Our Approach**
Fitness	adaptive capability / reproductive success	fitness function measures agent's behavior according to designer's expectations
Evolution	favors the development of adaptive behaviors through natural selection	GP procedure optimizes reward functions providing maximal fitness
Specific heredity	phylogenetically innate behaviors providing optimal response mechanisms in a specific environment	evolved reward functions allow for optimal performance in some set of environments
Learning	development of innate mechanisms to adapt to changing environments	RL algorithm allows the agent to learn the task / adapt to its environment
Emotions	general purpose processing mechanism evaluating the person-environment relationship	domain-independent rewards evaluating the agent's history of interaction with the environment
	directs behavior towards solving problems	guides the agent into solving the intended task
	shaped by evolution throughout time	emerged from evolutionary procedure

5.4.3 Parallel with Natural Evolution

In the context of our study, an appealing feature of GP over other search methods (such as gradient ascent (Sorg et al., 2010b)), is its close parallel with natural evolution. As discussed in Section 1.3, in nature, the individual's adaptive capabilities are shaped both by evolution, providing organisms with species-specific, *phylogenetic* processing mechanisms that enable advantageous behaviors in specific environments, and also individual learning, responsible for the *ontogenetic* development of organisms adapting to their environment throughout their lives (Anderson, 2000; Ginsburg and Opper, 1988). Table 5.4 summarizes this parallel between evolution, learning and emotions in nature and the experiments and validation performed in this chapter.

Likewise, the evolutionary-based procedure used in our experiment allowed us to discover those sources of information, encoded in the form of reward functions, providing maximal performance in some environment as measured by a fitness function. In some way, the reward functions used represent the agent's chromosomes, *i.e.*, innate phylogenetic (and internal) mechanisms for the agent to evaluate its environment during its lifespan. They make the agent to focus on those aspects of the environment it is predisposed to address. Similarly, the RL algorithm used by the agent is an ontogenetic mechanism as it allows it to adapt to its environment according to its innate characteristics. As denoted in Table 5.4, within this analogy, the fitness-function has the role of nature itself, evaluating the agents' performance according to a (external) fitness measure.

This analogy alone does not make the optimal sources of information have a more "emotional tone" or be more "emotion-like". However, the way they emerged parallels the way emotions have developed over time. The results from our experiments show that the reward signals emerging from the GP optimization procedure exhibit dynamics and properties that can be related to the way natural agents evaluate their environment, according to appraisal theories of emotions. The emerged features result from reward functions that provided optimal performance as measured by a fitness measure that *nothing has to do with emotions*. Moreover, these features, much like emotions in nature, also proved to be useful in different environments, providing a general-purpose reward mechanism for artificial learning agents.

5.4.4 Designed vs. Emerged Emotion-like Features

One of the contributions of this chapter is the establishment of a connection between the optimal sources of information complementing RL agents' perceptions, which emerged from the evolutionary procedure, and some appraisal variables of emotions evaluating the individuals' relationship with their environment in nature. Although that is not the main purpose of this study, we should compare the emerged reward features and the emotion-based features proposed in the previous chapter in Section 4.2.3 since their design was inspired by the same appraisal theories used in the analyzis in this chapter.

The first thing to note is that unlike the hand-designed emotion-based reward features of the previous chapter, in this chapter we used reward functions based on features that were emerged from an evolutionary procedure. The features proposed in Section 4.2.3 were guided by an interpretation over some appraisal dimensions, *i.e.*, we tried to mimic the types of evaluations proposed by appraisal theorists when designing the expressions for each feature. On the contrary, the GP algorithm in Section 5.2 was guided by variables relating the agent's history of interaction with the environment that have nothing to do with emotions. Moreover, the experiments had different objectives: in Section 4.3 we tested the usefulness and efficacy of using emotion-inspired rewards, regardless of the biological validity of the features proposed; in this chapter, the experiments aimed at verifying the emergence of features having an emotional tone and providing maximal fitness to the agents.

It is therefore expected that the majority of the expressions for the reward features developed in both chapters to be somehow distinct. In fact, we can only find similarities between the *Novelty* expression given by (4.1) in Section 4.2.3 and the *Frequency* feature defined in Section 5.2.5: they both promote exploration by rewarding states and actions less experienced. Also, the expression for *Valence* in (4.5) is similar in effect as that of *Advantage* defined in Section 5.2.5: they both evaluate how good it is to execute some action in a given state relatively to the best action in that state. In general, the emerged features are simpler in form than those proposed in the previous chapter.

Nevertheless, in both cases, a right combination of the emotion-based/emotion-like reward features allowed the agents to learn the intended task in the same set of foraging scenarios. For a comparison we refer to the results of the emerged features in Table 5.2 and the results of the emotion-based features in Tables 4.1–4.6.

5.5 Contributions

In this section we analyze in detail the contributions stemming from our second set of experiments in the context of this thesis. Generally speaking, these contributions help indicating the emergence of general-purpose informative signals in the context of reward function optimization. They complement the contributions of the previous chapter pointing towards the importance of emotion-based reward design in the performance of artificial learning agents, especially in the presence of perceptual limitations.

5.5.1 Implications for RL/IMRL

In this chapter we contribute for research within IMRL by emerging four *domain-independent, general purpose reward features* that can be applied in different scenarios with distinct purposes. Therefore, all the advantages of using these types of features to design reward functions, discussed in detail in the previous chapter in Section 4.5, are also true for the emerged features. Specifically, combining the emerged features enables the construction of:

- more *robust* agents, by being able to mitigate perceptual limitations inherent to artificial systems;
- more *flexible* agents, by successfully operating in a wide range of domains despite using the same reward base mechanism in each environment;
- more *autonomous* agents, by reducing the need for agent designers within RL to hand-code reward functions for a specific scenario.

Unlike the hand-designed reward features used in the previous chapter, these features were *emerged from an evolutionary procedure using GP.*

- In contrast with the approach in (Niekum et al., 2010) which used the same method within IMRL to solve the ORP but using domain-related information, in our experiments the GP algorithm departed from a set of primitive variables *summarizing the agent's history of interaction with its environment.*
- By using such domain-independent variables, the evolutionary algorithm is able to discover rewards that are *useful in a wider range of domains.* Although evolutionary computation in general can be used to optimize solutions in a particular domain (just like evolution shapes

individuals for optimal performance in *some* environment), our experimental results show that our solution is more flexible—when linearly combined, the emerged reward features can be used to learn in environments different from those in which they evolved;

- Moreover, the resulting reward functions fit the notion of intrinsic rewards according to the IMRL framework by: (a) providing motivation for behaviors not directly related to fitness; (b) evaluating aspects of the agent's history of interaction with its environment.

5.5.2 Implications for AC

We also contribute to the area of AC. Generally speaking, we go beyond the idea that emotions are only a *useful* mechanism for artificial agents by showing that emotions might indeed be *natural candidate* when considering several possible complements for the agent's perceptions.

- We used a *novel method for assessing the significance of embedding emotions into artificial agents* that, unlike previous approaches within AC, uses an evolutionary computation mechanism for the emergence of "emotion-toned" informative signals;

- The signals emerging from the GP procedure can be seen as *optimal sources of information* as they complement the agent's perceptual capabilities and guide it towards optimal performance in some environments of interest;

- The emerged sources of information have *dynamics and evaluative characteristics matching appraisal variables* proposed by common appraisal theories of emotions within the psychology literature;

- Some of the appraisal variables and dimensions are not covered by the emerged features, but this is mainly due to particular characteristics of the agents and the environments used in our experiments and not the primitive variables used by the genetic procedure;

- As discussed above, the evolved sources of information can be encoded as reward functions providing a *general-purpose attention-focusing and guiding mechanism*, as emotions are in nature;

- The evolutionary procedure used to determine the emotion-like reward mechanism resembles the way emotions are thought to have evolved in nature, *i.e.*, guided by a measure of their benefit to the individual's survival. This enables an interesting computational parallel to the existing evidence in biological systems—where indeed, the organisms with most complex emotional processing (humans) are arguably those better fitted to their environment;

- On the other hand, from the thousands of reward functions generated and tested by the evolutionary procedure, the ones providing maximal fitness for the agent have characteristics resembling the way humans use emotions to *appraise* their environment. This fact may also reinforce research in neurology and psychology attesting the necessity of a processing mechanism like emotions for the proper adaptation of individuals to their environment—the lack of which, as our results show, can lead to maladaptive behaviors;

- All the above findings resulting from our study thus point towards the idea that emotions might have a greater impact for the adaptation of artificial agents to their environments than thought before.

5.6 Summary

In this chapter we addressed the question relating the impact of emotion-based signals for the performance of autonomous agents. We followed a novel approach that is inspired by the way emotions have developed in nature throughout evolution as a mechanism that allows individuals of better adapting to their environment. We used an evolutionary computation algorithm guided by a measure of the agent's fitness to its environment within IMRL. As a result, we emerged a set of basic reward signals complementing the agent's perceptions and providing optimal performance in a set of foraging scenarios.

We then verified the generality and applicability of these sources of information by testing their use in a set of scenarios different from those in which they were evolved. In order to assess if these emerged signals have an emotional tone, we analyzed them from the perspective of appraisal theories of emotions. Indeed, we found that the kinds of evaluations they make about the agent's relationship with its environment share some structural and dynamical properties with common dimensions of emotional appraisal that, according to appraisal theorists, are used by individuals in nature to evaluate the impact of some situation for their goals and needs and respond accordingly.

In the following chapter we start to expand the work developed in the two previous chapters to multiagent domains. In that respect we use IMRL within multiagent learning to discover whether socially-aware, altruistic behaviors can emerge in a population of self-interested agents competing with each other for "food resources".

CHAPTER 6

Socially-Aware Learning Agents

This chapter represents a first step towards extending the work developed in Chapter 4 and Chapter 5 into multiagent systems (MAS). For that purpose we expand the IMRL framework into the multiagent paradigm and propose the design of reward functions inspired by the way humans and other animals signal socially-aware behaviors within a social group. Our approach is delineated in Figure 6.1.[1]

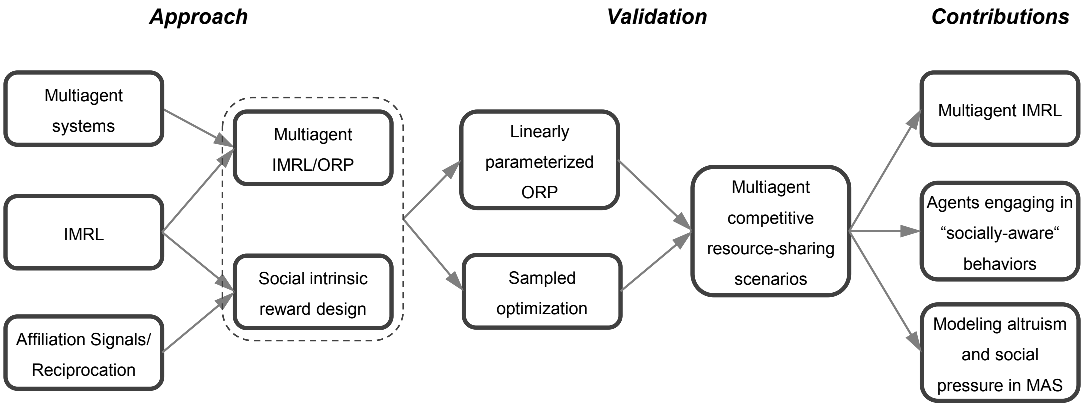

Approach
Validation
Contributions
Multiagent systems
IMRL
Affiliation Signals/ Reciprocation
Multiagent IMRL/ORP
Social intrinsic reward design
Linearly parameterized ORP
Sampled optimization
Multiagent competitive resource-sharing scenarios
Multiagent IMRL
Agents engaging in "socially-aware" behaviors
Modeling altruism and social pressure in MAS

Figure 6.1: Outline of our approach for socially-based intrinsic reward design.

6.1 Introduction

In this chapter we explore the impact of intrinsic motivational systems in multiagent scenarios. Multiagent systems (MAS) encompass groups of intelligent agents interacting within the same environment. Having sometimes little or no knowledge about the other agents' decision-making processes they are expected to share resources in their environment, coordinate to perform specific tasks or maximize some measure of joint utility. On top of this, each agent needs also to fulfill its own individual goals and needs, often *divergent* from the other agents. Over the years, MAS research has focused on creating languages, organizational paradigms and mechanisms that help structure the interactions in groups of agents (Wooldridge, 2002). In this context, RL has been widely applied within MAS as a means to gradually improve the performance of the agents in the environment (Busoniu et al., 2008; Claus and Boutilier, 1998).

As described throughout Chapter 2, RL is best described as a set of techniques aimed at helping agents facing a sequential decision problem to learn an intended task by gathering as much positive environmental feedback—or reward—as possible. Many such techniques come with proofs of convergence to a strategy that maximizes the agent's reward in the long-run, as long as some conditions are met (Kaelbling et al., 1996; Sutton and Barto, 1998). However, these guarantees seldom hold in multiagent settings, where an agent's reward depends on the other agents' actions (Busoniu et al., 2008; Claus and Boutilier, 1998). Additionally, when considering MAS, one may

[1] Part of the contributions within this chapter are included in (Sequeira et al., 2011b).

no longer be interested in maximizing a single agent's return overtime, but rather that the agents' strategies converge to some global maximum across agents.

As noted in (Schuster and Perelberg, 2004), cooperation in nature is an intrinsically social phenomenon that goes beyond physiological events like eating or mating. Cooperative behaviors like altruism have emerged throughout evolution between organisms who are inherently competitive, mainly due to the conditions of their habitat (Axelrod, 1984; Dawkins, 2006). Although this seems to contradict classic theory of evolution (focusing on the "survival of the fittest") (Darwin, 2009), the advantages of engaging in such "socially-aware" behaviors can be explained by other factors: individuals can take interest on the well-being of others by means of *close relatedness*. Either through *kin selection* or familiarity (*e.g.*, belonging to the same social group), closely-related individuals may engage in altruistic behaviors despite possible losses for the contributor (Axelrod, 1984; Dawkins, 2006; Hamilton, 1964; Schuster and Perelberg, 2004); even weakly or non-related organisms, by means of *reciprocation*, are able to evaluate the "kindness" of others and respond accordingly (Axelrod, 1984; Falk and Fischbacher, 2006; Trivers, 1971). Cooperation between individuals (even of different species) can then be promoted as a means to achieve benefits for all the interveners in the long-run (Axelrod, 1984; Trivers, 1971).

In this chapter we focus on the emergence of cooperation in groups of competing, self-interested, independent RL agents interacting with one another in a common environment. We investigate mechanisms that aid the individual learners in reaching cooperative behaviors without observing or explicitly reasoning about the strategies or rewards of other agents. For this purpose we explore *intrinsic motivation* mechanisms in multiagent settings. As we have seen in Section 2.4.3, most of the works within IMRL focus on single-agent scenarios, where the motivational system drives a single agent to engage in behaviors that are not directly "survival"-related. In this study, however, we examine how these same mechanisms may drive an agent to engage in behaviors that are "socially aware", in a sense to be made clear.

We explore *social motivational mechanisms* for groups of agents coexisting in limited-resource environments. In that respect we adopt ideas about how biological agents engage in cooperative behaviors despite their inherently competitive environment (*e.g.*, Axelrod and Hamilton, 1981; de Waal, 2008; Dörner, 1999). We formally extend to multiagent settings the framework of IMRL and study the impact of *social signaling* and *reciprocity* in the design of social intrinsic rewards. Our approach is fundamentally different from recent work on the *multiagent optimal reward problem* (Liu et al., 2012), as we do not assume shared perception across agents. We test our approach in several multiagent scenarios involving complex interdependences between the behaviors of up to 3 interacting agents. Our results show that the proposed rewarding mechanism indeed enables the emergence of "socially-aware" behaviors that drive the agents to learn strategies that exchange immediate individual gains for long-run group benefits. We analyze environmental conditions in which some group phenomena (such as alliances) may emerge, and examine possible implications of evolutionary game theory and certain social phenomena on our approach and results.

6.2 Background

6.2.1 Social Mechanisms for Cooperation

In order to develop reward signals that take into consideration social interactions among agents, we consider the notion of *affiliation* from Dörner (1999) Psi-theory Dörner (1999). Generally-speaking, Psi-agents have an urge to affiliate with other agents by sending and receiving *legitimacy signals*, also referred to as *l-signals*, that express the sender's belief in the social acceptability of the receiver's actions (Bach, 2009). According to this theory, individuals have an urge to affiliate with each other so as to enhance their legitimacy within the social group. Dörner (1999) defines three types of legitimacy signals that are of interest to our study:

- l-signals provide positive social reinforcement, thus rewarding successful interactions;
- anti-l-signals punish unsuccessful social interactions, thus diminishing the receiver's legitimacy within the group (which is equivalent to the expression of a "frown");
- internal l-signals, as the name indicates, are generated internally by the individual to reward socially-acceptable behaviors, without the need of having other sending the signals (they represent a notion of "honor").

Psi-theory does not identify any specific social behavior susceptible to be rewarded with legitimacy signals. It rather proposes signals that can be measured by evaluating the "okayness" of some behavior being considered within a social context (Bach, 2009). Moreover, according to this theory, as a result of exchanging these signals, individuals end up engaging in cooperative behaviors as a means to gain legitimacy within the social group.

6.2.2 Partially Observable Markov Games

In this study, we adopt a game-theoretic model to describe our multiagent system, the *partially observable Markov game* (POMG) with K players, where each player corresponds to an agent in our system. A POMG is an extension to the POMDP model described in Section 2.2.2 and can be denoted as a tuple $\Gamma = (K, \mathcal{S}, (\mathcal{A}^k), (\mathcal{Z}^k), \mathsf{P}, (\mathsf{O}^k), (r^k), \gamma)$, where

- K is the number of agents;
- $\mathcal{S}$ is the set of all possible environment states;
- $\mathcal{A}^k$ denotes the action repertoire of agent k;
- $\mathcal{Z}^k$ is the set of all possible observations of agent k;

- We write $\mathsf{O}^k(z^k \mid s)$ to denote the probability of agent k observing z^k at time-step t. Formally, this corresponds to
$$\mathsf{O}^k(z^k \mid s) \triangleq \mathbb{P}\left[z_t^k = z^k \mid s_t = s\right];$$

- $r^k(s, a)$ represents the *average reward* that agent k expects to receive for performing action a in state s;

- $0 \leq \gamma < 1$ is some *discount factor*.

The model proceeds as follows. Just like in a POMDP, at each time-step t, the game is in one of a finite set $\mathcal{S}$ of possible states, denoted by $s_t = s$. Each agent k has access to a partial view of s_t, henceforth referred as the *observation* of agent k at time-step t and denoted as $z_t^k \in \mathcal{Z}^k$. At each time-step t, every agent k in the game selects, *simultaneously and without explicit communication*, an action from a set $\mathcal{A}^k$ of possible actions. We write a_t^k to denote the individual action of agent k at time-step t, and write $a_t = \langle a_t^1, \ldots, a_t^k \rangle$ to denote the *joint action* of all agents at time-step t. We define the set of *joint actions* as $\mathcal{A} = \times_{k=1}^K \mathcal{A}^k$. The game then transitions, at time-step $t + 1$, to a state $s_{t+1} \in \mathcal{S}$ that depends only on s_t and a_t. Denoting by $\mathbf{h}_t$ the *history of the game* up to time-step t, we have

$$\mathbb{P}\left[s_{t+1} = s' \mid \mathbf{h}_t\right] = \mathbb{P}\left[s_{t+1} = s' \mid s_t, a_t\right] \triangleq \mathsf{P}(s' \mid s, a),$$

for $s \in \mathcal{S}$ and $a \in \mathcal{A}$. Each agent k is awarded a reward r_{t+1}^k that depends only on s_t and a_t and is such that
$$\mathbb{E}\left[r_{t+1}^k \mid s_t = s, a_t = a\right] = r^k(s, a),$$

for some bounded real-valued (reward) function $r^k : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. Each agent k receives a new observation z_{t+1}^k and the process repeats. Each agent is considered an *individual learner* (Claus and Boutilier, 1998), meaning that just like in the POMDP model, each agent will pursue its individual goal by selecting its actions so as to maximize the quantity in (2.1), *i.e.*,

$$V^k(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r^k t + 1 \mid s_0 = s\right], \tag{6.1}$$

for the POMG model.

Solving a POMG then amounts to computing, for each agent $k = 1, \ldots, K$, a mapping $\pi^k : \mathcal{S}^k \to \mathcal{A}^k$ that corresponds each possible state in $\mathcal{S}^k$ an action in $\mathcal{A}^k$ in such a way that (6.1) is locally maximized for all agents (a situation known as *equilibrium*). As we have seen, such mapping is known as a *policy*, and is intractable to compute except for the simplest games (Goldsmith and Mundhenk, 2007). However, several specialized models, like the single-agent MDP model (Puterman, 2009), are manageable to efficient solutions. An MDP is a POMG where $K = 1$,

$\mathcal{Z} = \mathcal{S}$, and O is the identity mapping[2]. In the same manner, a single-agent POMG is known as a POMDP.

As we have seen throughout Section 2.2.2, RL develops algorithms for individual learners that are able to compute the optimal policy π^* by gathering information from successive interactions with the environment and building estimates $\hat{\mathsf{P}}$ and $\hat{r}$ of P and r. However, in spite of their expressive power, the computational complexity involved in solving POMGs renders this model impractical for our purposes. In contrast, MDPs can be solved efficiently (Littman et al., 1995), although they are unsuitable to describe most multiagent problems of interest, since they do not accommodate for the possibility of an agent explicitly reasoning about how other agents influence its performance.

In this study we are mainly interested in studying the emergence of cooperative behavior in groups of self-interested agents co-existing in a common environment. Such agents can be seen as computational counterparts of simple fitness-maximizing individuals, and it may debatable whether such agents would reason extensively about the behavior of other agents in the environment (Axelrod, 1984). In such conditions, an ability to *reciprocate* to and *distinguish* between interacting individuals is fundamental for coordination to emerge from the complex patterns within the interaction history.

In that manner, in this study we follow our previous approach of considering agents that *reason about the world as if it were an MDP*, ignoring both the existence of other agents (to the extent that such existence is not evident from its observations) as well as the possibility that its perception of the state s_t of the world may be imperfect. Therefore, like with the previous experiments, *each* agent k can use prioritized sweeping (Moore and Atkeson, 1993) to build an approximated MDP model of its environment, $\hat{\mathcal{M}}^k = (\hat{\mathcal{S}}^k, \mathcal{A}^k, \hat{\mathsf{P}}^k, \hat{r}^k, \gamma)$, where $\hat{\mathcal{S}}^k = \mathcal{Z}^k$ and $\hat{\mathsf{P}}^k$ and $\hat{r}^k$ are built by averaging over the perceptions of agent k.

6.3 Socially-Motivated Learning Agents

In this section we introduce our main contribution, extending the IMRL framework to multiagent scenarios. We discuss two (social) reward signals to be used within IMRL and how these relate to specific social interactions studied in the specialized literature.

6.3.1 Multiagent IMRL

In general, the above approach will lead to a very crude model and have a significant impact in terms of the performance of the agent (Littman, 1994). As we have seen however, recent work on IMRL shows that, in single agent scenarios, it is possible to use richer reward functions that implicitly encode additional information about the task of the agents to overcome some of

[2] Since there is a single agent, we can drop the superscript ^{k}.

the agents' perceptual limitations. To extend the IMRL framework to multiagent settings, we introduce some additional notation. Formally,

- let $h_t^k = \{z_0^k, a_0^k, \rho_1^k, \dots, \rho_t^k, z_t^k\}$ to denote the *individual history* of agent k up to time-step t under partial observability settings. We denote the *joint history of all agents* as $\mathbf{h}_t = \langle h_t^1, \dots, h_t^K \rangle$ taken from a set $\mathcal{H}$ of possible (finite) joint interactions. Recall that $\{\rho_\tau^k, \tau = 1, \dots, t\}$ corresponds to the agent designer's "external" evaluation signal that, at each time step t, depends only on the underlying state s_{t-1} of the environment and the action a_{t-1}^k performed by agent k. In general, the distribution governing h_t^k depends both on the *environment* $e \in \mathcal{E}$ and on the *task*, where $\mathcal{E}$ is some *set of environments* we want our group of agents to perform well;

- let $\mathbf{h}_t^{-k} = \langle h_t^1, \dots, h_t^{k-1}, h_t^{k+1}, \dots, h_t^K \rangle$ denote a *reduced joint history*;

- just like in single-agent IMRL, $r^{\mathcal{F},k}$ denotes the *fitness-based reward function* of agent k that is formally defined as

$$r^{\mathcal{F},k}(s, a^k, h^k) = \mathbb{E}\left[\rho_{t+1}^k \mid s_t = s, a_t^k = a^k\right]. \tag{6.2}$$

- the task is encoded by the set of *individual reward functions* $r^k, k = 1, \dots, K$. We write $\mathbf{r} = [r^1, \dots, r^K]$ to denote the vector of all individual reward functions and henceforth refer to the vector-valued function $\mathbf{r} : \mathcal{S} \times \mathcal{A} \times \mathcal{H} \to \mathbb{R}^K$, with kth component corresponding by $r^k(s, a^k, h^k)$, as the *POMG reward function*;

- following the POMG model described in Section 6.2.2, each agent k is an *individual learner* that tries to maximize the reward provided by r^k overtime;

- given a particular joint history $\mathbf{h}_t$, we write $p_H(\mathbf{h}_t \mid \mathbf{r}, e)$ to denote the probability of the agents (as a group) observing the history $\mathbf{h}_t$ in environment e given the (POMG) reward function $\mathbf{r}$;

In extending IMRL to multiagent settings, we now measure the *combined performance* of a group of K agents in terms of their average fitness. In particular, given a finite joint history $\mathbf{h}_t$, we compute the *combined fitness* as

$$f(\mathbf{h}_t) = \frac{1}{K} \sum_{i=k}^{K} f^k(h^k) = \frac{1}{K} \sum_{i=k}^{K} \sum_{\tau} \rho_\tau^k, \tag{6.3}$$

where $f^k(h^k)$ corresponds to the individual fitness of agent k. From (6.3), we get that the *fitness* of the POMG reward function $\mathbf{r}$ can be given by

$$\mathcal{F}(\mathbf{r}) \triangleq \sum_i f(\mathbf{h}_t^i) p_H(\mathbf{h}_t^i \mid \mathbf{r}, e^i) p_E(e^i) = \frac{1}{K} \sum_{i,k} f^k(h^k) p_H(h_t^{i,k} \mid r^k, e^i) p_E(e^i), \tag{6.4}$$

where $\mathbf{h}^i$ denotes a particular joint history of interaction and e^i is sampled according to the distribution $p_E(\mathcal{E})$. Given a set $\mathcal{R}$ of POMG reward functions, a set $\mathcal{E}$ of possible environments and a distribution $p_E(\mathcal{E})$, the optimal reward problem (ORP) defined in Section 2.4.3 extended for multiagent settings can then be formalized as that of determining the optimal POMG reward function $\mathbf{r}^* \in \mathcal{R}$ such that

$$\mathbf{r}^* = \operatorname*{argmax}_{\mathbf{r}\in\mathcal{R}} \mathcal{F}(\mathbf{r}). \tag{6.5}$$

We note that, as discussed in the previous section, our agents do not explicitly reason about other agents, and whatever "collaborative" behavior emerges from their interaction does not result from any explicit social considerations that they may have on the well-fare of others. In scenarios where the fitness $\mathcal{F}$ depends critically on the ability of the individual agents to cooperate, IMRL provides a natural framework to study the emergence of cooperative behavior.

6.3.2 Multiagent Linearly Parameterized ORP

In this chapter we again follow the *linearly parameterized approach to ORP* described in Section 2.5.1 and extend it to multiagent settings. In this manner, each POMG reward function $\mathbf{r} \in \mathcal{R}$ is the linear combination of a family of vector-valued reward features $\{\boldsymbol{\phi}_\alpha, \alpha \in \Lambda\}$, where Λ is some index set and each $\boldsymbol{\phi}_\alpha$ is, in itself, one possible (POMG) reward function that captures relevant elements of the history interaction of the agents with the environment.[3] Therefore, every $\mathbf{r} \in \mathcal{R}$ can be written (componentwise) as

$$r^k(s, a^k, h^k) = \sum_\alpha \phi^k_\alpha(s, a^k, h^k)\theta^k_\alpha, \tag{6.6}$$

where ϕ^k_α is the value of the kth component of $\boldsymbol{\phi}_\alpha$ and, at each time-step t, is a function of s_t, a^k_t and h^k_t. Similarly, θ^k_α is the value of the parameter associated with $\boldsymbol{\phi}^k_\alpha$. Abusing the notation, we sometimes write

$$\mathbf{r}(s, a, \mathbf{h}) = \sum_{\alpha\in\Lambda} \boldsymbol{\phi}_\alpha(s, a, \mathbf{h})\boldsymbol{\theta}_\alpha,$$

with the understanding that the products are taken componentwise as indicated in (6.6), and refer to the resulting reward function as $\mathbf{r}(\boldsymbol{\theta})$. Like with the single-agent ORP, our goal is then to determine $\boldsymbol{\theta}^*$ such that

$$\boldsymbol{\theta}^* = \operatorname*{argmax}_{\boldsymbol{\theta}} \mathcal{F}(\mathbf{r}(\boldsymbol{\theta})).$$

6.3.3 Social IMRL Model

We now introduce our second contribution in this chapter—namely, a set of *reward features* that lead to the emergence of cooperative behavior in multiagent IMRL. Figure 6.2 depicts our model

[3] Given the set $\{\boldsymbol{\phi}_\alpha, \alpha \in \Lambda\}$, we henceforth write $\boldsymbol{\phi}_\alpha$ to denote the (vector-valued) reward feature with components $[\phi^k_\alpha]^K_{k=1}$, and $\boldsymbol{\phi}^k$ to denote the vector of (individual) reward features $[\phi^k_\alpha]^{\alpha\in\Lambda}$.

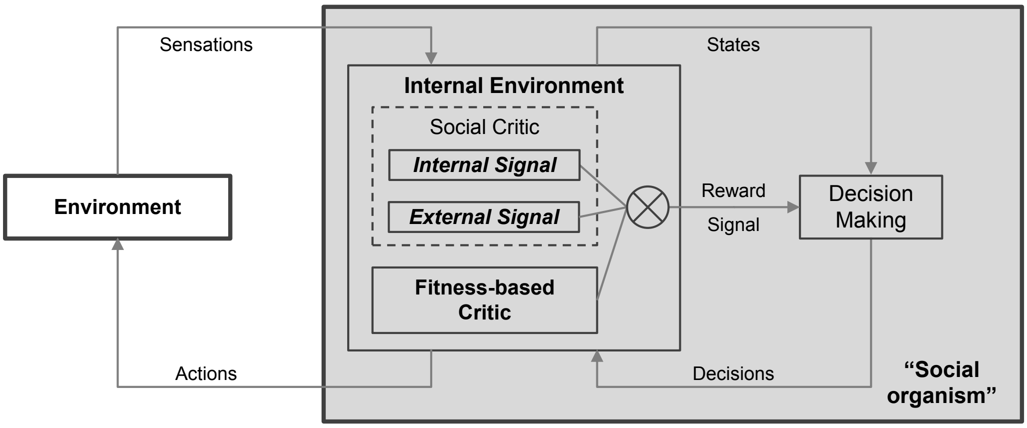


Figure 6.2: Proposed framework for intrinsic social rewards. Adapted from (Singh et al., 2009).

for an individual learner within social IMRL. It extends the general IMRL model in Figure 2.7 and parallels our model for emotion-based rewards in Figure 4.2. In this case, a social agent is modeled as having a *social critic* as a reward mechanism providing two intrinsic social reward features based on the notion of l-signals provided in Section 6.2. This critic can be seen as the mechanism evaluating the behavior of the agent from a social legitimacy point-of-view. We consider social reward features depending on social variables accessible to the agents of a social group in the context of resource sharing foraging scenarios.

Our approach is inspired by existing theories about how groups of individuals coordinate in nature. Specifically, we adopt the notions of *group relatedness* and *reciprocity* from *kinship* (Dawkins, 2006) and *reciprocity* (Falk and Fischbacher, 2006; Trivers, 1971) theories. The social structure is simplified and assumes that the interacting agents belong to the same *social group* and that their relatedness is assured by an urge to affiliate with each other (Dörner, 1999). We thus propose a set of reward features that can be interpreted as computational counterparts to the several social l-signals discussed in Section 6.2.1 (Bach, 2009; Dörner, 1999) depending on social variables accessible to agents of a certain social group in the context of resource sharing foraging scenarios, together with ideas from kinship and reciprocity theories (Dawkins, 2006; Falk and Fischbacher, 2006). Formally, we consider a set of three reward features, $\Phi = \{\phi^k_{\mathfrak{ext}}, \phi^k_{\mathfrak{int}}, \phi^k_{\mathfrak{fit}}\}$, where

- $\phi^k_{\mathfrak{ext}}(s, a^k, h^k)$ can be interpreted as a computational counterpart to the (external) l-signals. can be interpreted as a computational counterpart to the (external) l-signals. It indicates the change in the legitimacy of agent k within the social group when the latter executes action a^k in state s after observing history h^k, as implicitly determined by the other members of the group;

- $\phi^k_{\mathfrak{int}}(s, a^k, h^k)$ can be interpreted as a computational counterpart to the internal l-signals and indicates the adjustment of the agent's social legitimacy by trying action a in state s after observing history h^k, as internally dictated by the "social standards" of agent k;

- $\phi^k_{\mathfrak{fit}}(s, a^k, h^k) = \rho^k(s, a^k, h^k)$ is agent k's fitness-based reward for performing action a^k in state s after observing history h^k.

For each agent k, the reward features above can be interpreted as representing either how satisfied the affiliation need of agent k is, or the change in social legitimacy decurrent of having performed action a^k in state s after observing history h^k.[4]

For each $\alpha \in \{\mathfrak{ext}, \mathfrak{int}, \mathfrak{fit}\}$ and each agent $k = 1, \ldots, K$, we restrict θ^k_α to lie in the interval $[-1.0; 1.0]$. Each θ^k_α determines the relative contribution of the corresponding social reward-feature, ϕ^k_α, in the overall reward. For example, a parameter vector $\boldsymbol{\theta}^k = [0; 0; 1]$ indicates that agent k values only fitness-based reward and completely ignores the social rewards, *i.e.*, it *disregards its social context*. Different scenarios will "demand" that each agent pays attention to its social context in a specific manner, so that cooperation is achieved and the fitness of the group of agents as a whole is maximized. Like in single-agent IMRL, we consider the specific configuration of $\boldsymbol{\theta}$ to be fixed during learning, meaning that the agents are somehow predisposed to acknowledge social information in some "innate" manner.

We remark that our general formulation of multiagent linearly parameterized ORP enables us to model each agent as having its own "personality". In other words, by agent-specific features, ϕ^k_α and agent-specific weights, θ^k_α, it is possible to have agents with different innate personalities, that perceive different reward features and weight them differently in the overall reward, which in turn implies that each agent will have specific motivations towards its (social) environment. Moreover, by studying the absolute value of the weight $\boldsymbol{\theta}_\alpha$ associated with each feature ϕ_α, it is possible to assess whether our proposed social features indeed provide useful information for the coordination of the group as a whole—in which case we would expect $\|\boldsymbol{\theta}_\alpha\| \gg 0$. Finally, one third advantage is that we can compare across the different agents in the group the corresponding weight vectors, $\{\boldsymbol{\theta}^k, k = 1, \ldots, K\}$, and assess the (non-)homogeneity of the agents' personalities, allowing a richer analysis of the structure of the group fostering maximal global fitness in the environments of interest.

6.3.4 Resource-Sharing Environments

In order to evaluate the proposed framework for multiagent IMRL and the social reward features discussed in the previous section, we analyze their impact on the behavior of a group of agents inhabiting an inherently competitive environment, where a limited amount of resources is available. Each resource, if consumed, directly enhances an agent's (individual) fitness. In our study, we adopt simplifications of several foraging scenarios introduced in (Singh et al., 2009, 2010) and design several new scenarios in which coordination can only occur if the agents learn to perform

[4] We note that, although the above reward features are indicated as generally depending on the state s_t of the environment, in most cases of interest they will depend only on a^k_t and h^k_t. We included the explicit dependence of s_t for generality, since it allows us to trivially consider situations with full-state observability.

cooperative joint actions to obtain the shared resources. It is important to note that the introduced simplifications focus mainly on the dimension of the environments used, not in terms of the corresponding coordination problems. Therefore, they do not impact our purpose of investigating the usefulness of the proposed social intrinsic rewards for the emergence of cooperative behaviors.

To instantiate the social reward features discussed in Section 6.3.3 in the context of resource (food) sharing scenarios, we note that legitimacy signals provide "social feedback" on the agent's feeding behavior, and the agents are able to reciprocate through an implicit signaling mechanism that rewards "kind" behaviors and punishes "unkind" ones (Falk and Fischbacher, 2006).[5]

The social critic in Figure 6.2 can determine when the agent was the last to consume a resource from the environment. Each agent is also able to detect if another agent is collocated in the environment. For all scenarios considered in this chapter, we define the following local events, used to determine the different social reward features for each agent k:

- $\textsc{food}^k(t)$ denotes the event that, at time-step t, agent k is collocated with a food resource;
- $\textsc{full}^k(t)$ denotes the event that, at time-step t, agent k is in a fully satiated state;
- $\textsc{hungry}^k(t)$ denotes the event that, at time-step t, agent k is in a hungry state. We note that each agent cannot be hungry and full at the same time, *i.e.*, $\forall_t \textsc{full}^k(t) \rightarrow \neg\textsc{hungry}^k(t)$, but they can be in an intermediate satisfied state, when they are neither full or hungry;
- $\textsc{eat}^k(t)$ denotes the event that, at time-step t, agent k decided to consume a food resource;
- $\textsc{other\_eat}^k(t)$ denotes the event that, at time-step t, agent k is collocated with some other agent, and that the other agent decided to consume a food resource;
- $\textsc{other\_ate}^k(t)$ denotes, at time-step t, the number of agents (other than agent k) that ate a food resource since k last consumed a resource itself.

Fitness-based Reward Feature

In the proposed environments, consuming food resources is the only behavior that directly contributes to the fitness of each agent (and therefore that of the population), while being hungry decreases the fitness. Therefore,

$$\phi^k_{\texttt{fit}}(t) = \mathbb{I}\left[\textsc{full}^k(t)\right] - \beta_{hp} \cdot \mathbb{I}\left[\textsc{hungry}^k(t)\right] \tag{6.7}$$

for agent k, where β_{hp} is the (environment-specific) *hunger penalty* and $\mathbb{I}\left[\varepsilon\right]$ denotes the *indicator function* for event ε.

[5] We again emphasize that our agents do not have any prior common knowledge or shared information except that which is conveyed by their individual observations. Therefore, there is no way for the agents to explicitly communicate their individual behaviors. In particular, the social signals are implicitly encoded in the associated intrinsic reward features received by each agent as part of its reward and, as such, do not involve any means of explicit information sharing.

6.3.5 Intrinsic Social Reward Features

The above events can be perceived by each agent k from its observation z_k at time t. Using these definitions, we can now define the *social intrinsic reward features* $\phi^k_{\mathfrak{ext}}$ and $\phi^k_{\mathtt{int}}$ and discuss their relation with the internal and external l-SIGNALS. In the context of this thesis we only provide a possible interpretation of these quantities. Moreover, we do not claim this interpretation to be universal or even biologically plausible. However, as will become apparent, our interpretation facilitates the discussion of the results. Depending on the social environments being considered, especially if not related with resource sharing, other features may be defined.

In the context of resource sharing $\phi^k_{\mathfrak{ext}}$ and $\phi^k_{\mathtt{int}}$ should signal whether the feeding behavior of an agent was "considerate" to the other members of the social group. For example, a *socially aware* agent takes into consideration whether it was the last to eat before deciding to consume some resource. Therefore, we define the *strength of the social feedback* to agent k at time-step t as

$$\sigma^k(t) = \frac{K - \textsc{other\_ate}^k(t) - 1}{K - 1}.$$

Note that $\sigma^k(t) = 0$ if all other agents have eaten since agent k last ate, and $\sigma^k(t) = 1$ if agent k was the last to eat. The strength of the social feedback signals to agent k the severity/thoughtfulness of its behavior.

External Social Reward Feature

Formally, we define the *external social reward feature* for agent k as

$$\phi^k_{\mathfrak{ext}}(t) = \sigma^k(t) \cdot \mathbb{I}\left[\textsc{food}^k(t)\right] \cdot \mathbb{I}\left[\textsc{other\_eat}^k(t)\right] \cdot \left(\mathbb{I}\left[\neg\textsc{eat}^k(t)\right] - \mathbb{I}\left[\textsc{eat}^k(t)\right]\right).$$

The feature $\phi^k_{\mathfrak{ext}}$ is provides a "positive signal" to agent k whenever it allows other agents to eat, given that agent k was the last to eat. Conversely, it provides a "negative signal" when agent k eats a food resource *in the presence of another agent*, if agent k was the last to eat.

Recall from Section 6.2 that the PSI-theory defines l-signals and anti-l-signals as somehow encoding the degree of acceptance of the conduct of an individual by other members of its social group, *i.e.*, its *social legitimacy* (Bach, 2009). In the proposed setting, $\phi^k_{\mathfrak{ext}}$ allows for reciprocation by taking a positive value whenever agent k has the possibility to consume a food resource but chooses not to because it was the last to eat, thus giving the other agents (in its social group) the opportunity to be satiated. Since this corresponds to a thoughtful behavior, the positive value of $\phi^k_{\mathfrak{ext}}$ can be seen as an implicit "acceptance signal" by the other agents that increase agent k's legitimacy. A similar argument can be made with respect to the negative values provided by $\phi^k_{\mathfrak{ext}}$, which can be interpreted as "social punishment" for agent k's lack of consideration for the group's well-being.

Internal Social Reward Feature

We define the *internal social reward feature* for agent k as

$$\phi^k_{\texttt{int}}(t) = \sigma^k(t) \cdot \mathbb{I}\left[\textsc{food}^k(t)\right] \cdot \left(\mathbb{I}\left[\neg\textsc{eat}^k(t)\right] - \mathbb{I}\left[\textsc{eat}^k(t)\right]\right).$$

The feature $\phi^k_{\texttt{int}}$ provides a “positive signal” to agent k whenever it decides not to eat, given it was the last to eat. Conversely, it provides a “negative signal” when agent k decides to consume a resource, if agent k was the last to eat. The difference between the $\phi^k_{\texttt{crt}}$ and $\phi^k_{\texttt{int}}$ is that the latter does not require a conflicting eat action by another agent.

In Section 6.2 we saw that internal l-signals measure how much an agent’s actions are in accordance to its own *internal standards* (Bach, 2009). In this manner, the reward feature $\phi^k_{\texttt{int}}$ takes a positive value whenever agent k has the possibility to consume a food but decides not to because it was the last to eat—independently of the presence or absence of other agents. Because the environments considered in our study deal with resource sharing, this reward feature somehow encodes the *degree of satisfaction* that agent k gets for engaging in such *altruistic* behavior.

Socially-aware agents feel *intrinsically rewarded* when they feel they engage in a behavior for the benefit of other members of its social group (de Waal, 2008). Moreover, altruistic behaviors may carry an initial cost that is only compensated after a certain time period (Axelrod, 1984; de Waal, 2008). In the proposed setting, $\phi^k_{\texttt{int}}$ “rewards” agent k for altruistic behavior, even if this implies a smaller momentary individual fitness. Similarly, it “punishes” inconsiderate behaviors, even if the agent was not directly competing with another agent for a food resource.

6.4 Experiments and Results

In Figure 6.1 we outline the method to validate our approach for social-based intrinsic reward design. To that purpose we design a series of experiments in resource sharing scenarios similar to those used in the experiments in Section 4.3, in which two or three learning agents coexist.

6.4.1 Objectives

The purpose of our experiments is to determine whether socially-aware behaviors carry some benefit for the agents which perform them. We model our experiments in foraging scenarios where a group of agents inhabit some environment having limited resources available at each time step. We model the agents as being rats within a classical “laboratory” type of experiment where to obtain food the agents must first press one or two levers located in the environment.

We note that, in some of the environments, from an individual agent’s point-of-view there is no need for each agent to share the resources with the other members of the social group—its fitness depends *only* on the amount and frequency of consumed resources, as denoted by the fitness-based reward feature in (6.7). In some of the scenarios we model mutual dependencies between the agents, *i.e.*, consuming food resources no longer depends solely on the agents actions but also of their interacting partners.

Our objective is therefore to investigate whether by paying attention to some signals perceived from their social context, agents engage in socially-aware behaviors that trade-off individual well-being in the short-term for the sake of the group's fitness in the long-run. On the other hand, our study can allow us to assert the conditions under which the exchange of social signals between a group of agents becomes a crucial aspect of their mutual survival.

6.4.2 Methodology

As mentioned earlier, our agents are modeled as entities moving in the environment and trying to eat resources at each time step.

Agent Description

In all the scenarios, each agent k has available 2 possible actions, $\mathcal{A}^k = \{Left, Right\}$. We assume that the agents automatically consume food when entering in a cell containing food, except if there is another agent in the same position. In both cases, the directional actions move the agent deterministically to the adjacent cell in the corresponding direction. When available, action *Eat* consumes a food resource if one is present in the agent's current location, and does nothing otherwise. A lever in the environment controls the appearance of food—only when one or both agents enter the lever position the food becomes available. Whenever an agent consumes a food resource, it becomes full for one time step, after which it returns to the satisfied state. An agent automatically becomes hungry if it does not consume any resources for β_{mh} number of time steps, depending on the specific environment. As described earlier, the fitness-based reward feature ϕ^k_{fit} for each agent k generally depends on the hunger status of that agent.

At each time step, each agent k is able to observe:

- its current (x, y) *position*;
- its *satiation status*, *i.e.*, whether it is hungry, satisfied or full;
- whether *food* is present at its current location;
- whether *another agent* is present at its current location. There are $K - 1$ binary observations of this kind, K being the number of agents in the environment;
- how many agents have consumed a food resource since B_k last ate.

As with the previous experiments our agents treat such individual observations as if they were states. Importantly, because agents cannot observe each others' positions or hunger status, from the perspective of each agent the environment is non-Markovian, since there are elements of the underlying state which the agents cannot observe that are needed for them to perform optimally.

Once again we resorted to prioritized sweeping (Moore and Atkeson, 1993) to learn an approximate (single-agent) model of the environment that it then uses to compute an individual policy, according to the its (perceived) reward function $\hat{r}^k$. In our implementation, we used a learning rate $\alpha = 0.3$, a discount factor $\gamma = 0.9$, a backup limit of 10 state-action pairs and a minimum priority threshold of 10^{-4}. Each agent follows an ε-greedy policy with a decaying exploration rate $\varepsilon_t = \lambda_\epsilon^t$, with $\lambda_\epsilon = 0.999$.

Computing Social Group Fitness

As mentioned earlier in Section 6.3, to determine the optimal reward parameters, $\boldsymbol{\theta}^*$, we adopt a simple search strategy similar to the one described in (Singh et al., 2010) that was already used in the experiments in Section 4.3 and Section 5.3. We sample a set of M different reward parameters, $\{\boldsymbol{\theta}_i, i = 1, \ldots, M\}$, such that $\|\boldsymbol{\theta}_i^k\|_1 = 1$. Specifically, we sampled a total of $M = 4,359$ for 2-agent scenarios and $M = 287,499$ for 3-agent scenarios. For each $\boldsymbol{\theta}_m, m = 1, \ldots, M$, we deploy K agents driven by the (POMG) reward function $\mathbf{r}(\boldsymbol{\theta}_i)$ in each of the test scenarios and run, for each scenario, a total of $N = 200$ independent Monte-Carlo trials of $T = 5,000$ time-steps each, during which the agents are allowed to interact and learn in the environment. The fitness associated with each $\boldsymbol{\theta}_i$ is then computed as

$$\mathcal{F}(\mathbf{r}(\boldsymbol{\theta}_m)) \approx \frac{1}{N}\sum_{n=1}^{N} f(\mathbf{h}_m^n) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{K}\sum_{k=1}^{K}\sum_{\tau=1}^{T}\rho_{t,m}^{k,n},$$

where $\mathbf{h}_m^n$ denotes the nth joint history sampled with $\mathbf{r} = \mathbf{r}(\boldsymbol{\theta}_m)$ and $\rho_m^{k,n}$ is the corresponding fitness-based external evaluation signal for agent k.

6.4.3 Multiagent Scenarios

We ran a total of 9 different experiments, each consisting of a variation of the general problem defined above.[6] All scenarios are episodic: whenever one of the agents consumes a resource, each agent is placed in one of the available "start-positions". For ease of explanation, each scenario is generically identified as "**A-R-S(-L) Scenario**", where **A** is the number of agents, **R** the number of resources, **S** the number of start-positions and **L**, for the Lever Scenarios, is the number of levers available in the environment. We also present the values for β_{hp}, the penalty for being hungry in (6.7), and β_{mh}, the maximum number of time steps an agent can be without becoming hungry.

2-1-1-1 Scenario: The environment for this scenario is represented in Figure 6.3. Only

[6] We refer to Section B.2 where we present a series of multiagent experiments in larger versions of these lever environments. We conclude that despite their relative small size, the "game-like" situations that the environments in this chapter present is what defines the complex multiagent tasks that we need in order to assess the usefulness of our social intrinsic rewards in achieving cooperation.

one agent is needed to press the lever for the food to become available. After eating, both agents return to the start-position (middle cell) as indicated. In this scenario, we set $\beta_{hp} = -0.3$ and $\beta_{mh} = 6$ time steps. ◊

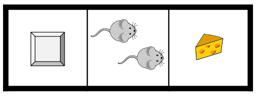


Figure 6.3: Environment used in the 2-1-1-1 Scenario, 2-1-1-1-Heavy Scenario and 2-1-1-1-Electric Scenario. See text for details.

2-1-1-1-Heavy Scenario: This scenario is identical to the previous one, except that the food becomes available only when *both* agents enter the lever position at the same time. $\beta_{hp} = -0.18$ and $\beta_{mh} = 8$. ◊

2-1-1-2 Scenario: The environment for this scenario is depicted in Figure 6.4. The behavior and parameters are the same as in the previous scenario, only that there are two levers available that must be pressed at the same time by the agents. ◊

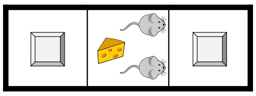


Figure 6.4: Environment used in the 2-1-1-2 Scenario. See text for details.

2-1-1-1-Electric Scenario: In this scenario a food resource is always available in the right cell as indicated in Figure 6.3. The difference is that the environment is ELECTRIFIED unless *one* of the agents enters *and remains* in the lever position, upon which the environment becomes NORMAL. Eating a resource while the environment is ELECTRIFIED decreases the fitness of *both* agents by one value, *i.e.*, $\phi^k_{\mathtt{fit}} = -1$. In this scenario the agents' receive an extra binary observation about the ELECTRIFIED status of the environment. We set $\beta_{hp} = -0.3$ and $\beta_{mh} = 6$. ◊

2-1-2-2 Scenario: This scenario is depicted in 6.5. It follows the dynamics of the 2-1-1-2 Scenario, meaning that the agent that "goes for the right lever" stays closer to the food resource when it becomes available. Also, after eating, the agent remains in the same position (middle-right) while the other is positioned in the middle-left start-position as indicated. We set $\beta_{hp} = -0.18$ and $\beta_{mh} = 8$ for this scenario. ◊

2-2-2-2 Scenario: This scenario, the environment of which is depicted in Figure 6.6, is similar to the previous one, only now *two* resources are provided whenever both agents press the levers at the same time. $\beta_{hp} = -0.4$ and $\beta_{mh} = 4$. ◊

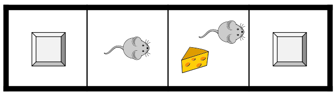


Figure 6.5: Environment used in the 2-1-2-2 Scenario. See text for details.

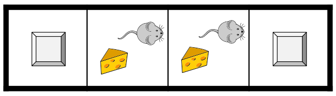


Figure 6.6: Environment used in the 2-2-2-2 Scenario. See text for details.

3-1-1 Scenario: This scenario is represented by the environment in Figure 6.7. In it, three agents, starting from the left cell, compete for one resource located on the right side of the environment. We set $\beta_{hp} = -0.2$ and $\beta_{mh} = 10$. ◊

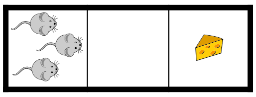


Figure 6.7: Environment used in the 3-1-1 Scenario. See text for details.

3-1-3 Scenario: This scenario, illustrated in Figure 6.8, is identical to the previous one except for the positioning of the agents after eating. We assume that the agents have a specific number associated, from 1 to 3. There are 3 possible start-positions and the agent that consumes the resource starts closer to it, in the middle-right location. The other two agents are then placed in one of the other start-positions, from left to right, according to their number[7]. ◊

3-2-3 Scenario: The environment for this scenario is depicted in Figure 6.9. Three agents, each starting in a different location, compete for the two resources available in the positions indicated. Agents are positioned from left to right according to their number, priority given to the ones that ate. In this scenario, $\beta_{hp} = -0.7$ and $\beta_{mh} = 6$. ◊

6.4.4 Results and Discussion

Table 6.1 summarizes the results of our experiments by comparing for each scenario the mean cumulative fitness obtained by the group of agents using the optimal parameter vector $\boldsymbol{\theta}^*$ against a group of agents each receiving only fitness-based reward. When relevant, we provide charts depicting the evolution of the cumulative fitness of the group.[8]

[7] For example, suppose that Agent 3 just ate. In that case, Agent 1 is positioned in the left-most cell while Agent 2 starts from the middle-left cell, thus closer to the food resource (see Figure 6.8).

[8] Illustrative videos of the observed behaviors in different stages of the learning process in all the scenarios are available online at http://gaips.inesc-id.pt/~psequeira/multi-coop/.

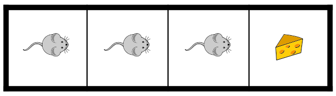


Figure 6.8: Environment used in the 3-1-3 Scenario. See text for details.

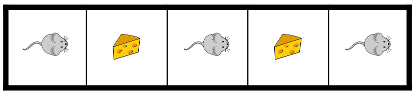


Figure 6.9: Environment used in the 3-2-3 Scenario. See text for details.

Table 6.1: Mean cumulative group fitness attained in each scenario. In each scenario, we indicate the optimal parameter vector $\boldsymbol{\theta}^{*,k}$, and the parameter set corresponding to a group of agents that receive only the fitness-based reward. In all but the 3-1-1 Scenario, $\boldsymbol{\theta}^{*,1} = \ldots = \boldsymbol{\theta}^{*,k} = \ldots = \boldsymbol{\theta}^{*,K}$.

Scenario		$\theta_{\mathfrak{ext}}$	$\theta_{\mathfrak{int}}$	$\theta_{\mathfrak{fit}}$	**Mean Fitness**
2-1-1-1 Scenario	$\boldsymbol{\theta}^{*,k}$	0.25	0.50	0.25	914.7 ± 20.2
	Fit.	0.00	0.00	1.00	−93.5 ± 437.6
2-1-1-1-Heavy Scenario	$\boldsymbol{\theta}^{*,k}$	0.75	0.00	0.25	360.5 ± 16.6
	Fit.	0.00	0.00	1.00	−81.0 ± 85.8
2-1-1-1-Electric Scenario	$\boldsymbol{\theta}^{*,k}$	0.25	0.50	0.25	745.8 ± 21.7
	Fit.	0.00	0.00	1.00	515.2 ± 196.7
2-1-1-2 Scenario	$\boldsymbol{\theta}^{*,k}$	0.25	0.50	0.25	447.6 ± 71.7
	Fit.	0.00	0.00	1.00	−145.5 ± 36.0
2-1-2-2 Scenario	$\boldsymbol{\theta}^{*,k}$	0.25	0.50	0.25	74.9 ± 98.1
	Fit.	0.00	0.00	1.00	−367.8 ± 132.3
2-2-2-2 Scenario	$\boldsymbol{\theta}^{*,k}$	0.00	−1.00	0.00	699.0 ± 176.7
	Fit.	0.00	0.00	1.00	556.0 ± 225.6
3-1-1 Scenario	$\boldsymbol{\theta}^{*,1}$	0.25	−0.25	−0.50	163.5 ± 7.7
	$\boldsymbol{\theta}^{*,2}$	0.50	0.25	0.25	
	$\boldsymbol{\theta}^{*,3}$	0.75	0.00	0.25	
	Fit.	0.00	0.00	1.00	−272.5 ± 24.5
3-1-3 Scenario	$\boldsymbol{\theta}^{*,k}$	0.25	0.50	0.25	209.7 ± 7.2
	Fit.	0.00	0.00	1.00	160.2 ± 8.1
3-2-3 Scenario	$\boldsymbol{\theta}^{*,k}$	0.50	0.25	0.25	1,085.3 ± 43.4
	Fit.	0.00	0.00	1.00	144.1 ± 34.6

Overall Analysis

Generally speaking, our results indicate that socially motivated agents attain greater degrees of fitness as a group when compared to agents using only the fitness-based reward during learning. These agents are implicitly "selfish" and, combined with the structure of the environments, this behavior typically leads one of the agents to starvation. An interesting fact of the experiments is that with the exception of the 3-1-1 Scenario, the optimization procedure discovered similar reward parameters for all agents, *i.e.*, $\boldsymbol{\theta}_1^* = \ldots = \boldsymbol{\theta}_K^*$. This homogeneity in the social group is in line with findings in *concurrent learning*[9] and other team learning schemes stating the importance of the agents' individual skills and the environment's structure for the heterogeneity of the group (Panait and Luke, 2005). In our experiments, because all agents have the same skills (actions) available and interact within the same closed space, competing for the same resources, task specialization cannot emerge, and therefore agents tend to behave the same way so that the group's fitness gets maximized. As will shortly be analyzed in detail, the 3-1-1 Scenario presented a particular coordination challenge with few resources for the number of agents in competition, allowing "sacrifice" or "alliance-like" behaviors to emerge therefore creating heterogeneous groups.

Having More Agents than Resources

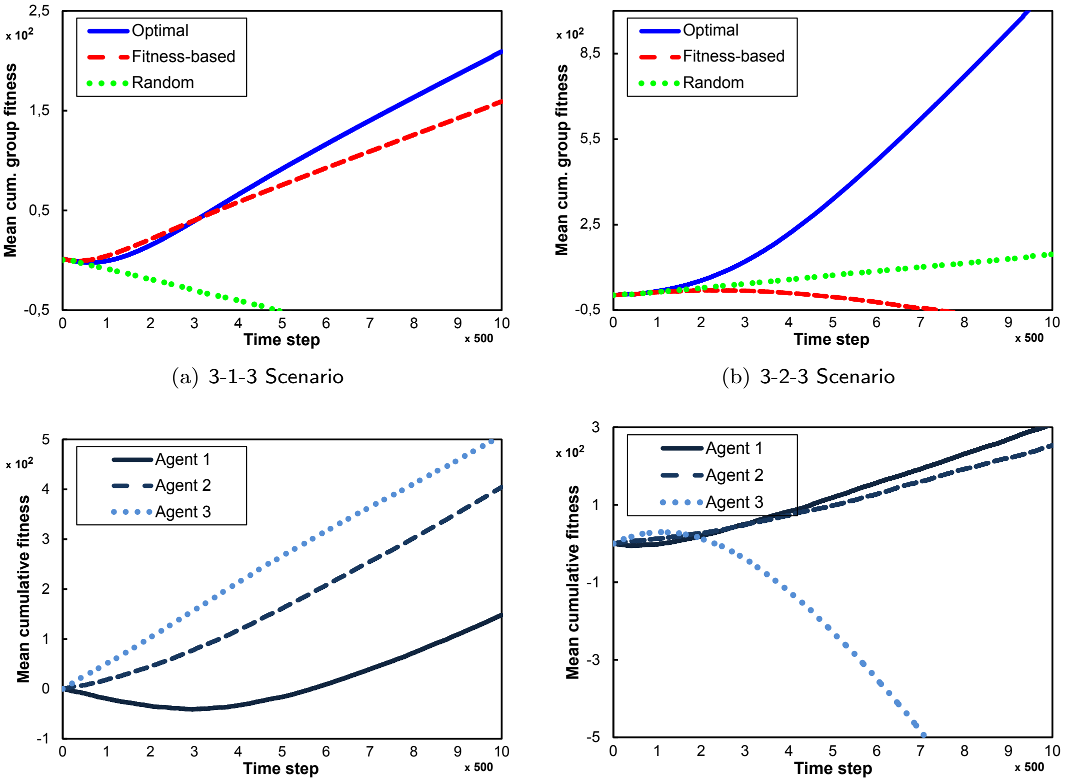
In general, the results of our experiments show that in inherently competitive scenarios, *i.e.*, where the number of food resources available is less than the number of agents, the optimal parameter vector $\boldsymbol{\theta}$ fairly distributes the importance of the three reward features considered. In fact, as indicated in Table 6.1, 5 out of the 9 experiments determined $\boldsymbol{\theta}^{*,k} = [0.25; 0.50; 0.25], k = 1, \ldots, K$, to be the optimal set of parameters. This suggests that the proposed features, inspired by the social legitimacy-signals of (Dörner, 1999), do provide a relevant trade-off between fitness-based reward and the social intrinsic reward features. Each agent takes into consideration not only its own well-being but are also sensible to the social "reinforcement" received for allowing food sharing.

As an example of this effect, coordination under competitive conditions emerged in the 3-1-3 Scenario and 3-2-3 Scenario. As indicated in Figures 6.10(a) and 6.10(b), our social were able to coordinate and share food resources thus achieving a higher degree of fitness as whole group. Figure 6.10(c) illustrates the individual performance of the 3 optimal agents in the 3-2-3 Scenario. As we can see, Agent 1 and Agent 2 learned to take turns to eat while Agent 3 benefited from the positioning policy to be able to consume more food resources. However, by looking at the individual performance of the fitness-based agents in Figure 6.10(d) we see that this positioning policy was actually a disadvantage to Agent 3 as it had more competition with the other agents. Overall, as can be seen from Figure 6.10(b), this deeply impacted the group's performance, especially when compared with the optimal group of agents.

Mutual Dependencies

Figure 6.11(a) shows the performance of the learning agents in the 2-1-1-1 Scenario. We can see that our agents, by means of the reciprocity mechanism, were able to feed in turn despite not having any impediment not to do so, *i.e.*, either agent could press the lever alone and then consume the resource. On the contrary, in the 2-lever scenarios the agents' fitness is dependent on the other agent's actions. Unlike the 2-1-1-1 Scenario, in the 2-1-1-1-Heavy Scenario both agents have to press

[9] In concurrent learning, each agent tries to improve the performance of the group by learning through independent processes (Panait and Luke, 2005).



(a) 3-1-3 Scenario

(b) 3-2-3 Scenario

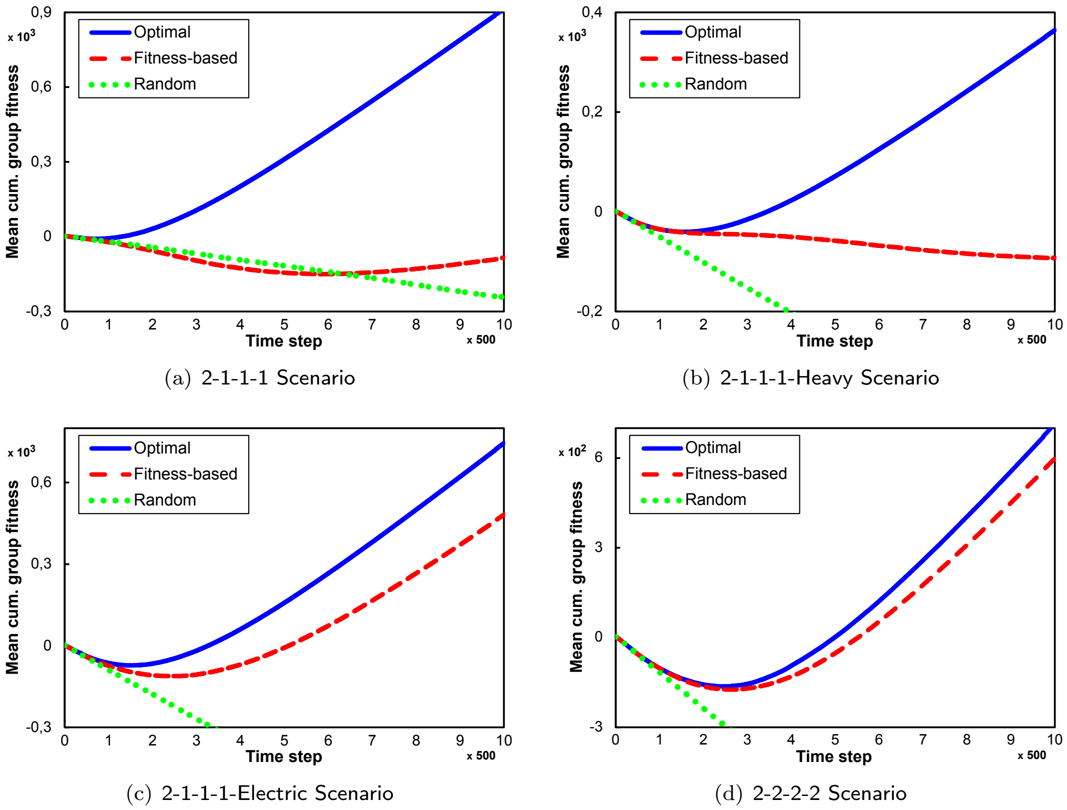
(c) 3-2-3 Scenario “Optimal” indiv. performances

(d) 3-2-3 Scenario “Fitness-based” indiv. performances

Figure 6.10: Evolution of the individual or group fitness in the 3-1-3 Scenario and 3-2-3 Scenario. Results correspond to averages over 200 independent Monte-Carlo trials. “Optimal” corresponds to a group of agents learning with the optimal set of agent parameters, $\boldsymbol{\theta}^*$. “Fitness-based” corresponds to a group of agents receiving only fitness-based reward, and “Random” is a group of agents acting randomly, *i.e.*, receiving no reward at all.

the lever to allow food delivery and as such cannot exploit the other. Despite being unaware of this mutual dependency, our “socially-aware” agents were able to cooperate by learning the mutual press behavior and the reciprocity mechanism allowed them to coordinate food consumption. We note that pressing the lever is a learned, intrinsically motivated behavior that does not have any direct relation to fitness enhancement by itself. As depicted in Figure 6.11(b), the “selfish” agents on the other hand could not learn how to coordinate feeding and as a consequence failed to solve the mutual lever-pressing problem. The 2-1-1-2 Scenario presents a similar problem but in which each agent chooses its own lever to press. As soon as they learned this cooperative behavior, the optimal group was able to outperform the “selfish” group once more, as indicated by the results in Table 6.1.

The 2-1-1-1-Electric Scenario presents a different kind of mutual dependency—the change in fitness for eating a resource depends on whether some agent presses the lever. As depicted in Figure 6.11(c), despite some initial loss during exploration, both the “Fitness-based” and the “Optimal” groups were able to cooperate, going for the lever in turns in order for the other agent to eat. This is due to the fact that in this scenario, pressing the lever *is* related to fitness, as both



(a) 2-1-1-1 Scenario

(b) 2-1-1-1-Heavy Scenario

(c) 2-1-1-1-Electric Scenario

(d) 2-2-2-2 Scenario

Figure 6.11: Evolution of the group fitness in the 2-lever scenarios.

agents received the negative reward associated to the "electric shock". Receiving the negative reward therefore helped the agents towards learning the intended coordination, the difference being that socially motivated agents did it faster.

Having Sufficient Resources

The 2-2-2-2 Scenario provides insights on situations where food resources are "abundant" in the environment. In this scenario there needs to be no food sharing or competition over the available resources—in fact, agents learn to go to one of the levers and consume the closest resource, unaware of the dependence on the other agent to be able to do that. Interestingly, the optimal set of parameters for each agent k values behaviors in which the agents are inconsiderate with each other: as indicated in Table 6.1, $\theta^{*,k}_{\mathtt{int}} = -1$. Because there was no direct competition for food, $\phi^{k}_{\mathfrak{crt}}$ did not pay any relevant role—in fact, if regarded positively, it would prevent the agent from eating in certain situations and impact its individual fitness, with no group benefit. This scenario is an example in which the conditions of the environment, *i.e.*, resource abundance, shaped the agents in behaving ungenerously. Moreover, we can observe from Figure 6.11(d) that without a need for cooperation, the group fitness achieved by either the optimal and the fitness-based agents were very similar.[10]

Emerging Alliances

In the 3-1-1 Scenario and 3-1-3 Scenario our objective was to see whether coordination could emerge in a group where the resources available were scarce for the number of agents therein. The results for solving the ORP for these scenarios showed the emergence of interesting social phenomena by means of our social intrinsic rewards. Like in the previous scenarios, the optimal social group attained a significantly higher ($p < 10^{-4}$) than the group of fitness-based agents, as indicated in Table 6.1.

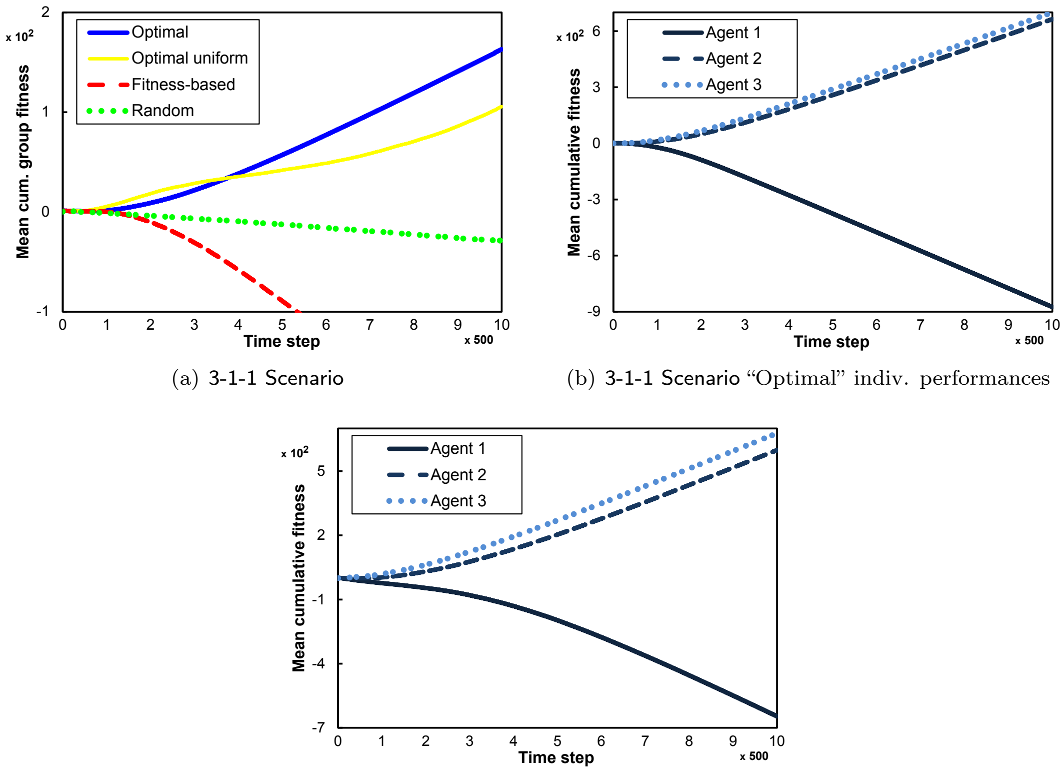
The results obtained in the 3-1-1 Scenario showed the emergence of the only heterogeneous social group within all experiments. The difference to the best uniform parameter vector is depicted in Figure 6.12(a), and the results show a statistically significant difference ($p < 10^{-4}$). An analysis to the individual agent parameters of the optimal social group enables us to see that for the population to thrive, one of the agents had to be "sacrificed" while the other two coordinated to obtain food. Agent 1's reward parameters, $\boldsymbol{\theta}_1^* = [0.25; -0.25; -0.50]$, pushed it to ignore food and therefore starve, leaving the other agents the ability to coordinate in food consumption, as illustrated by the individual performances in Figure 6.12(b). Because the agents started from the same position, they all had equal opportunity to get to the food resource. However, resources were very scarce for coordination to occur and one of the agent's had to ignore fitness enhancement "for the sake" of the social group.

A similar effect occurred in the 3-1-3 Scenario, as shown by the individual performances depicted in Figure 6.12(c). However, in this scenario this was a consequence of the placement policy of the agents, based on their predefined cardinality. In these conditions, Agent 1 has a greater chance of appearing in the left-most cell of the environment, further away from the food. This scenario kind of simulates a "ranked" population in which higher-numbered agents have a greater possibility of increasing fitness. By looking at the optimal parameter set, $\boldsymbol{\theta}_k^* = [0.25, 0.50, 0.25], k = 1, \ldots, K$, we see that a careful consideration of the social and fitness-based aspects of the environment allowed Agent 2 and Agent 3 to form an "alliance" and coordinate their feeding behavior while leaving Agent 1 to starve.

6.5 Overall Discussion

This section contextualizes our approach and results in this paper within social theories analyzing the dynamics of populations as a result of social interactions. We examine how our approach relates to those theories and how our experiments can be analyzed to enable a more formal discussion.

[10] Nevertheless, the difference in mean fitness observed between the two groups of agents in the 2-2-2-2 Scenario, registered in Table 6.1, is statistically significant ($p < 10^{-4}$).



(a) 3-1-1 Scenario

(b) 3-1-1 Scenario "Optimal" indiv. performances

(c) 3-1-3 Scenario "Optimal" indiv. performances

Figure 6.12: Results for the 3-1-1 Scenario and 3-1-3 Scenario.

6.5.1 Related Work

The first thing we want to remark is that within MAS, coordination in the general case implies that the agents' strategies converge to a Nash equilibrium (NE), basically meaning that the agents reach a consensus where none of them benefits from departing from it (Abdallah and Lesser, 2008; Busoniu et al., 2008). To facilitate coordination many multiagent reinforcement learning (MARL) algorithms use explicitly communicated information or implicit shared knowledge between the agents. This often requires that the actions and rewards of the other agents or the underlying game structure (and correspondent NE) are known (Abdallah and Lesser, 2008; Busoniu et al., 2008; Claus and Boutilier, 1998; Sen et al., 1994). Other approaches include taking advantage of the heterogeneity in the agents' knowledge of the task either through expert learning or imitation.

In realistic settings however, such information sharing assumptions are too restrictive and these techniques usually induce exponential growth with the number of agents (Abdallah and Lesser, 2008; Claus and Boutilier, 1998). Coordination is also possible with no information sharing when the actions of the other agents provoke only small perturbations in the environment's dynamics (Sen et al., 1994) or when the gradient of the expected return is known (Abdallah and Lesser, 2008).

Our approach for social intrinsic reward design allows the agents to consider the implicit impact of their behavior on the social group's welfare. Our results show that this mechanism enables them in achieving coordination without any explicit communication or even awareness of the other agent's existence or the impact of their actions in the fitness of others. Also, just like our emotion-based rewards proposed in Chapter 4 or the features emerged from the experiments in Chapter 5, the mechanism behind the intrinsic social rewards proposed in this chapter is domain-independent as long as there is a notion of "considerate" and "selfish" actions within a specific social context.

Moreover, within IMRL, recent work by Liu et al. (2012) already considered the ORP problem in a multiagent setting. However, the focus of the paper was *strong mitigation* in the context of ORP, where the computational effort must be balanced between fitness optimization and reward optimization. In light of the different focus, the multiagent setting considered in (Liu et al., 2012) addresses decentralization in reward optimization and planning, but simplifies considerably the agent model. In particular, they consider $\mathcal{Z}^1 = \ldots = \mathcal{Z}^K = \mathcal{S}$ (all agents have full state observability), $\rho^1 = \ldots = \rho^K$ (all agents receive the same external "fitness-based" evaluation signal) and the environment parameters are known. Moreover, although action selection is decentralized, all agents observe the actions of the other agents once they are performed. In this sense, the decision problem faced by our agents is significantly harder.

6.5.2 Evolutionary Game Theory

In the experiments within this chapter, and given the evolutionary interpretation proposed for the IMRL framework and the social nature of the scenarios considered, there is a close relation between our approach and *Evolutionary game theory* (EGT) (Maynard Smith and Price, 1973). EGT, as the name indicates, is an extension to *game theory* (GT)[11] (von Neumann and Morgenstern, 1944) that tries to explain how a population of strategies evolves over time by merging ideas from ecology and economics (Sigmund and Nowak, 1999; Tuyls and Nowé, 2005).

EGT was formulated to address games in biological settings, where individuals are unable to take hyper-rational decisions but are still able to optimize their behavior to increase their chances of reproduction, *i.e.*, their fitness. EGT follows the approach on evolution by considering a population whose members are *players*, *strategies* are phylogenetic traits and fitness is measured as the game's resulting *payoff* (Sigmund and Nowak, 1999). As opposed to traditional GT, games in EGT are played repeatedly between the members of the population each following some strategy. In its simplest form, EGT analyzes the dynamics of a population when invaded by a small group of mutants playing some strategy. The interaction between individuals (mutant or non-mutant) is modeled as a strategic game in which the payoffs measure the fitness of the individuals after each interaction (game). Game theoretic notions are then used to analyze the dynamics of the fitness of the population and predict whether a given strategy will eventually extinguish or not (Axelrod

[11] Game theory (von Neumann and Morgenstern, 1944) addresses problems as *games* between decision makers (the *players*) who have to choose between different *strategies* whose *payoff* depends on the decisions of other players. To choose between the strategies, the players take into account their own preferences and the decisions of others.

and Hamilton, 1981; Maynard Smith, 1982). When a population using some strategy cannot be invaded by a rare mutant adopting a different strategy, the first is referred to as an*evolutionary stable strategy* (ESS) (Maynard Smith and Price, 1973).

In the context of our study, each parameter vector $\boldsymbol{\theta}$ used by the agents within a social group influences the *strategy* used by the agents in the environment by focusing in different aspects of their social context. For example, a parameter vector $\boldsymbol{\theta}^{\text{fit}} = [0, 0, 1]$ will drive the agents into a "selfish" strategy by focusing only on fitness-based rewards. As such, the optimal parameter vector $\boldsymbol{\theta}^*$ obtained after the optimization procedure can be seen as a *stable equilibrium* in a population of siblings (*i.e.*, having the same "genetic characteristics"), as discussed in (Bergstrom, 1995).

6.5.3 Reciprocation Factors

The results of the 2-1-1-1 Scenario are in line with theories about the importance of a proper reciprocation mechanism for the maintenance of cooperation (Axelrod, 1984; Falk and Fischbacher, 2006). In this scenario, both agents were trying to compete for the same resource and could easily exploit the other by waiting for it to press the lever and then consume the resource. In such cases, "greedy" agents cease to cooperate by not receiving any kind of reinforcement for the behavior. On the contrary, by examining the agents' behavior in the optimal case we can observe a "socially-aware" strategy in which the agents feed in turns.

Discrimination is one of the abilities allowing individuals to engage in cooperation by reciprocation (Hamilton, 1964). Distinguishing interacting partners allows for different degrees of relatedness to evolve throughout time, therefore influencing the marginal *inclusive fitness* gained by an altruist during cooperation. In our study we consider that all agents belong to the same social group, thus meaning their relatedness and probability of future interactions is (implicitly) high. Although we do not model an explicit memory about past interactions with other agents of the group, the fact that each agent is able to distinguish the presence of others in its current location makes that distinct strategies towards different agents can emerge. This fact could explain the appearance of the "alliances" reported in Section 6.4.4. Two of the agents acquire a cooperative strategy (*i.e.*, resource sharing) between each other but not with the third agent due to their implicit "relatedness" being higher—recall that one of the agents had a greater chance of being farther from the food resource, and as such it was less likely to have encounters with the other agents.

6.5.4 Social Pressures

Another aspect of our approach relates to the *altruism* vs *social pressure* dualism in motivating selfless behavior (DellaVigna et al., 2012). For example, pure charity is seen as utility-maximizing in individuals that simply enjoy giving, while others feel obligated to doing so in the presence of others because of social pressure (DellaVigna et al., 2012). In our model, pure altruism is somehow encoded within the internal social reward feature, rewarding "charitable" behaviors for the sake of the social group. On the other hand, social pressure is encoded by the external social reward feature, directly rewarding and punishing behaviors of an agent when in the direct presence of other agents.

Our experiments somehow confirm the importance of *social pressure* in influencing the behavior of altruistic individuals (Becker, 1974; DellaVigna et al., 2012). The difference in strategies attained in the 2-1-2-2 Scenario relative to the 2-2-2-2 Scenario shows this dual effect. By having sufficient food resources in the 2-2-2-2 Scenario, the agents did not have to share resources and could exploit the environment at their will. As such, an agent not eating in the presence of a food resource *is not contributing* in any way for the well-being of the other agent. In fact, being altruistic under those conditions has no utility for the "selfless agent", and therefore being "selfish" is the best global strategy in this scenario. On the other hand, competition for scarce food resources shows the importance of "social pressure", by punishing situations where both agents disputed the same food resource. In conclusion, it seems that by being able to (externally) exchange social signals and having an internal mechanism representing an ideal of "social standards", our agents are able to achieve cooperation in resource-sharing contexts.

6.6 Contributions

In this chapter we extend the IMRL framework and ORP formulation described in Chapter 2 to multiagent settings. We develop social intrinsic reward features for IMRL agents that take into consideration social interactions among agents, inspired by the notion of *group affiliation* from MAS research (Bach, 2009; Dörner, 1999).

6.6.1 Implications for RL/IMRL

We formally develop an extension of the IMRL framework for multiagent settings. This involved defining the following concepts within multiagent IMRL:

- a *social group of agents* acting within the same environment where each agent has access to only a partial, individual view over the shared environment;
- each agent is considered a self-interested, *individual learner* trying to compute its own optimal policy with respect to its observations and its individual *social intrinsic reward function*;
- a new measure of *group fitness* within the ORP formulation that takes into account the joint history of the interacting agents.

We propose two social reward features for IMRL based on *legitimacy signals* from Psi-theory (Bach, 2009; Dörner, 1999) and a *reciprocity* mechanism (Axelrod, 1984; Falk and Fischbacher, 2006; Trivers, 1971) in the context of resource-sharing scenarios. Namely, we propose:

- an *external social reward feature* rewarding and punishing individual agent's feeding behaviors when in the presence of other agents that can be related to the kinds of social pressures individuals are subject to within their social groups;
- an *internal social reward feature* rewarding and punishing individual agent's feeding behaviors independently of the presence of other agents that relates to a notion of altruism or self social standards.

We show that the IMRL framework can be used to endow agents with *social motivation* provided by the intrinsic reward features that drives them to learn "socially-aware" behaviors that benefit the group of agents as a whole;

- Particularly, we show that it is possible for agents to *individually* acquire *socially aware behaviors* that trade-off individual well-fare for social acknowledgment, leading to a more successful performance of their social group.

6.6.2 Implications for Multiagent Systems

We now analyze the impact of our approach and the results of our experiments for multiagent research. Our approach relates to EGT in that the agents' parameterization encodes the *strategies* that they will follow during learning within the social context.

- Recall that in almost all of the multiagent scenarios used in our experiments, the agents of the optimal group shared the same parameter configuration. Because all the agents are influenced by the same social strategy, our social groups of agents relate to *populations of siblings* in nature (Bergstrom, 1995);
- Although tested in simple multiagent scenarios, our formulation of social rewards allow for the investigation of *emerging stable equilibria* within social groups of agents.

The results of our experiments are in accordance with social theories claiming that altruistic and cooperative behaviors can thrive in populations with high relatedness and in which reciprocity is possible (Axelrod and Hamilton, 1981; Falk and Fischbacher, 2006; Trivers, 1971). Specifically, our approach supports:

- a high *relatedness* between the agents within a social group due to the high probability of future encounters between them;
- a *reciprocity mechanism* that is able of rewarding and punishing behaviors that are considerate (or not) for the well-being of the social group that is based on the exchange of social signals.

By being able to both exchange social signals between each other and having an internal mechanism representing an ideal of "social standards", our agents are able to achieve cooperation in resource-sharing contexts even in the presence of mutual-dependencies for their fitness. This means that in the context of MAS our model is able to incorporate:

- the effects of *pure altruistic behaviors* by means of the influence of the internal social reward feature on the agent's behavior;
- the effects of *social pressure* by means of the influence of the external social signals (implicitly) exchanged between the agents during learning.

Moreover, our model does not require that the agents share any information about the task or structure of the environment.

- Particularly, our method does not require any explicit communication or that the "game" dynamics behind each scenario are known.
- Our agents are also unaware of the existence of other agents or the impact of their actions in for their fitness.

6.7 Summary

In this chapter we investigated the impact that a social motivation system can have in the emergence of socially aware behaviors in groups of learning agents. The framework for IMRL was extended to multiagent settings in order to explore social reward features inspired by the notion of social legitimacy and a need for affiliation proposed in the Psi-theory (Bach, 2009; Dörner, 1999). The results of our experiments in multiagent environments show that, indeed, the socially motivated agents as a whole perform much better than "selfish" agents only focusing on their own well-being, while having little impact in their individual fitness. Also, the results show that even in the presence of dominating agents, *i.e.*, agents that do not require socially-aware behavior to maximize their individual fitness, socially-aware behaviors lead to an improved social group fitness. Finally, the results show that the social motivation does not blindly lead to selfless behavior: in scenarios where resources abound, the agents learn to disregard social legitimacy within their social context, since its actions do not impact the fitness of others.

The next chapter summarizes the main contributions stemming from our approach throughout this thesis and projects possible future developments.

CHAPTER 7

Conclusions and Future Work

In this chapter we summarize all the work developed throughout this thesis and the contributions stemming from it, and also propose some research directions that can be followed.

7.1 Conclusions

In this section we summarize the main results stemming from Chapters 4, 5 and 6 and relate them with our problem and hypothesis formulated during Chapter 1.

Recall that the behavior of RL agents is guided by a reward mechanism embedded into the agent by its designer. The analogy made in Section 1 parallels the design of reward functions with robot training procedures—just like training a robot to perform some desired task, designing flexible reward mechanisms, *i.e.*, capable of guiding the agent in learning the task intended by its designer, is a *very demanding endeavor*. On one hand artificial agents have inherent limitations—particularly, they cannot perceive *everything* from their environment in order to perform optimally therein. On the other hand, traditional approaches to RL pose elegant solutions that are however too restrictive given the agents limitations, potentially leading to poor performances. Therefore, applying RL in complex problems often requires a great amount of manual fine-tuning on the agents so that they *perform well* in a given scenario. This demand grows even more when we want to design agents that are able to perform well in a variety of different situations, often involving complex interactions with other agents.

Let us now reconsider our hypothesis to address the aforementioned challenges within RL that we introduced in Section 1.3:

> In this thesis we take inspiration from information processing mechanisms shaped throughout evolution that are behind the adaptive success of humans and other biological organisms. We focus on the role of *emotions* and also on the way individuals *interact and cooperate with each other* as a social group to design more *flexible* and *robust* reward mechanisms that enhance the *autonomy* of RL agents in both single and multiagent settings.

Throughout this thesis, we conducted a series of studies targeted at developing mechanisms that are able to mitigate the mentioned problems by following the above stated approach. By being a predominantly multidisciplinary approach, our results have implications to different computational areas. We now summarize the contributions detailed in Sections 4.5, 5.5 and 6.6 in light of these different areas.

7.1.1 RL and IMRL

In general, our approach contributed to the area of RL by alleviating common limitations with learning agents and also modeling effort by the agents' designers. We follow the IMRL formulation by designing reward features that fit the notion of intrinsic rewards, thus providing motivation for behaviors not directly related to the task intended by the agent's designer. Furthermore, the results of all the conducted experiments within IMRL and the analyses performed support our claim that by following our approach we end up building more *autonomous*, *robust* and *flexible* agents.

Autonomy

As discussed earlier, *autonomy* refers to the ability of an artificial agent operating without the intervention of external assistants, especially without that of humans. It thus implies that the less intervention and fine-tuning an agent needs to be able to perform well a given task, the more autonomous it is. Recall that the main technical contribution in Chapter 4 is a set of *four domain-independent emotion-based reward features*, namely *novelty*, *valence*, *goal relevance* and *control*. When tested in a variety of scenarios each posing distinct challenges for a learning agent in the context of IMRL, the use of these reward features *alleviates the need for the agent designer to handcraft reward functions for a specific domain*, thus making the agents more *autonomous*.

We again note that having to determine the correct parameter vectors or the best GP combination to discover the optimal reward function for a specific scenario does not reduce the autonomy of our agents and should not be seen as a limitation of our approach. In parallel to what occurs with natural organisms, agents accustomed to a certain type of environment should perform well in similar environments which dynamics and challenges offered do not vary much from those of the original task. For example, in the Hungry-Thirsty Scenario experiment in Section 4.3.3, the "challenge" is that the agent must first drink before eating, independently of the resources locations, so the strategy that was learned by the discovered "optimal" agent allowed it to cope with the change in position of the water and food on the several environments. On the other hand, it is expected that when faced with environmental characteristics contrasting with those where learning took place, agents do not perform so well (even when comparing with the "standard" RL agent), just like animals are sometimes unable to survive to sudden and drastic changes of their habitat.

Therefore, our focus in this thesis is in the *design* of the rewards and the *quality of the behaviors* and *autonomy* that emerge from their practical use. In a sense, we can say that a greater concern is placed on the *end result* rather than the means that lead to it, what has been termed "weak mitigation" (Bratman et al., 2012). Nevertheless, we acknowledge that the time optimizing the parameters/discovering the GP reward functions must be taken into account when aiming at practical solutions. When that is the case, other methods exist that take into account computational resources limitations of the agents and make use of available information to improve the agent's reward function online (Bratman et al., 2012; Sorg et al., 2010b). We note that our implementation of the domain-independent features is compatible and can be used in conjunction with such solutions in order to reduce the time used to search for good reward functions, in order to achieve "strong mitigation".

Robustness and Flexibility

By *flexible* and *robust* we refer to agents that are able to operate in a wide variety of different situations without having to specifically program them for each possible problem encountered. In Chapter 5 we emerged *domain-independent, general purpose reward features* by using a GP algorithm departing from a set of variables *summarizing the agent's history of interaction with its environment.* Together with our emotion-based approach in Chapter 4, our results show that by using domain-independent reward features evaluating the agent's history with its environment, our agents were able to not only mitigate perceptual limitations but also to perform well in a variety of environments, including non-stationary environments or complex environments providing ambiguous outcomes. Moreover, our rewarding mechanisms are flexible enough to be used in conjunction with any RL algorithm and are independent of the state-representation used or the exploration strategy employed.

Of particular interest for our approach is its close parallel to the development of individuals in nature, whose reward mechanisms are optimized through evolution to allow optimal performance in particular environments. By using reward functions based on natural processing mechanisms, as emotions are (Chapter 4), or evolutionary computation algorithms to discover reward functions (Chapter 5), our agents came to benefit from the same kinds of survival tools found in nature, thus being able to adapt to different situations without having to embed in them specific domain knowledge.

Multiagent IMRL

In Chapter 6 we develop an extension of the IMRL framework for multiagent settings, which involved defining the concepts of *social group of agents* acting within competitive resource-sharing environments and a new measure of *group fitness* within the ORP formulation that takes into account the *joint history* of the interacting agents. We also consider our agents to be *self-interested*, *individual learners* in the sense that each learns according to its own *partial observations* over the environment and its individual intrinsic reward function. In order to validate our approach for multiagent IMRL we developed two *social reward features* based on social theories about the way humans approve/reprehend considerate/selfish behaviors within their social context. We tested our solution in resource-sharing scenarios and our results show that "social IMRL agents" are able to learn behaviors that trade-off individual benefit for their *social group's well-fare.*

7.1.2 Affective Computing

A great part of our approach focuses on *emotions* and the benefits that they bring to natural organisms. It is therefore natural our results to have implications for the field of AC and related areas. Generally speaking, our experiments and results in Chapter 4 reinforce previous approaches within AC advocating the benefits of adapting emotion-based mechanisms to enhance the perceptual and information-processing capabilities of artificial agents. We however depart from previous works by creating an emotion-based reward mechanism that *does not interfere* in any way with the RL algorithm used by the agent. We also did not focus in any particular set of basic emotions to influence the behavior of our learning agents. We rather provide them with a general purpose mechanism inspired by *appraisal theories of emotions* and let an optimization procedure to discover what combinations of appraisals are best for a given situation, in a similar manner to what occurs in nature, where emotional mechanisms are shaped by evolution to allow adaptation to the environment.

The work on Chapter 5 further reinforces the role of emotions as a *fundamental complement* to the agent's information-processing capabilities. Specifically, we proposed a novel method for assessing the significance of *embedding emotions into artificial agents* that, unlike previous approaches within AC, uses an evolutionary computation mechanism for the emergence of emotion-like informative signals. Once emerged, we discovered that those signals have dynamics and evaluative characteristics matching some common *appraisal variables*, and that by using them as reward features provides learning agents a *general-purpose attention-focusing and guiding and mechanism*, as emotions are in nature. The analogy that can be made between the way the emotion-like reward features emerged and the way emotions evolve in nature contributes to the idea that the inclusion of emotions in artificial agents may have a greater impact for their performance than before thought.

7.1.3 Multiagent Systems

Our approach in Chapter 6, besides extending the IMRL framework into multiagent scenarios, contributes to research to MAS in general and also related areas. Our approach relates to research within EGT by having the parameterization of socially-inspired reward functions influence the strategy used within a social group. A formal study about the dynamics of reward function optimization may allow for the investigation of *emerging stable equilibria* within the social group. Moreover, we model our agents as closely related individuals, *i.e.*, sharing the same parameterization and making their interaction frequent during learning, and having a *reciprocity* mechanism, by means of the external social reward feature. The results of our experiments in resource-sharing scenarios thus confirm previous research within EGT claiming that *altruistic* and *cooperative* behaviors can thrive in populations with high relatedness and in which reciprocity is possible. Furthermore, having agents that can *exchange social signals* and have internal mechanisms representing an ideal of *social standards* enables an interesting parallel with what occurs in nature, where both *pure altruism* and *social pressures* influence individuals in engaging in cooperative, unselfish behaviors.

7.2 Future Work

In this section we discuss several possible directions for future research stemming from our results.

7.2.1 Single Agent Research

Let us start by analyzing future developments to our single agent models, especially relating computational aspects to RL/IMRL and emotion research in AC.

More Complex Applications

An interesting development to our approach is to apply the proposed reward features, both in Chapter 4 and 5 in more practical and complex domains. In Chapter 4 each of the scenarios our experiments were designed in order to assert the utility of appraisal-based mechanisms embedded in learning agents, so each scenario stressed the utility of one or more emotional reward features. In Chapter 5 the emerged reward features had a different purpose—we were trying to examine the emergence of signals having emotional properties. Although we did test the features in the more complex Pac-Man scenarios in Section 5.3.2, it would be interesting to test our approach in more practical problems aside from game-like or grid-world domains, *e.g.*, robotic-control due to our connection with biology. There are also well-studied problem domains within the POMDP researchfield in which our reward features could be tested (*e.g.*, Cassandra, 1998).

Increasing Adaptability

As we noted throughout this thesis, our focus on reward design makes us abstract from the particular algorithm that is used to discover the combination of feature weights or the genetic program that leads to the optimal reward function. Nevertheless, it could be useful to impose limitations in the agent's learning/decision-making capabilities and therefore compare the solutions discovered under those conditions with those emerged from our offline optimization procedures. In that perspective we could assess under which circumstances the "computationally-bounded" solutions would allow the learning agent to solve the complex tasks presented in our proposed scenarios.

On the other hand, the quality of the optimized reward functions can also be assessed by the *strategy* that it allows the agent to learn in some scenario. In order to evaluate whether a reward function optimized for a specific environment is indeed domain-independent one could test whether an agent, interacting with a different environment whose task relies on a similar strategy to be solved, is able to learn such strategy and have a performance comparable to the optimal reward function for that environment. Consider for example the Persistence Scenario experiment in Section 4.3.4, where the agent learned that it should "persist" in taking the *Up* action in order to cross the fence and achieve the desired reward provided by the hare. Because the optimized reward function allowed the agent in learning this "persisting" behavior, it would be interesting to discover other scenarios where such strategy would also be useful. One could envisage environments where the fence is placed elsewhere in the environment, or the structure of the environment is different, or even where the number of actions required or the values provided each type of reward, and then evaluate whether the same reward function parameterization allows the agent to learn the required task. In the same manner, it would be interesting to asses how the performance of the agent is affected according to the changes in the environments, or how easily it is to cope with such changes, using for example an online optimization procedure similar to those above-mentioned.

Comparing with Other Approaches

Another aspect that can be addressed about the usefulness of the single agent reward features has to do with the comparison of their performance against the fitness-based reward function defined by the agent's designer. Recall that both in the foraging scenarios of Section 4.3 and the Pac-Man scenarios in Section 5.3.2 we defined a single fitness-based reward function for each scenario that relied on aspects of the scenarios themselves, *e.g.*, the number of preys eaten in foraging scenarios. There are some ways in which we could assess the impact of our reward features in the performance of the learning agents.

For example, one option is to compare the performance of our agents with limited perception using the optimized reward functions against that of *fully-observing* agents using the fitness-based reward function having access to the underlying MDP for each scenario. We could also compare those performances with that of *reward shaping* when applied in the above conditions, *i.e.*, with the fitness-based reward function learning on the MDP. An obvious examination that can be made is to consider other domain-independent reward features proposed in the IMRL literature and compare the performances with our emotion-based and GP-generated features in a variety of scenarios. Recall that we define domain-independent reward features throughout our studies, in contrast with the majority of previous approaches within IMRL proposing domain-dependent features. We could therefore also expand the set of reward features available in each experiment by adding these domain-dependent features and then examine the results of the optimization procedure to assess the impact (relative contribution) of our features in the optimal reward function attained.

Research on Emotions

Within the area of AC, an interesting development to our study performed in Chapter 5 is the comparison between the emerged reward features and human data relating emotions. Recall that for the examination of the "emotional tone" of the emerged features we analyzed the types of evaluations they make with those performed by common appraisal variables, according to appraisal theories of emotions. As suggested in (Gratch et al., 2009), one way of validating appraisal-based computational models of emotions is to compare the predicted emotions generated by the model in typified emotion-eliciting situations with human studies relying on self-reported data. Although in our studies we never build an explicit computational model of emotions, in Section 4.4.4 we discussed how we could generate emotion labels or determine the emotional state (*e.g.*, happy, sad, afraid, etc.) of the agents based on the values of the emotion-based reward features at each time step. Having this extension to our model would thus permit us to better analyze emotional nature of the reward mechanisms.

7.2.2 Multiagent Research

The work developed in this thesis, especially our study in Chapter 6, opens the door to future investigations within MAS.

Emerging Social Reward Features

One of the aspects of our socially-inspired learning model defined in Section 6.3 is that it fosters heterogeneous social groups, *i.e.*, in which each member uses its own reward function to learn in the environment. By extending the ORP formulation to multiagent settings, we are no longer trying to optimize *one* but a *combination* of agent reward functions. In the experiments described in Section 6.4, the optimization procedure discovers the best combination of parameters guided by a fitness function evaluating their joint performance. We used a linearly parameterized approach to perform an exhaustive search over the possible combinations of parameters for each agent and select the one maximizing the overall fitness measure of the group.

A natural step to take is to extend the approach used in Chapter 5 to multiagent settings in order to discover interesting domain-independent social reward features. Besides the primitive sources information used to discover interesting single-agent reward expressions, one could include information shared by the agents when they encounter each other. A possible solution for the discovery of the reward functions is to consider coevolving cooperative algorithms (CCEAs) for MAS (*e.g.*, Colby and Tumer, 2012) that consider evolutionary approaches to optimize cooperative populations of agents. In this manner, each agent would evolve its own reward function with respect to the group's fitness measure being optimized. Following such approach could allow for a richer analysis of the emergence of cooperative behaviors, *e.g.*, by analyzing which factors of its own/shared history of interaction make for the appearance of cooperation in highly competitive scenarios.

Enhancing Social Simulations

One of the appealing features of our social model is the ability to test the emergence of interesting interactions between the individuals and being able to analyze them in the perspective of different social theories (*e.g.*, see the discussions in Section 6.5). The main objective with our experiments in Chapter 6 was to test the emergence of cooperative behaviors in simple resource-sharing scenarios.

There are however some developments to the current experiments that could make our social simulations more interesting. For example:

- our current simulations do not model explicit cooperation, *i.e.*, our agents do not explicitly choose (by means of action selection) to cooperate with other members of their group, we rather examine whether they are considerate with each other in terms of resource sharing. Having explicit cooperative behaviors (*e.g.*, in the sense of the prisoner's dilemma (Rapoport and Chammah, 1965)) can allow us to perform a more formal analysis of the impact of our social reward features in the behaviors of the agents and group fitness attained, and also formally relate the optimized strategies as evolutionary stables strategies as defined within EGT (Maynard Smith and Price, 1973);

- we can also design more complex scenarios, meaning larger environments with more (and more complex) agents interacting with each other. Specifically, we can extend our simulations so that the agents have more behaviors available where each agent can contribute to the enhancement of the social group in different ways. Together with the development of CCEAs, we could observe the emergence of *specializations* among populations of agents and thus yield more elaborate interactions therein.

Emotions in MAS

Recall that in Chapter 4 we proposed a set of emotion-based reward features inspired by appraisal theories of emotions. Throughout evolution emotions evolved due to their adaptive advantage in social contexts (de Waal, 2008; Oatley and Jenkins, 2006). As discussed in Sections 4.2 and 5.4.2, we did not consider the so-called "social dimensions" in our models because single agent scenarios did not allow the agents to take considerations about their social context. Having a more formal and complex social model would then allow us to integrate the emotion-like reward features into the multiagent simulations. In turn, this would allow us to assert the impact of emotions for the cooperation of groups of agents, thus extending existent work in MAS (*e.g.*, Nair et al., 2005).

7.3 Thesis Summary

In this thesis we addressed problems within the computational framework of RL. We adopted the IMRL framework as a testbed for our approach inspired by information processing mechanisms commonly found in natural organisms. We specifically focused on the importance of emotions as an evaluating and coping mechanism for humans and other animals and also on how people interact with each other within their social context.

In Chapter 1 we introduced and motivated the general problem we were trying to solve and advanced our hypothesis on how to solve it. Chapter 2 described the computational framework of

RL and IMRL and detailed the technical challenges that each pose, thus posing open questions to be addressed throughout the thesis. Because most of the contributions herein relate to emotions, Chapter 3 provided theoretical background on the importance of emotions in nature and introduced the area of AC, describing related research on using emotions to improve the performance of learning agents. The following three chapters detailed the work from which our contributions result: in Chapter 4 we introduced our approach for emotion-based reward design inspired by the way humans and other animals appraise their environment in nature; in Chapter 5 we examined the importance of emotions for the performance of learning agents by analyzing their emergence within IMRL; in Chapter 6 we provided the first steps towards extending the previous results into MAS within IMRL by designing reward inspired by the way humans signal socially-aware behaviors.

Having into account the results of our experiments and all the discussion throughout the thesis, we conclude that by embedding into artificial agents processing mechanisms that relate to the way humans and other animals interact with their natural environment, agents come to benefit from the same kinds of survival and adaptive skills that have been shaped by evolution since the origins of time. We therefore contributed to build more autonomous, robust and flexible learning agents.

APPENDIX A

Associative Metric for Factored MDPs

In the main chapters of the thesis we have discussed some approaches to reward design based on natural processing mechanisms. In this chapter we take a different approach by considering a mechanism based on classical conditioning that alleviates some of the problems discussed in Section 2.3.1. Classical conditioning is a behaviorist paradigm that allows organisms to acquire predictive associations between stimuli in the environment whenever co-occurrences between them are frequent. In this chapter we propose a novel *associative metric* based on classical conditioning that, much like what happens in nature, identifies associations between stimuli perceived by a learning agent while interacting with the environment[1].

We use an associative tree structure to identify associations between the perceived stimuli and measure the degree of similarity between states in RL scenarios. Our approach provides a state-space metric that requires no prior knowledge on the structure of the underlying decision problem and which is learned online, *i.e.*, while the agent is learning the RL task itself. We combine our metric with Q-learning, generalizing the experience of the agent and improving the overall learning performance. We illustrate the application of our method in several problems of varying complexity and show that our metric leads to a performance comparable to that obtained with other well-studied metrics but which require full knowledge of the decision problem. We conclude the chapter by analyzing the impact of our metric in typified conditioning experiments, showing that combining our associative metric with standard TD(0) learning leads to the replication of common phenomena described in the classical conditioning literature.

A.1 Introduction

Associative learning is a paradigm from the field of behaviorism that posits that learning occurs whenever a change in behavior is observed (Anderson, 2000). It comprises a set of simple mechanisms that allow animals to avoid harm and seek resources, and even simpler organisms to adjust their behavior according to associations detected between events in the environment (Anderson, 2000; Cardinal et al., 2002). Classical conditioning is one of the best-known associative learning paradigms. It is one of the most basic survival tools found in nature that allows organisms to expand the range of contexts where some of their already-known behaviors can be applied. By associating co-occurring stimuli from the environment, the organism can activate innate phylogenetic responses (*e.g.*, fight or flight responses) to new and previously unknown situations (Anderson, 2000).

In this chapter we leverage associative mechanisms based on the classical conditioning paradigm to RL problems. As discussed in Section 2.2.3, many classical RL methods, such as Q-learning, allow the agent to successively estimate how good each action is in every state, eventually conveying to the agent the information necessary to select only the best actions in all states. This typically

[1] Part of the contributions within this chapter can be found in (Sequeira et al., 2013).

requires the agent to experience *every action* in *every state* a sufficient number of times (Sutton and Barto, 1998). This need for "sufficient" visits to every state-action pair is often impractical, particularly in large environments, and several general approaches have been proposed to mitigate this need, relying mostly on function approximation (Szepesvári, 2010) or state abstractions (Li et al., 2006).

However, certain scenarios present some particular structure that can be leveraged by the learning algorithm to improve the learning performance—namely, by alleviating the requirement of sufficient visits to every state-action pair. For example, in scenarios where the state is described by a finite set of state-variables (*i.e.*, where the state is *factored*), it is possible to leverage this structure to improve efficiency of RL methods (Guestrin et al., 2002; Kearns and Koller, 1999).

Our approach follows the classical conditioning paradigm and identifies associations between stimuli perceived by a learning agent during its interaction with the environment. Given a learning scenario with a factored state space, we use a pattern mining technique to build an *associative tree* that identifies the occurrence of frequent *patterns* of state-variables (henceforth referred as *stimuli*) (Sequeira and Antunes, 2010). These associations are similar in spirit to those that natural organisms identify in their interaction with the environment, and are used by the agent to build an *associative metric* that identifies two states as being "close" if they share multiple/frequent stimuli. This metric is learned online and then combined with Q-learning. The proposed method improves the learning performance of our agents by using current information to update the Q-values of states that are considered *similar* according to the associative metric.

The main contribution of our approach is to provide a general-purpose state-space metric that requires *no prior knowledge* on the structure of the underlying decision problem. Furthermore, the associative tree and the similarity metric are both learned online, *i.e.*, while the agent is learning through successive interactions with its environment. We illustrate the application of our method in several factored Markov decision processes (MDPs) of varying complexity and show that our metric leads to a performance comparable to that obtained when using well-studied metrics from the literature (Balkenius and Morén, 1998; Ferns et al., 2004). Finally, and because we draw inspiration from the classical conditioning paradigm, we further explore the validity of our approach by combining the associative metric with the TD(0) algorithm (Sutton and Barto, 1987) in typified conditioning experiments. We analyze the appearance of several learning phenomena associated with classical conditioning which are observed in nature (Balkenius and Morén, 1998).

A.2 Classical Conditioning

While measuring the salivation level of dogs when tasting meat powder, Pavlov (1927) observed that after a few measurement sessions, the animals started to salivate as soon as he entered the test room, before feeding the animals. This allowed the discovery of a behavioral phenomenon

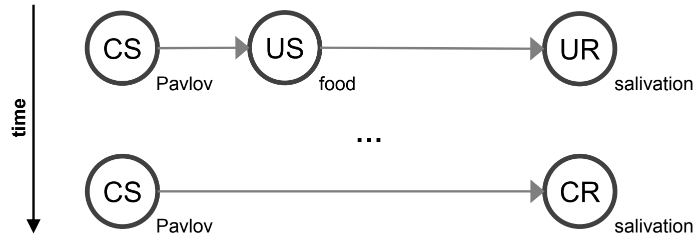


Figure A.1: Example of the *classical conditioning* paradigm in a dog, inspired by Pavlov's experiments.

known as classical (or Pavlovian) conditioning (Anderson, 2000).

Figure A.1 illustrates a typical setting for a classical conditioning experimental procedure. In a first phase, known as *initial pairing* or *training*, an organism's biologically significant *unconditioned stimulus* (US) is paired with a neutral, biologically meaningless stimulus, called the *conditioned stimulus* (CS) (Anderson, 2000; Pavlov, 1927). The US, such as food or an electrical shock, reflexively evokes innate, automatic unconditioned responses (UR), for example, salivating or freezing. The neutral CS can be any event that does not result in an overt behavioral response from the organism under investigation, like the sound of a bell, a light or even a person. In a second phase (*testing*), and after a few pairings between the US and CS, have occurred, the experimenter measures the level of response from the organism when exposed to the CS alone, with no US being presented. The experimenter typically observes a change in response from the organism in the presence of the CS, which now evokes a conditioned response (CR) similar to the UR evoked by the US. Following the example in Figure A.1, the presence of Pavlov alone made the dogs start salivating in anticipation of food delivery. This change in response is due to the development of an *association* between a representation of the CS and one of the US. This is the main idea behind Pavlov's *stimulus substitution theory* (Pavlov, 1927), where the CS "substitutes" the US in evoking the reflexive response.

The evolutionary purposes behind this associative mechanism are the ability of organisms in augmenting the space of contexts where to apply some advantageous response and to anticipate the biological significance of co-occurring events (Cardinal et al., 2002). By determining associations between stimuli in the environment, animals are able to:

1. recognize *contexts* (states) of the environment and thus *anticipate* rewards or punishments and consequences of behavior that are similar to those observed in previous interactions;

2. *integrate information* from previous observations with new, never before experienced stimuli.

Our learning approach follows these ideas from classical conditioning and applies them to RL problems by:

1. *spreading* action and reward information (the Q-values) between similar states;

2. *integrating* information in new, unknown states, from the Q-values of previously experienced similar states.

A.3 Background

In this section we introduce the necessary technical background on factored MDPs and learning with spreading functions, all on which we base our approach for the associative metric.

A.3.1 Learning with Spreading

The need for infinite visits to every state-action pair is unpractical in many situations, and several general approaches have been proposed to mitigate this need. We adopt a simple technique proposed in (Ribeiro and Szepesvári, 1996), where Q-learning is combined with a *spreading function* that "spreads" the estimates of the Q-function in a given state to neighboring states. Formally, given a similarity function $\sigma_t(x, y)$ that measures how close two states x and y are, the Q-learning with spreading update is given by

$$\hat{Q}(x, a_t) \leftarrow (1 - \alpha_t)\hat{Q}(x, a_t) + \alpha_t \sigma_t(x, x_t)\big(r_t + \gamma \max_b \hat{Q}(x_{t+1}, b)\big). \tag{A.1}$$

As discussed in (Ribeiro and Szepesvári, 1996), convergence of Q-learning with spreading to the optimal Q-function can be guaranteed as long as the spreading function σ_t converges to the Kronecker delta-function at a suitable rate.[2]

A.3.2 Learning in Factored MDPs

In many MDPs the state can be described by a finite set of state-variables known as *factors*. In such cases, the MDP is referred to as a *factored MDP* where we can denote by $\mathcal{X}$ a finite set of states that can be factored as $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_n$. Each factor $\mathcal{X}_i$ can take on values $x_i \in \mathcal{D}(\mathcal{X}_i)$ where $\mathcal{D}$ is a finite domain. Let us further refer to an element $x = (x_1, \ldots, x_n) \in \mathcal{X}$ as a *state* and to an element $x_i \in \mathcal{X}_i$ as a *stimulus*.

In the context of our study we consider stimuli in domains of categorical nominal data, *i.e.*, variables that describe discrete values with no ordering between them. One aspect about factored MDPs that we explore is that often not all the factors contribute to the state or the reward in the next time step (Kroon and Whiteson, 2009). In such cases it is possible to leverage the structure of the factored MDP to improve the efficiency of RL methods (Guestrin et al., 2002; Kearns and Koller, 1999). This is particularly important if many of the state-variables are irrelevant for the task that the agent must learn, and it is possible to improve the learning performance by identify-

[2] Actually, the algorithm described in (Ribeiro and Szepesvári, 1996) also considers spreading across actions. In this study we consider only spreading across states, as this is sufficient for our purposes.

ing such irrelevant state-variables and allowing the learning agent to focus only on those that are relevant (Jong and Stone, 2005; Kroon and Whiteson, 2009).

A.4 Associative Metric for RL

In this section we introduce a new associative metric to be used in factored MDPs. For that we combine the classical conditioning paradigm and the RL framework introduced in the previous section, as a mean to improve the performance of a learning agent.

To better explain our learning procedure let us consider a behavior phenomenon associated with the classical conditioning paradigm known as *secondary conditioning* or *sensory preconditioning* as an example to follow throughout this section. These phenomena take place when a CS (CS1) that is trained to predict some US is paired with a different CS (CS2), either before or after CS1 and US are paired. By means of this secondary association, CS2 also becomes associated with the US value through its association with CS1, and ends up evoking the same kind of CR (Balkenius and Morén, 1998). Figure A.2 illustrates an example of the secondary conditioning phenomenon, where, for explanatory purposes, we consider that the stimuli come from different perceptual modalities.

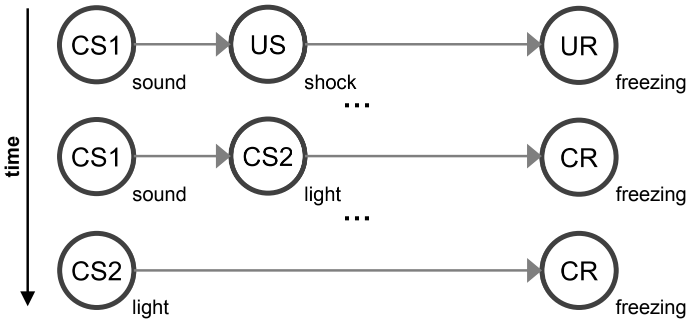


Figure A.2: Example of a *secondary conditioning* phenomenon.

Biologically speaking, the objective is that the agent, after being trained with sound-shock pairings followed by sound-light pairings, is able to predict the presence of the shock whenever it perceives the light. In a more computational perspective, the objective of the learning procedure is to discover that the environmental states involving the light and the shock are somehow “similar”. In this manner, both states should have similar “value”, and executing similar actions in those states should lead to similar outcomes.

We can therefore decompose the agent’s problem into two sub-problems: one of *identifying similar states* and another of *using the information* gathered in some states in states that are similar. Sections A.4.1 and A.4.2 explain our approach in solving the first subproblem, where we propose the combination of a sensory pattern tree and a new associative metric to measure the distance between similar states. In Section A.4.2 we also explain how this metric can then be combined with Q-learning with spreading in factored MDPs to improve the performance of a

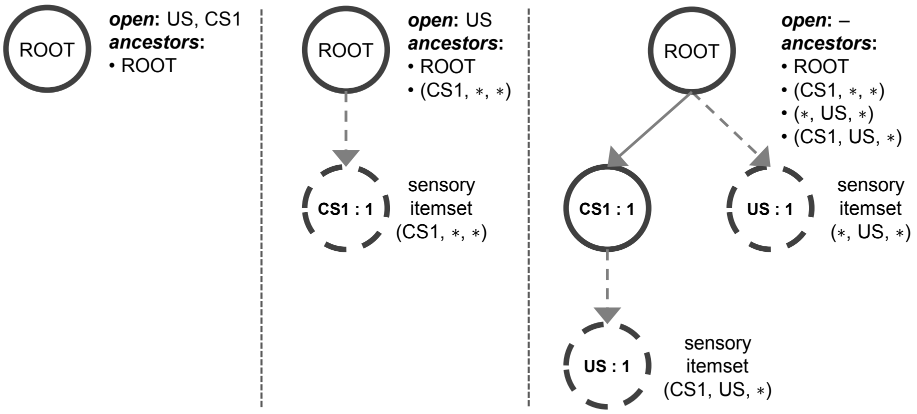


Figure A.3: The steps involving the construction of an associative sensory tree after an initial observation of state $x_1 = (\mathsf{CS1}, \mathsf{US}, \emptyset)$, where the sensory itemsets associated with each new node are explicitly indicated. The updated and inserted nodes at each step are marked with a dashed line. See text for detailed explanation.

learning agent.

A.4.1 Sensory Pattern Mining

As we have seen in Section A.2, one of the fundamental aspects in the classical conditioning paradigm is the ability of individuals to establish associations between the stimuli they perceive. Stimuli that are frequently perceived together are more likely to lead to similar *value* and *outcome* than stimuli that seldom co-occur. Departing from this idea, our learning agents can later use such associations to determine how "similar" two states are. To determine such associations we follow the method from (Sequeira and Antunes, 2010), where a *sensory pattern mining technique* identifies associations between stimuli occurring in the agent's perceptions online, while the agent interacts with its environment. This method identifies such associations by incrementally constructing an *associative sensory tree*, using a variation of the *FP-growth* algorithm (Han et al., 2004) for transactional pattern mining.

Building the Associative Tree

Let us denote any possible subcombination of stimuli $\mathbf{s} = (x_{i_1}, \ldots, x_{i_k})$ with $k <= n$ as a *sensory itemset*. We assume without loss of generality that the sets $\mathcal{X}_i, i = 1 \ldots, n$, are ordered sets.[3]

Following the example provided in the beginning of this section, let us consider states in the form $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3$, where $\mathcal{X}_1 = \{\mathsf{CS1}, \emptyset\}$, $\mathcal{X}_2 = \{\mathsf{US}, \emptyset\}$ and $\mathcal{X}_3 = \{\mathsf{CS2}, \emptyset\}$ are binary variables and symbol $\emptyset$ represents the absence of a stimulus in the observed state. Figures A.3 and A.4 shows the steps involving the construction of the tree when the agent perceives the state $x_1 = (\mathsf{CS1}, \mathsf{US}, \emptyset)$ (sound-shock pairing) from the environment followed by state $x_2 = (\mathsf{CS1}, \emptyset, \mathsf{CS2})$

[3] We note that the specific order of the elements $\mathcal{X}_i \in \mathcal{X}$ is not important, as long as it remains fixed throughout learning. This is a requirement of the tree construction algorithm that guarantees a minimal representation (Sequeira and Antunes, 2010).

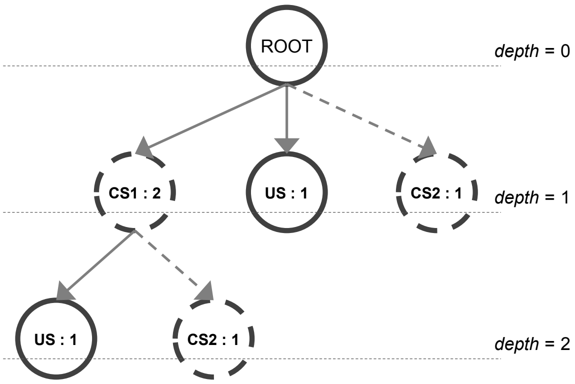


Figure A.4: Updated tree after observing state $x_2 = (\mathsf{CS1}, \emptyset, \mathsf{CS2})$. The depth of each node is explicitly indicated. See text for detailed explanation.

(sound-light pairing). The general sensory pattern mining algorithm in (Sequeira and Antunes, 2010) dynamically builds a sensory tree as follows:

- At every time-step t, the agent observes state $X(t) = (x_1(t), \ldots, x_n(t))$;

- For each state $X(t)$, the algorithm updates the tree by keeping at each step two lists: an "open" list, initially containing all elements $x_i(t)$ to be inserted into the tree; an "ancestor" list containing the nodes in the tree updated so far for the current state insertion, which initially only contains the ROOT node (see Figure A.3);

- The algorithm then picks one element $x_i(t)$ from the "open" list at a time, ignoring absent elements ($\emptyset$). For each element, a child node $\mathbf{s}$ is created for each node in the "ancestor" list, with counter $n(\mathbf{s}) = 1$. If the child node already exists, its counter is incremented by 1 (see Figure A.4). Each new node in the tree represents a sensory itemset, *i.e.*, a subcombination of stimuli within $X(t)$. Therefore, the nodes' counters represent the number of times the corresponding itemset was observed by the agent so far.

Measuring the Degree of Association

At each time, one can measure the degree of association between the stimuli in some sensory itemset $\mathbf{s}$ by using the *Jaccard index* (Jaccard, 1912). Given the sensory itemset $\mathbf{s} = (x_{i_1}, \ldots, x_{i_k})$, let $\mathrm{d}(\mathbf{s})$ and $n(\mathbf{s})$ denote, respectively, the depth of and the counter associated with the corresponding node in the tree. For nodes not directly below the root node, the Jaccard index associated with $\mathbf{s}$ is given by

$$J(\mathbf{s}) = \frac{n(\mathbf{s})}{\sum_{\mathbf{s}_d} (-1)^{\mathrm{d}(\mathbf{s}_d)+1} n(\mathbf{s}_d)}, \tag{A.2}$$

where the summation is taken over all nodes $\mathbf{s}_d$ in the *dependency tree* of $\mathbf{s}$, *i.e.*, the sensory subtree containing all nodes in the "ancestor" list obtained after introducing itemset $\mathbf{s}$ in the tree.

Following our example, we can now calculate the Jaccard index of state x_1 by solving (A.2):

$$J(\mathbf{s}) = \frac{n(\mathsf{CS1}, \mathsf{US}, *)}{n(\mathsf{CS1}, *, *) + n(*, \mathsf{US}, *) - n(\mathsf{CS1}, \mathsf{US}, *)} = \frac{1}{2}$$

As expected, the index is inferior to 1, as stimulus CS1 also appears in x_2, where the US is absent.

A.4.2 Associative Metric for Factored MDPs

To define a metric using the sensory tree described in the previous section, we introduce some additional notation that facilitates the presentation. For any state $x \in \mathcal{X}$, let $\mathcal{S}(x)$ denote the set of all sensory itemsets associated with x which, as we have seen, represents the set of all subcombinations present in the dependency tree of x.

We are now in position to introduce our state-space metric. Given the sensory tree at time-step t we consider the distance between two states x and y as

$$d_A(x, y) = 1 - \frac{\sum_{\mathbf{s} \in \mathcal{S}(x) \cap \mathcal{S}(y)} J(\mathbf{s})}{\sum_{\mathbf{s} \in \mathcal{S}(x) \cup \mathcal{S}(y)} J(\mathbf{s})}. \tag{A.3}$$

Mapping d_A is indeed a proper *metric*, as it can be reduced to the *Tanimoto distance* (Lipkus, 1999) between two vectors associated with x and y, each containing the Jaccard indices for the sensory patterns associated with x and y, respectively.

Having defined the associative metric we can tackle the first problem defined in the beginning of the section and determine whether two states are similar or not. We can define $\mathcal{S}(x_1) = \{(\mathsf{CS1}, \mathsf{US}, *), (\mathsf{CS1}, *, *), (*, \mathsf{US}, *)\}$ and $\mathcal{S}(x_2) = \{(\mathsf{CS1}, *, \mathsf{CS2}), (\mathsf{CS1}, *, *), (*, *, \mathsf{CS2})\}$. The distance between x_1 and x_2 can then be calculated from (A.3) as

$$d_A(x_1, x_2) = 1 - \frac{1}{0.5 + 0.5 + 1 + 0.5 + 0.5} = \frac{2}{3}$$

This means that the degree of similarity between the two states is $1/3$. It follows that our proposed model supports the secondary conditioning phenomena described above: the light and foot shock stimuli have some degree of association by means of the sound stimulus, although CS2 and US were never observed together by the agent.

Now that we are able to identify similar states we describe how the metric in (A.3) can be combined online with Q-learning with spreading. In the experiments reported in this chapter, we use two different spreading functions:

- for the computational experiments in Section A.5.1, we use $\sigma_{1,t}(x, y) = e^{-\eta_t d_A(x,y)^2}$;
- for the biological experiments in Section A.5.2, we use $\sigma_{2,t}(x, y) = (1 - d_A(x, y))^{\eta_t(y)}$.

In both cases, η_t is a slowly increasing value that ensures that σ_t approaches the Kronecker delta function at a suitable rate, and d_A is the metric defined in (A.3). As seen in Section A.3.1, at each time step t the spreading functions $\sigma_{i,t}$ use information from the current state $X(t)$ to update all other states $y \in \mathcal{X}$, depending on the similarity between $X(t)$ and y calculated according to the structure of the sensory tree at t.

A.5 Experimental Results

In this section we describe several experiments carried out in order to validate our method from both computational and biological perspectives. The main results stemming from the experiments are analyzed and discussed.

A.5.1 Computational Experiments

In this set of computational experiments we show the potential of combining the proposed associative metric with spreading in Q-learning, providing a boost in the agent's performance in several factored MDP problems.

Methodology

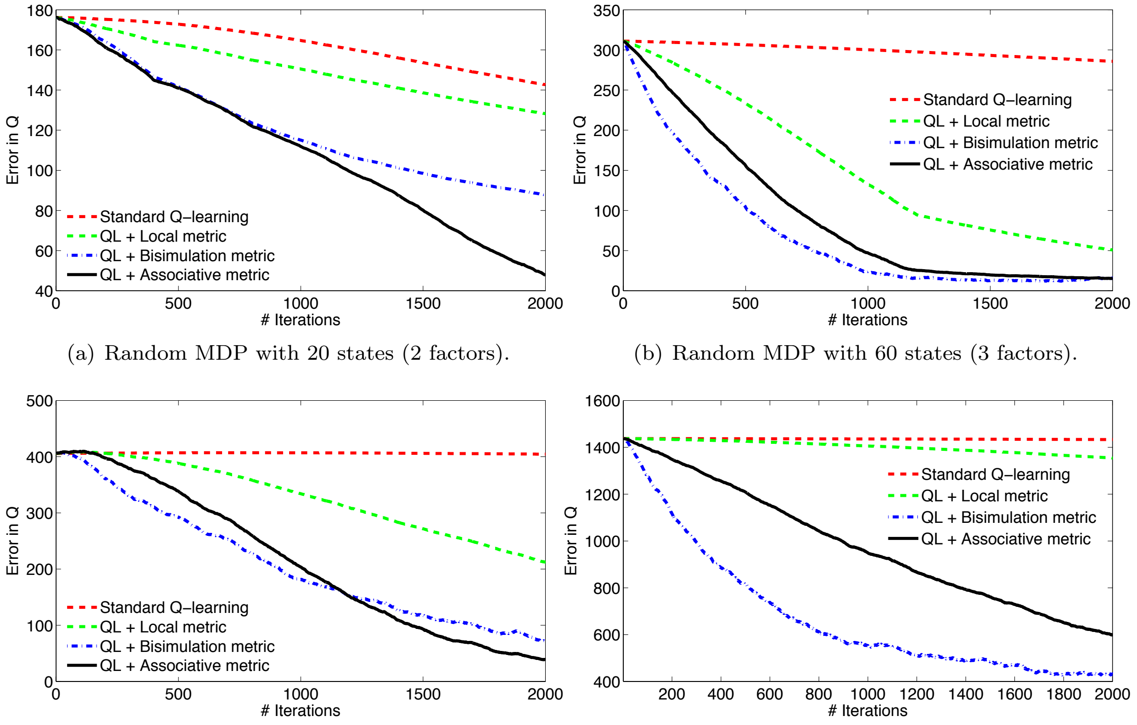
To better assess the applicability of our method, we applied Q-learning with spreading using $\sigma 1_t$ defined earlier and our associative metric in several factored environments, with a state-space that could be factored into between 1 and 4 factors, with a number of states between 20 and 481, and 5 actions. The transition probabilities between states and the reward function were generated randomly. We present the results obtained in 4 of those environments having, respectively, 20 states (5×4), 60 states ($5 \times 4 \times 3$), 120 states ($5 \times 4 \times 3 \times 2$) and 481 states ($9 \times 7 \times 7$, where the dimension and number of factors was chosen randomly). In all scenarios we use $\gamma = 0.95$ and uniform exploration.

We compare the performance of standard Q-learning (with no spreading) with that of Q-learning with spreading using several metrics. In particular, we compare 3 metrics:

- A *local metric*, d_ℓ, computed from the transition probabilities of the MDP. Given two states $x, y \in \mathcal{X}$, $d_\ell(x, y)$ corresponds to the average number of steps necessary to transition between the two states. The distance between states that do not communicate was set to an arbitrary large constant.

- A simplified *bisimulation metric*, d_b (Ferns et al., 2004). The distance d_b is a simplified version of the bisimulation metric that relies on the total variation norm proposed in (Ferns et al., 2004, Section 4.2).[4] We note that this is a theoretically sound metric that, however, requires complete knowledge of both P and r.

[4] To simplify the computation of this metric, we treat each state as an equivalence class. We refer to (Ferns et al., 2004).



(a) Random MDP with 20 states (2 factors).

(b) Random MDP with 60 states (3 factors).

(c) Random MDP with 120 states (4 factors).

(d) Random MDP with 481 states (3 factors).

Figure A.5: Performance of Q-learning with spreading in different factored scenarios, using different metrics with varying knowledge of the MDP parameters.

- The associative metric d_A described in Section A.4.2.

For each of the test scenarios, we ran 10 independent Monte-Carlo trials, and evaluated the learning performance of the different methods by comparing the speed of convergence to the optimal Q-function. The parameter η_t was optimized empirically for each metric in each environment so as to optimize the performance of the corresponding method.

Results

The average results are depicted in Figure A.5. We note, first of all, that our method always outperforms both standard Q-learning and the local metric. The fact that our method learns faster than standard Q-learning indicates that, in these scenarios, the associations between stimuli provide a meaningful way to generalize the Q-values across states. It is also not surprising that our method generally outperforms the local metric, since it implicit assumes that there is some “spacial” regularity that can be used to generalize Q-values across neighboring states. However, this is generally not the case, meaning that in some scenarios the local metric does not provide a significant improvement in performance—see, for example, Figures A.5(a) and (d).

The bisimulation metric, although simplified from (Ferns et al., 2004), is a metric that takes into consideration both the transition structure and the reward function of the MDP. As such, it is not surprising that it allows for good generalization. The fact that our metric performs close to the bisimulation metric in several scenarios—see, for example, Figures A.5(a), (b) and (c)—is, on the other hand, a significant result, since our metric is *learned online*, while the agent interacts with the environment and so uses no prior knowledge on the MDP.

Finally, we note that our metric relies on the factorization of the state-space to build the sensory tree, since the latter is built by associating state-variables that co-occur frequently. In a non-factored MDP, our method would essentially reduce to standard Q-learning. The reliance of our metric on the factorization of the state-space justifies, to some extent, the result in Figure A.5(d). In fact, this corresponds to a large MDP where the "factors" of the state-space are also large. Therefore, not only is the problem larger and, thus, harder to learn, but also our method is able to generalize less than in other more factored scenarios.

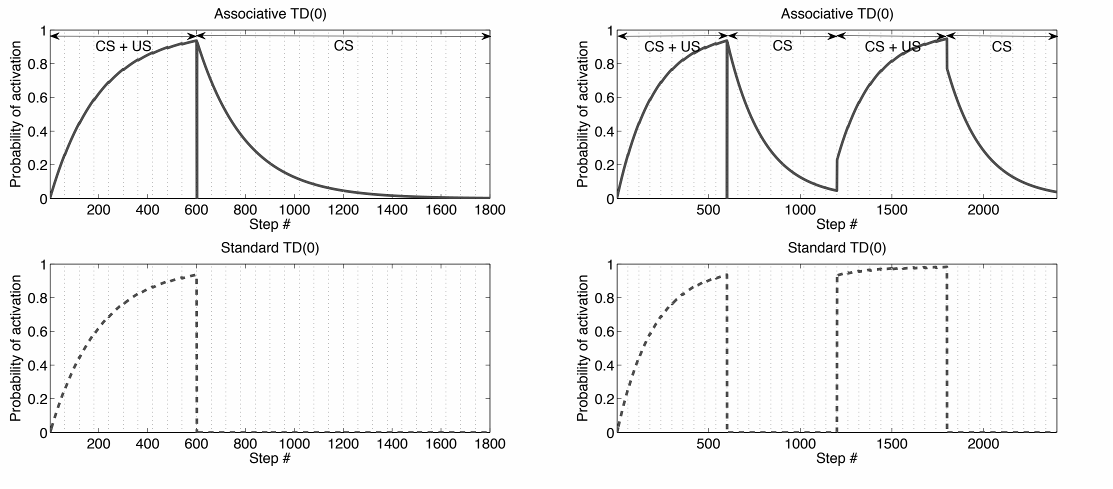
A.5.2 Biological Considerations

From a more biological perspective, it is important to assess whether the proposed method replicates common learning phenomena observed in nature as a consequence of conditioning. As we have seen in Section A.2, these phenomena are the result of the dynamic learning process that takes place in classical conditioning and allows animals to change their behavior according to changes observed in the environment. In order to evaluate the capacity of our model to emulate such phenomena we designed a set of experiments inspired in real-world experimental procedures.

Methodology

In our experiments, the state of the agent corresponds to the stimuli generated according to the schedule of the experiment, possibly including the US, the CS, and other stimuli relevant for that particular experiment. Whenever US is presented, the agent receives a reward of $+1$, and 0 otherwise. Our agent learns the value of each state using TD(0) algorithm (Sutton and Barto, 1987), which is equivalent to using Q-learning and setting $\gamma = 0$, since we are concerned with immediate values, and having only one action available, since we are interested only in the value of the states. We then combine TD(0) with our associative metric as described in Section A.3.1.

Because we give positive reward whenever stimulus US is presented, the value learned for each state can be interpreted as the activation probability of the CR. We use spreading function $\sigma 2_t$ defined in Section A.4.2. Both during training and testing, the presentation of stimuli to the agent took place in cycles of 60 steps. The particular stimuli presented in each phase of the different experiments are detailed below. The presentation of the conditioned/unconditioned stimuli is intertwined with intervals of 70 to 100 time-steps during which either no stimuli or (depending on the experiment) only contextual stimuli/clues are present.



(a) Acquisition/Extinction.

(b) Reacquisition.

Figure A.6: Evolution of the probability of CR activation for the *Acquisition*, *Extinction* and *Reacquisition* experiments for both the conditioned (Associative) and standard TD(0) agent.

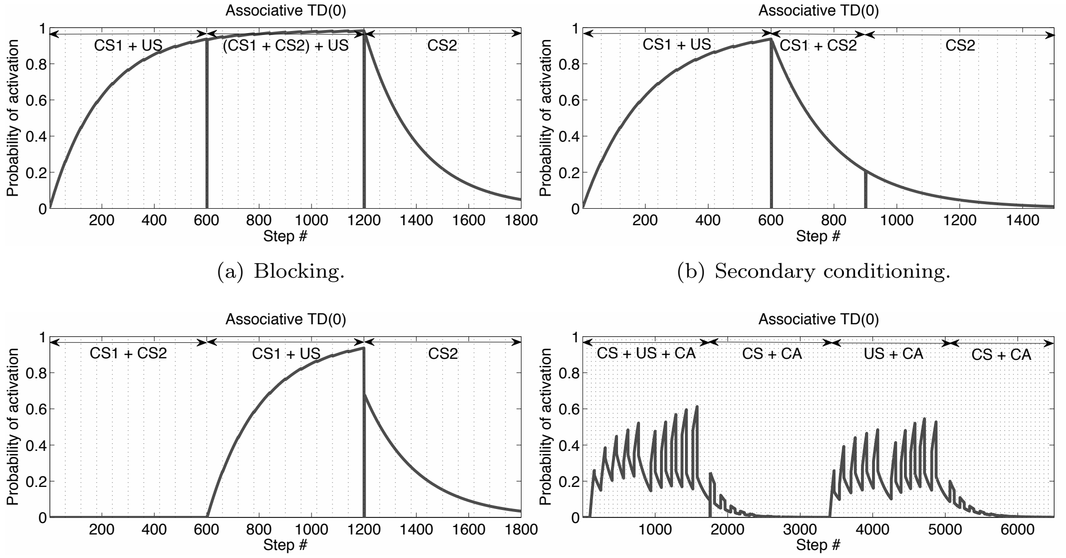
Results

Acquisition and **extinction** are the most basic and fundamental phenomena within the classical conditioning paradigm. Acquisition represents the ability of a CS to evoke a CR after some number of CS+US pairings had occurred. Extinction occurs after acquisition and is defined as the loss of the ability of the CS to elicit the CR (Anderson, 2000; Balkenius and Morén, 1998). As we can see from Figure A.6(a) our model correctly reproduces both phenomena, showing a high probability of activation when only the CS is presented after the CS+US pairing phase. This probability eventually decreases to zero, corresponding to the extinction phenomena since CR is not observed. For comparison, we also present the results obtained with standard TD(0). As expected, the state "only CS" is treated as a completely new state and acquires no value from the previous observations of US.

Spontaneous Recovery is a phenomenon which occurs some time (for example a day) after a series of extinction trials, when the body shows some recovery of the CR when exposed to the CS, meaning that its value (and the association with the US) was not forgotten (Anderson, 2000).

Reacquisition occurs when a second acquisition phase is presented to the individual after extinction has taken place (Pavlov, 1927). The number of CS+US pairing trials necessary for the CS to evoke again the CR decreases with the number reacquisition phases occurred (Balkenius and Morén, 1998; Bouton, 2002). Our model also reproduces this effect, as can be seen from the CR activation probability during the second CS presentation phase in Figure A.6(b). As desired, the starting CR activation level is higher in the second CS+US acquisition phase than it was in the first one. For reference, we again show the performance of the standard TD(0) algorithm.

Blocking is the ability of a previously trained CS (CS1) to block another CS (CS2) from



(a) Blocking.

(b) Secondary conditioning.

(c) Sensory preconditioning.

(d) Reinstatement.

Figure A.7: Evolution of the probability of CR activation for the *Blocking*, *Secondary conditioning*, *Sensory preconditioning* and *Reinstatement* experiments for the conditioned (Associative) TD(0) agent.

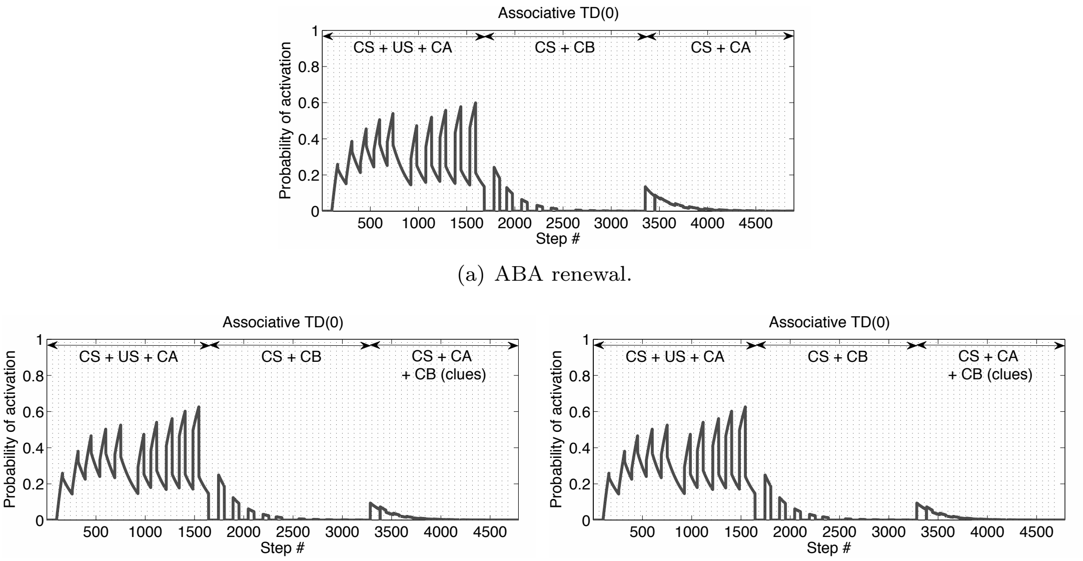
predicting the same US. Because CS1 is already associated with the US, CS2 is unable to form an association and predict its value when paired together with CS1 and US. Therefore no CR will occur in the presence of CS2 alone (Balkenius and Morén, 1998). As can be seen from Figure A.7(a), our model does not reproduce this effect. This is due to the fact that the reward given to the agent when the US is presented is independent of the presence of other stimuli. Because of that, CS2 also benefits from the value of US and thus reproduces CR activation.

Secondary conditioning and **sensory preconditioning**, are two behavioral phenomena presented in Section A.4. The proposed metric allows the reproduction of these kinds of phenomena. To better illustrate this result in a more experimental procedure, Figures A.7(b) and (c) show that our model reproduces both conditioning phenomena through the CR activation curve when only CS2 is presented.

The next set of experiments is focused in conditioning phenomena that occur as a consequence of the context where the procedures take place. Due to secondary conditioning and relapse mechanisms, some animals come to evoke CRs in the presence of contextual or background clues from where acquisition took place (Bouton, 2002). Such clues can be generated from different perceptual modalities like sounds, colors, etc.[5]

Reinstatement is a phenomenon that causes the CS to evoke a CR after being extinguished, by presenting the US alone a certain number of times after extinction (Bouton, 2002). This occurs because contextual clues are being provided along with the US whenever acquisition is taking place. This way, when the US is presented alone, the context gains the power of producing a CR, even

[5] These results are based on the experiments reported in (Bouton, 2002).



(a) ABA renewal.

(b) ABA renewal reduction (A).

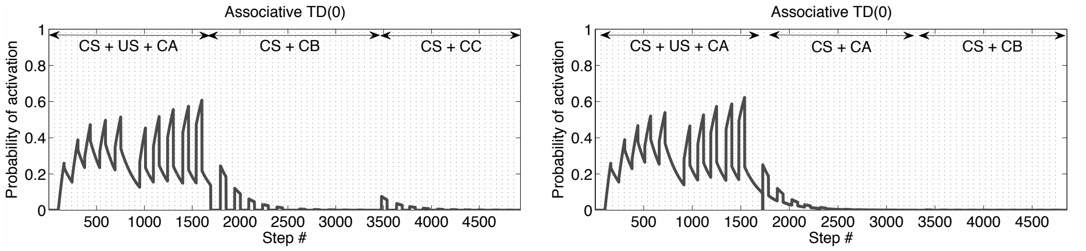
(c) ABA renewal reduction (B).

Figure A.8: Evolution of the probability of CR activation for the *ABA renewal*, *ABA renewal reduction (A)* and *ABA renewal reduction (B)* experiments.

when CS is again presented. Thus, CR is being produced via context and not via CS (Bouton, 2002). As we can see from Figure A.7(d), our model reproduces this phenomenon by showing CR activation in the presence of CS after the US is presented alone (during the experiment, contextual stimulus CA is always presented). We note that, unlike with the previous experiments, we include in these graphs the intervals between the presentation cycles, where only a contextual stimulus (*e.g.*, CA) is presented. This is important because extinction between cycles is already occurring, as we can see from the descending probability curves between the presentation peaks.

ABA Renewal is a conditioning procedure in which CS+US acquisition occurs in a certain context (CA), extinction occurs in a different context (CB), and testing occurs again in the initial context (Bouton, 2002). One could expect that because CS was previously extinguished, it should no longer produce a CR when presented again in the first context. However this is not the case, as the contextual elements where extinction took place are different from the ones where acquisition took place. Therefore, CA "renews" the value of CS and still evokes a CR, as the animal is able to discriminate between the places where the US was present or absent (Bouton, 2002). From Figure A.8(a) it is clear that our model supports such phenomenon. Although the extinction in CB made the CS value to decrease, it did not extinguish it completely, showing a CR activation curve when the CS occurs again in CA.

ABA Renewal Reduction (A): renewal effects can be reduced in several ways. One of them is to include during the test phase occurring in CA contextual elements from CB, the context where extinction took place (Bouton, 2002). Figure A.8(b) shows that our model produces this effect, showing a slightly lower activation level when CS is presented in CA without contextual clues from CB.



(a) ABC Renewal.

(b) AAB renewal.

Figure A.9: Evolution of the probability of CR activation for the *ABC Renewal* and *AAB renewal* experiments.

ABA Renewal Reduction (B): another way of reducing the renewal effect in CA is to present the US in CB where the extinction took place, after extinction. This way, the animal somehow shifts the attribution of US occurrence to context CB, while diminishing the CR activation in CA. Figure A.8(c)

ABC Renewal is another renewal phenomena where testing occurs at a different context (CC than those where acquisition (CA) and extinction (CB) took place. In such case, one can still observe CR responding to a completely new context, although such responding is lower than in the ABA renewal case (Bouton, 2002). Again this is due to the fact that the animal associates extinction to context CB, and because of that, CS still evokes CR activation when presented alone in CC. Figure A.9(a) shows that our conditioned agent produces such effect, showing a slight CR activation level when in context CC. Also as desired, this degree of responding is lower than in the ABA renewal case (Figure A.8(a)).

AAB Renewal occurs when testing occurs in a different context (CB) from where both acquisition and extinction occurred (CA). This procedure reveals that the CS, although extinguished in CA is still capable of evoking a CR in a different context (although more difficult to detected than ABA or ABC renewal cases), which is another proof that unlearning did not occur (Bouton, 2002). To demonstrate this effect in our model we had to restrict the range of CR activation to 0.01, as depicted in Figure A.9(b). As desired, our agent showed a small degree of responding when in context CB, showing that the ability of CS in evoking a CR was not completely forgotten during extinction, although such degree of activation would be despised in real settings.

A.6 Discussion

We conclude this section with some relevant observations concerning our proposed method. First of all, the size of the associative sensory tree exhibits a worst-case exponential dependence in the number of *state-factors* (not states). However, aside from the memory requirements associated therewith, the structure of the tree is such that the computation of the distance is linear in the number of factors, which is extremely convenient for the online processing of distances. Moreover, the adopted tree representation can safely be replaced by other equivalent representations, such as the FP-tree (Han et al., 2004) that, while more efficient in terms of memory requirements, render the computation of the distance computationally more expensive, as they typically imply explicitly searching the (much smaller) tree.

Secondly, we note that the maximal size of the tree is only achieved when *all* the state space has been explored. However, it is in the early stages of the learning process—when little of the state space has been explored—that the use of associative metric may be more beneficial. In other words, when the size of the tree approaches its maximum size, the contribution of the associative metric to learning is generally small. Therefore, limiting the tree size to some pre-specified maximum or using tree-pruning techniques as those discussed in (Sequeira and Antunes, 2010) should have little impact on the performance of our proposed method.

Thirdly, and independently of the representation used, our approach does not require the learning agent to know the set $\mathcal{X}$ of possible states in advance. In fact, the agent can build an estimate set $\hat{\mathcal{X}}$ that contains all states visited so far. Every time the agent enters a new state x, it can "add" it to its current estimate $\hat{\mathcal{X}}$ and modify $\hat{Q}$ to accommodate for the new state. The Q-values associated with the new state can be initialized using information from similar (known) states, generalizing the information from these states to the new state. Therefore, our model can to reproduce classical conditioning behaviors observed in nature and already discussed in Section A.2:

1. *anticipate* consequences of behavior from previous interactions in similar states;

2. *integrate* information from previous observations with new, never before experienced states.

This ability to generalize experiences to states never visited before leads not only to an improved learning performance, as illustrated in Section A.5.1, but also to the appearance of several phenomena that can be matched to those observed in typical conditioning experiences in nature, as discussed in Section A.5.2. Furthermore, we note that the sensory tree update algorithm explained in Section A.4.1 runs online and interleaved with the agent's learning in the MDP. This means that as time goes by and the tree structure changes, so the distance between the several observed states will change by means of the associative metric. As a consequence, the spreading makes the agent's model of the environment to dynamically change according to its experience with it.

A.6.1 Future Work

As we have seen, the learning mechanism proposed in this chapter builds an association tree of state-variables. It is then expected that, in more realistic scenarios involving a larger state space with more state variables considered, the proposed data structure may require a large amount of data in memory to contain the associations. Tests will be carried out to measure the efficiency of the proposed state-distance metric when operating in such conditions. A validation procedure could be performed that compares such computational demand with the degree of generalization provided by using such method.

Also, a future study can compare the proposed metric with other generalization methods to test the robustness of the solution being proposed. In a more biological perspective, it is be important to compare the proposed conditioning model with other models developed for the same effect, for example by performing more profound tests as the ones performed in (Balkenius and Morén, 1998).

A.7 Summary

In this chapter we deviated from previous approaches within this thesis relating reward design for IMRL and proposed a new state-space associative metric for factored MDPs that draws inspiration from classical conditioning in nature. Our metric relies on identified associations between state-variables perceived by the learning agent during its interaction with the environment. These associations are learned using a sensory pattern-mining algorithm and determine the similarity between states, thus providing a state-space metric that requires no prior knowledge on the structure of the underlying decision problem. The sensory pattern-mining algorithm relies on the *associative sensory tree*, that captures the frequency of co-occurrence of stimuli in the agent's environment.

APPENDIX B

Other Experiments

In this chapter we include the results and discussion on other experiments that were performed in the preparation of this thesis. They aid to support and further illustrate the outcome achieved during the experiments described in the main document.

B.1 Universal Parameter in the Pac-Man Scenarios

In this experiment we assess the universality of the parameter vectors optimized in the foraging experiments of Chapter 4 by using such parameterization in the Pac-Man scenarios described in Section 5.3.2. The objective is to test our hypothesis for the non-universality of particular parameter vectors, especially in cases of scenarios which task demands very different strategies from the agents.

B.1.1 Methodology

We test the universal agent discovered in the experiments in Section 4.3.6 in the 4 Pac-Man scenarios. The universal agent thus uses a parameter vector $\boldsymbol{\theta}^U = [0.0, 0.0, -0.3, 0.0, 0.7]^\top$ in conjunction with the emotion-based reward features defined in Chapter 4. The methodology is the same as the foraging experiments described in Section 4.3.2, *i.e.*, the agent uses prioritized sweeping with parameters $\alpha = 0.3$, $\gamma = 0.9$, $\lambda_\epsilon = 0.9999$, $\lambda_\mathsf{n} = 1.001$, a backup limit of 10 state-action pairs and a minimum priority threshold of 10^{-4}. We generate a total of $N = 200$ histories as independent Monte-Carlo trials for the reward function $r(\boldsymbol{\theta}^U)$ where we simulate the agent for $T = 100,000$ learning steps.

B.1.2 Results and Discussion

Table B.1 compares the performance of the universal agent, a fitness-based agent and the optimal agent for each Pac-Man scenario using the emerged reward features in the experiment in Section 5.2. As we can see, the universal parameter vector, that showed to be "good enough" for the foraging scenarios, performs very poorly when put in scenarios which dynamics are very different from those environments where it was optimized. In the case of the Pac-Man scenarios, the universal agent performed much worse than the agents being guided solely by the fitness-based, external reward. This means that in these scenarios, the specific "emotional motivation" provided by the configuration of the parameter vector $\boldsymbol{\theta}^U$ was in fact detrimental for the agent's fitness attainment. In a sense, this configuration made the agent to incorrectly *appraise* its environment and weight the different aspects of its environment in a maladaptive fashion, ultimately leading it to very poor performances.

This result thus reinforces our claim that there are no universal parameter vectors that perform well—even when compared to the fitness-based agent—in all kinds of scenarios with all kinds of

Table B.1: Comparison of the performance of the universal emotion-based agent and the optimal and fitness-based agents in each Pac-Man scenario. The results correspond to averages over 200 independent Monte-Carlo trials.

Scenario	**Mean Fitness**		
	Universal	Optimal	Fitness-based
Power-Pellet Scenario	$-4,303.8 \pm 135.8$	$1,265.0 \pm 424.9$	$-1,902.6 \pm 183.5$
Eat-all-Pellets Scenario	-591.5 ± 44.1	$1,005.5 \pm 207.1$	25.3 ± 215.5
Rewarding-Pellets Scenario	$1,118.6 \pm 102.1$	$4,343.7 \pm 210.1$	$3,060.8 \pm 208.6$
Pac-Man Scenario	126.7 ± 37.4	$1,223.6 \pm 117.5$	862.2 ± 95.7

different challenges presented by them. For each scenario we have to optimize the specific configuration of the parameter vector, especially in cases where the task demands different strategies to be solved.

B.2 Multiagent Foraging Experiments

This set of experiments follows those described in Section 6.4 in the context of social intrinsic reward design. We design a series of experiments within a multiagent context of up to 3 learning agents in foraging scenarios similar to those included in the experiments of Section 4.3. We therefore test the impact of having larger environments where the agents can explore the existence of food resources with a lower probability of encountering other agents. In this manner we can test the robustness of the social intrinsic reward features proposed in Chapter 6 in scenarios with such characteristics.

In these experiments we model our agents as predators trying to eat available preys in the environment. We abstract from the fact that, in nature, some species of predators have strict social hierarchies or that some may not even engage in socially-aware behaviors. We rather simulate the fact that there are resources within the environment, available for any agent to consume them, and that agents have the choice to eat them or not at some point in time.

B.2.1 Methodology

In all the foraging scenarios, each agent k has now available 5 possible actions, $\mathcal{A}^k = \{Up, Down, Left, Right, Eat\}$. The directional actions move the agent deterministically to the adjacent cell in the corresponding direction. We explicitly included an *Eat* action that consumes a food resource if one is present in the agent's current location, and does nothing otherwise.

Like the experiments reported in Section 6.4, at each time step each agent k is able to observe:

- its current (x, y) *position*;
- its *satiation status*, *i.e.*, whether it is hungry, satisfied or full;
- whether *food* is present at its current location;

- whether *another agent* is present at its current location. There are $K-1$ binary observations of this kind, K being the number of agents in the environment;
- how many agents have consumed a food resource since k last ate.

We recall that whenever an agent consumes a food resource, it becomes full for one time step, after which it returns to the satisfied state. Also, an agent becomes hungry if it does not consume any resources for β_{mh} time-steps. In all the multiagent foraging scenarios we set $\beta_{mh} = 30$ and the penalty for being hungry $\beta_{hp} = 0.15$.

We again used prioritized sweeping (Moore and Atkeson, 1993) to learn an approximate (single-agent) model of the environment that it then uses to compute an individual policy, according to the its (perceived) reward function $\hat{r}^k$. We used a learning rate $\alpha = 0.3$, a discount factor $\gamma = 0.9$, a backup limit of 10 state-action pairs and a minimum priority threshold of 10^{-4}. Each agent follows an ε-greedy policy with a decaying exploration rate $\varepsilon_t = \lambda_\epsilon^t$, with $\lambda_\epsilon = 0.99995$. To compute the social group's fitness we used the method described in Section 6.4.2 but running a total of $N = 100$ independent Monte-Carlo trials of $T = 100,000$ time-steps. Also, given the results of the experiments in Chapter 6 we considered only populations of "uniform" groups, *i.e.*, where the parameter vectors are the same for all the agents in the group ($\boldsymbol{\theta}^{*,1} = \boldsymbol{\theta}^{*,2} = \ldots = \boldsymbol{\theta}^{*,K}$).

B.2.2 Multiagent Foraging Scenarios

We follow the notation used in Section 6.4.3 to describe the scenarios but adding the prefix "Foraging", *i.e.*, each scenario is identified by "**A-R-S Foraging Scenario**", where **A** is the number of agents, **R** the number of resources, **S** the number of start-positions in the environment.

2-1-2 Foraging Scenario: The environment for this scenario is depicted in Figure B.1. In this scenario, two agents try to eat a rabbit that is *always* available in the indicated position. Agent 1 departs from the top-left position while Agent 2 departs from the top-right position. Whenever one agent consumes the food resource it is repositioned in top-left, while the other agent is repositioned at top-right. Whenever the two agents try to consume the same resource simultaneously, *neither of them succeeds*. The placement of the agents thus gives an advantage to the last eating agent, as it can reach the food source faster and allow the other to starve. Whenever the two agents try to consume the resource simultaneously, neither of them succeeds. ◊

2-2-2 Foraging Scenario: In this scenario there are always two food resources available in the positions indicated in Figure B.2. As before, Agent 1 departs from top-left while Agent 2 departs from top-right. Also, whenever one agent consumes a resource, it is repositioned in top-left and the other agent is repositioned at top-right. Also trying to eat the same food resource simultaneously results in neither of the agents succeeding.

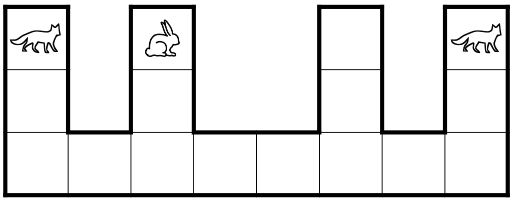


Figure B.1: Environment used in the 2-1-2 Foraging Scenario. See text for details.

However, the fact that there are two food resources available allows both agents to *eat simultaneously*. ◊

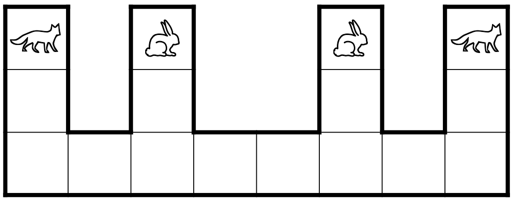


Figure B.2: Environment used in the 2-2-2 Foraging Scenario. See text for details.

2-1-1 Foraging Scenario: In this scenario there is always one rabbit available in the position indicated in Figure B.3. Both agents depart from the top-right position. Whenever one of the agents consumes a food resource they are both repositioned in that location. In this scenario, however, Agent 1 is *stronger* than Agent 2 and whenever the two agents try to consume the resource simultaneously, only Agent 1 succeeds. This gives an advantage to Agent 1, as it can always overpower Agent 2 and allow it to starve. ◊

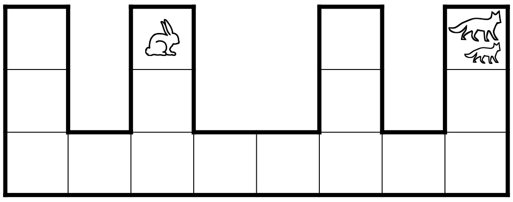


Figure B.3: Environment used in the 2-1-1 Foraging Scenario. See text for details.

3-1-1 Foraging Scenario: In this scenario there are three agents competing for one food resource which is always available at the location specified in Figure B.4. All the agents depart from the top-left position, and whenever one of the agents consumes the food resource *all* are repositioned in the same departing position. The placement of the agents makes that none of them is in advantage because they are all at the same distance to the food resource. Like with the previous scenarios, if two or more agents try to consume the food resource at the same time, none of them succeeds. ◊

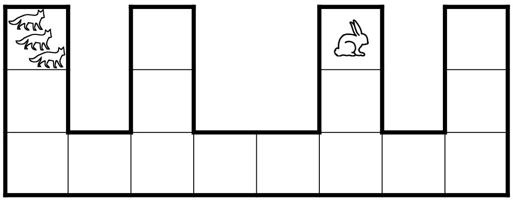


Figure B.4: Environment used in the 3-1-1 Foraging Scenario. See text for details.

3-1-3 Foraging Scenario: Like with the previous scenario, three agents try to eat a food resource always available in the position indicated in Figure B.5. However, similar to what occurred in the 3-1-3 Scenario, in this scenario the agents always depart from different positions. There are 3 possible departing positions as indicated, located at the top-left, top-middle-left and top-right positions. This scenario has a placement policy similar to the 2-1-2 Foraging Scenario as the agent that just ate is placed *closer* to the food resource, *i.e.*, in the top-right position. The remaining agents are placed in the other starting positions according to their number k within the social group, from left to right. For example, imagining that Agent 2 eats, then Agent 1 is placed in top-left and Agent 3 in the top-middle-left position. In this scenario we simulate a kind of *social ranking structure* within the group that implicitly favors agents with the highest numbers. ◊

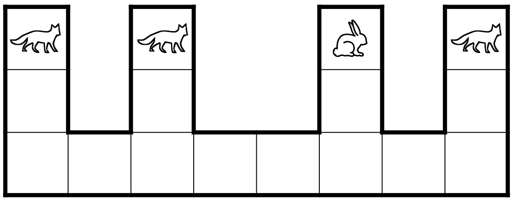


Figure B.5: Environment used in the 3-1-3 Foraging Scenario. See text for details.

3-2-3 Foraging Scenario: In this scenario we used an extended version of the previous environments which is depicted in Figure B.6. The three agents compete for two food resources which are always available in the specified locations. Like in the previous scenario, each agent departs from a different location, positioned at top-left, top-middle or top-right. In this case, the agent that eats at some time goes to the top-middle position, thus starting from an equal distance relative to both food locations, and the other agents are positioned according to their "social rank" as described above. ◊

B.2.3 Results and Discussion

Table B.2 summarizes the results of our experiments by comparing for each scenario the overall fitness obtained by the group of agents using the optimal parameter vector $\boldsymbol{\theta}^*$ against a group of

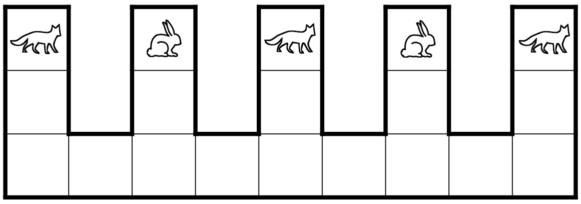


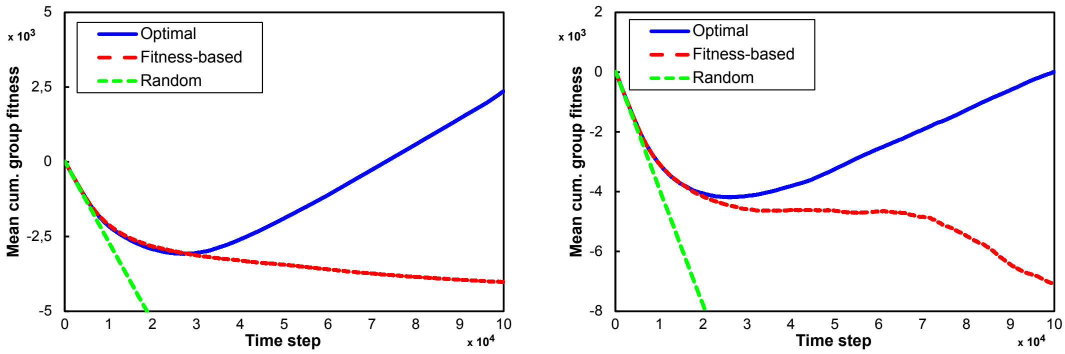
Figure B.6: Environment used in the 3-2-3 Foraging Scenario. See text for details.

Table B.2: Mean cumulative group fitness attained in each scenario. In each scenario, we indicate the optimal parameter vector $\boldsymbol{\theta}^{*,k}$, and the parameter set corresponding to a group of agents that receive only the fitness-based reward.

Scenario		$\theta_{\mathfrak{ext}}$	$\theta_{\mathfrak{int}}$	$\theta_{\mathfrak{fit}}$	**Mean Fitness**
2-1-2 Foraging Scenario	$\boldsymbol{\theta}^{*,k}$	0.50	0.25	0.25	$2,426.3 \pm 104.4$
	Fit.	0.00	0.00	1.00	$-1,377.4 \pm 40.1$
2-2-2 Foraging Scenario	$\boldsymbol{\theta}^{*,k}$	0.75	0.00	0.25	$8,068.6 \pm 90.2$
	Fit.	0.00	0.00	1.00	$8,059.3 \pm 74.2$
2-1-1 Foraging Scenario	$\boldsymbol{\theta}^{*,k}$	0.50	0.25	0.25	$2,512.7 \pm 75.4$
	Fit.	0.00	0.00	1.00	$-2,038.4 \pm 95.3$
3-1-1 Foraging Scenario	$\boldsymbol{\theta}^{*,k}$	0.25	0.50	0.25	431.5 ± 265.0
	Fit.	0.00	0.00	1.00	$-1,643.5 \pm 187.7$
3-1-3 Foraging Scenario	$\boldsymbol{\theta}^{*,k}$	0.50	0.25	0.25	-41.4 ± 182.8
	Fit.	0.00	0.00	1.00	$-2,443.0 \pm 192.3$
3-2-3 Foraging Scenario	$\boldsymbol{\theta}^{*,k}$	0.75	0.00	0.25	$5,213.8 \pm 73.6$
	Fit.	0.00	0.00	1.00	$4,753.2 \pm 125.4$

agents receiving only fitness-based reward. The results correspond to averages of 100 independent Monte-Carlo trials. As can be seen from the results, socially motivated agents attain greater degrees of fitness as a group when compared to the group of agents using only the fitness-based reward during learning. These agents are implicitly "selfish" and, combined with the structure of the environments, this behavior typically leads one of the agents to starvation. To better illustrate these results, Figures B.7-B.9 depict the evolution of the fitness of the social group in each tested scenario.

In general, the results of these experiments follow the ones described in Section 6.4 for the lever environments and show that we do not get much different results by increasing the size of the environments used. In fact, what matters for the emergence of cooperation are the specific "game-like" situations behind the dynamics of each environment. Generally speaking, the results in this chapter also show that in the scenarios where the number of food resources available is less than the number of agents in the social group, the optimal parameter vector $\boldsymbol{\theta}^*$ fairly distributes the importance of the three reward features considered. For example, we can clearly see this effect



(a) 2-1-2 Foraging Scenario

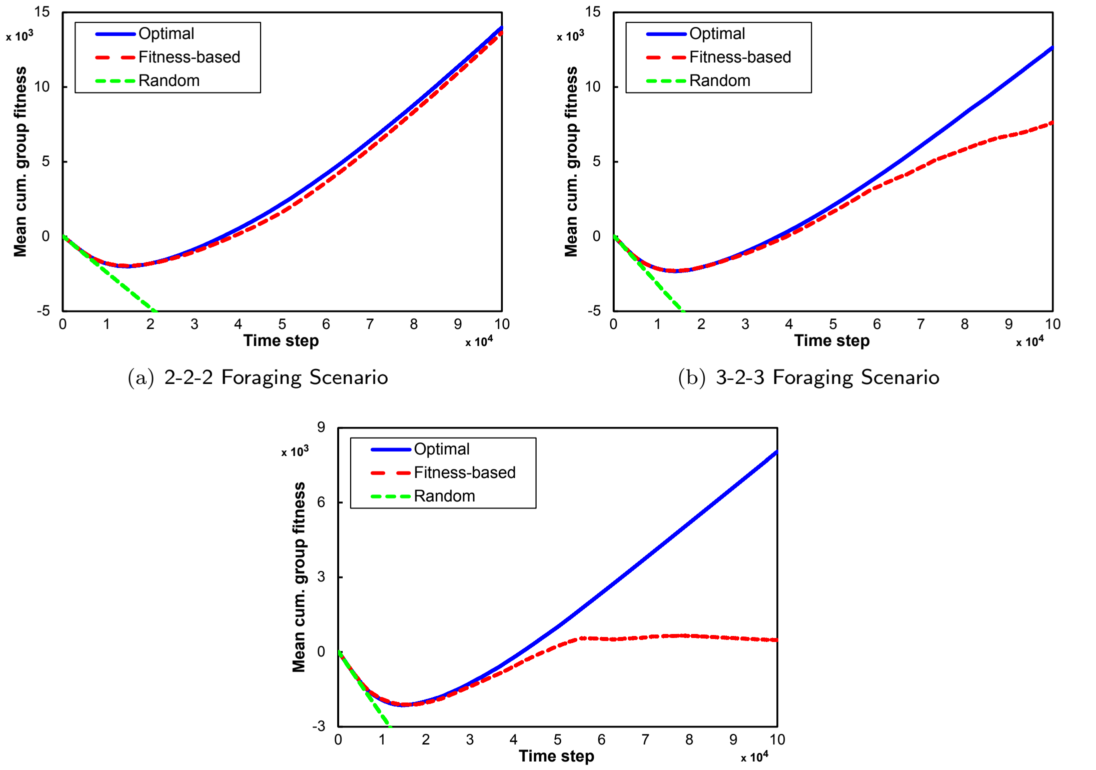
(b) 3-1-1 Foraging Scenario

Figure B.7: Evolution of the fitness of the social group in the 2-1-2 Foraging Scenario and 3-1-1 Foraging Scenario. Results are averages over 100 independent Monte-Carlo trials. "Optimal" corresponds to a group of agents learning with the optimal parameter vector $\boldsymbol{\theta}^*$. "Fitness-based" corresponds to a group of agents receiving only fitness-based reward $r^{\mathcal{F}}$, and "Random" is a group of agents acting randomly, *i.e.*, receiving no reward at all.

in the 2-1-2 Foraging Scenario, where the "socially-aware" agents attain a much greater fitness than the group of "selfish" agents, as illustrated in Figure B.7(a). The usefulness of the social reward signals can also be seen in the 3-1-1 Foraging Scenario, the only difference being the number of agents that compete for the food resource. Because the agents start from the same position, the fitness-based parameter vector $\boldsymbol{\theta}^{\text{fit}}$ makes them compete for the resource without being able to consume it, leading to a generalized starvation as depicted in Figure B.7(b). On the contrary, by examining the agents' behavior in the optimal case we can observe a "socially-aware" strategy in which the agents feed in turns, each waiting for the other two agents to eat before consuming the food themselves.

Having Sufficient Resources

Much like the lever experiments in which the food resources are *sufficient* in relation to the number of agents within the social group, the results of the 2-2-2 Foraging Scenario and the 3-2-3 Foraging Scenario show that "selfish" behaviors can thrive in situations where resource sharing is not needed. However, unlike the 2-2-2-2 Scenario, the optimal parameter vectors $\boldsymbol{\theta}^*$ for these scenarios, presented in Table B.2, completely ignore $\phi^k_{\mathtt{int}}$ while giving more importance to the fitness-based reward provided by eating food ($\theta^{*,k}_{\mathtt{int}} = 0$). In this case, by having *unlimited* food resources available to both agents, they don't have to signal each other for socially-acceptable feeding behaviors. However, this does not mean that in such cases they should ignore all intrinsic motivation provided by the social reward features. For example, due to the placement policy after eating, in the beginning the agents might learn to obtain reward by always trying to eat the food resource located on the left side while ignoring the fact that there is also another resource in the right side of the environment. In such cases, as the optimal parameter vector $\boldsymbol{\theta}^*$ indicates, $\phi^k_{\mathfrak{ext}}$ plays an important



(a) 2-2-2 Foraging Scenario

(b) 3-2-3 Foraging Scenario

(c) 2-1-1 Foraging Scenario

Figure B.8: Evolution of the fitness of the social group in the 2-2-2 Foraging Scenario, 3-2-3 Foraging Scenario and 2-1-1 Foraging Scenario.

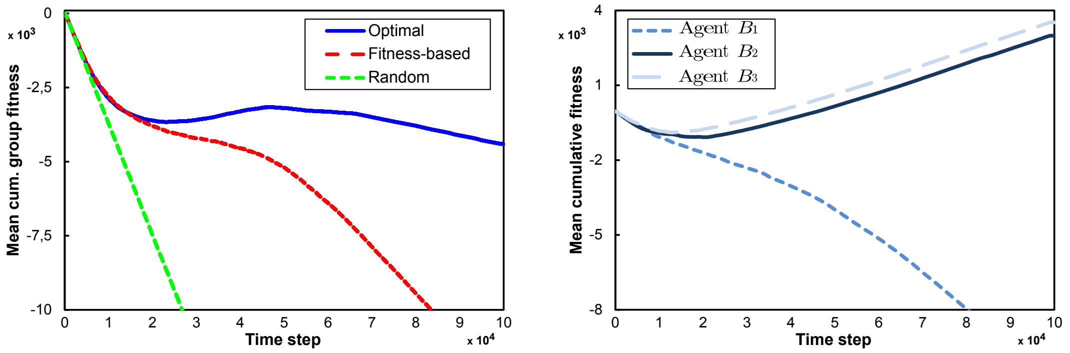
role by reprehending "selfish" behaviors during direct competition for food.

Having a Stronger Agent

In the 2-1-1 Foraging Scenario the objective was to examine whether the social behavior of the two agents observed in the 2-1-2 Foraging Scenario arose out of a *need* to cooperate with each other to avoid starvation if one agent gets to be the last to learn. In this scenario, by letting both agents depart from the same position and having Agent 1 always overpower Agent 2, Agent 1 strictly has *no need to cooperate* with Agent 2. However, even in this situation, we observe that socially-aware behavior emerges leading to resource sharing, as illustrated in Figure B.8(c). This can also be seen from the optimal parameter vector $\boldsymbol{\theta}^*$ obtained for this scenario in Table B.2, which places significant consideration in both $\phi^k_{\mathtt{ext}}$ and $\phi^k_{\mathtt{int}}$ reward features.

Emerging Alliances

The results of the 3-1-3 Foraging Scenario are in line with those reported in Section 6.4.4 for the 3-1-3 Scenario. Due to the placement policy of the agents in the environment based on a "social ranking", we again observe a behavior strategy where only two of the agents, namely Agent 2 and Agent 3 feed in turns, leaving Agent 1 to starve, as can be observed in Figure B.9(b) depicting the



(a) 3-1-3 Foraging Scenario

(b) 3-1-3 Foraging Scenario individual

Figure B.9: Evolution of fitness in the 3-1-3 Foraging Scenario. (a) social group fitness; (b) individual fitness of the three "optimal" agents.

individual fitness attained by each agent of the optimal group. However, in this experiment we purposely set the penalty for being hungry, β_{hp}, so that the group would not benefit as a whole if *any* of its members was not increasing its fitness. As such, the result of having two agents sharing food while leaving the more "low-ranked agent" to starve in this case is a negative net value in terms of the fitness of the whole social group, as illustrated in Figure B.9(a). We again note that this "alliance" strategy was the better strategy learned by the agents given the test conditions.

Bibliography

P. Abbeel and A. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the 21st International Conference on Machine learning*, ICML '04, pages 1–, New York, NY, USA, 2004. ACM.

S. Abdallah and V. Lesser. A multiagent reinforcement learning algorithm with non-linear dynamics. *Journal of Artificial Intelligence Research*, 33:521–549, 2008.

N. Abe, N. Verma, C. Apte, and R. Schroko. Cross channel optimized marketing by reinforcement learning. In *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge discovery and Data Mining*, KDD '04, pages 767–772, New York, USA, 2004. ACM.

H. Ahn and R. Picard. Affective cognitive learning and decision making: The role of emotions. In *Proceedings of the 18th European Meeting on Cybernetics and Systems Research*, pages 1–6, 2006.

P. Amorapanth, J. LeDoux, and K. Nader. Different lateral amygdala outputs mediate reactions and actions elicited by a fear-arousing stimulus. *Nature Neuroscience*, 3(1):74–79, 2000.

J. Anderson. *Learning and Memory: An Integrated Approach.* Wiley, 2000.

J. Armony, D. Servan-Schreiber, J. Cohen, and J. LeDoux. Computational modeling of emotion: Explorations through the anatomy and physiology of fear conditioning. *Trends in Cognitive Sciences*, 1(1):28–34, 1997.

M. Arnold. *Emotion and personality. Vol. I. Psychological aspects.* Columbia University Press, Oxford, England, 1960.

R. Axelrod and W. Hamilton. The evolution of cooperation. *Science*, 211(4489):1390–1396, 1981.

R. M. Axelrod. *The evolution of cooperation.* Basic Books, Inc., Publishers, New York, USA, 1984.

J. Bach. *Principles of synthetic intelligence: PSI, an architecture of motivated cognition.* Oxford University Press, 2009.

J. Bagnell and J. Schneider. Autonomous helicopter control using reinforcement learning policy search methods. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1615–1620. IEEE, 2001.

L. Baird. Advantage updating. Technical Report WL-TR-93-1146, Wright Laboratory, Wright-Patterson Air Force Base, 1993.

C. Balkenius and J. Morén. Computational models of classical conditioning: a comparative study. In *Proceedings of the 5th international Conference on simulation of adaptive behavior on From animals to animats 5*, volume 3, pages 348–353, Cambridge, MA, USA, 1998. MIT Press.

A. Barto and O. Şimşek. Intrinsic motivation for reinforcement learning systems. In *Proceedings of the 13th Yale Workshop on Adaptive and Learning Systems*, pages 113–118, 2005.

A. Barto, R. Sutton, and C. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, & Cybernetics*, 13(5):834–846, 1983.

A. Bechara, H. Damasio, and A. Damasio. Emotion, decision making and the orbitofrontal cortex. *Cerebral Cortex*, 10(3):295–307, Mar. 2000.

G. Becker. A Theory of Social Interactions. Working Paper 42, National Bureau of Economic Research, June 1974.

C. Becker-Asano and I. Wachsmuth. Affect simulation with primary and secondary emotions. *Intelligent Virtual Agents*, 5208:15–28, 2008.

R. Bellman. *Dynamic Programming.* Dover Publications, Inc., 2003.

D. Bentivegna, A. Ude, C. Atkeson, and G. Cheng. Humanoid robot learning and game playing using PC-based vision. In *2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2449–2454. IEEE, 2002.

T. Bergstrom. On the evolution of altruistic ethical rules for siblings. *American Economic Review*, 85(1):58–81, 1995.

M. Bouton. Context, ambiguity, and unlearning: Sources of relapse after behavioral extinction. *Biological Psychiatry*, 52(10):976–986, 2002.

R. Brafman. R-MAX - A General Polynomial Time Algorithm for Near-Optimal Reinforcement Learning. *Journal of Machine Learning Research*, 3:213–231, 2003.

J. Bratman, S. Singh, J. Sorg, and R. Lewis. Strong Mitigation : Nesting Search for Good Policies Within Search for Good Reward. In *Proceedings of the 11th International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS 2012, pages 407–414, 2012.

D. J. Broekens. *Affect and Learning: a computational analysis.* Doctoral thesis, Leiden University, 2007.

D. J. Broekens, W. Kosters, and F. Verbeek. On affect and self-adaptation: Potential benefits of valence-controlled action-selection. *Lecture Notes in Computer Science*, 4527:357–366, 2007.

L. Busoniu, R. Babuska, and B. De Schutter. A Comprehensive Survey of Multiagent Reinforcement Learning. *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 38(2):156–172, 2008.

D. Cañamero. Modeling motivations and emotions as a basis for intelligent behavior. In *Proceedings of the 1st International Conference on Autonomous Agents*, pages 148–155, 1997.

R. Cardinal, J. Parkinson, J. Hall, and B. Everitt. Emotion and motivation: the role of the amygdala, ventral striatum, and prefrontal cortex. *Neuroscience and Biobehavioral Reviews*, 26 (3):321–352, 2002.

G. Caridakis, A. Raouzaiou, E. Bevacqua, M. Mancini, K. Karpouzis, L. Malatesta, and C. Pelachaud. Virtual agent multimodal mimicry of humans. *Language Resources and Evaluation*, 41:367–388, 2007.

A. Cassandra. Acting optimally in partially observable stochastic domains. In *Proceedings of the 12th National Conference on Artificial Intelligence*, AAAI'94, Seattle, WA, 1994.

A. Cassandra. *Exact and Approximate Algorithms for Partially Observable Markov Decision Processes.* Phd, Brown University, 1998.

C. Claus and C. Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the 15th National Conference on Artificial Intelligence*, pages 746–752, 1998.

M. Colby and K. Tumer. Shaping Fitness Functions for Coevolving Cooperative Multiagent Systems. In *Proceedings of the 11th International Joint Conference on Autonomous Agents and Multiagent Systems*, 2012.

O. Şimşek and A. Barto. An intrinsic reward mechanism for efficient exploration. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 833–840, 2006.

A. Damasio. *Descartes' Error: Emotion, Reason, and the Human Brain.* G.P. Putnam, New York, 1994.

C. Darwin. *The Expression of the Emotions in Man and Animals.* John Murray, London, UK, 1872.

C. Darwin. *The Origin of Species.* Bridge-Logos, Alachua, FL, USA, 2009.

M. Dawkins. Animal Minds and Animal Emotions. *American Zoologist*, 40(6):883–888, 2000.

R. Dawkins. *The Selfish Gene.* Oxford University Press, New York, USA, 30th anniv. ed. edition, 2006.

F. de Waal. Putting the altruism back into altruism: The evolution of empathy. *Annual Review of Psychology*, 59(1):279–300, 2008.

E. Deci and R. Ryan. *Intrinsic Motivation and Self-Determination in Human Behavior.* Plenum Press, 1985.

S. DellaVigna, J. List, and U. Malmendier. Testing for Altruism and Social Pressure in Charitable Giving. *The Quarterly Journal of Economics*, 127(1):1–56, Jan. 2012.

J. Dias and A. Paiva. Feeling and reasoning: A computational model for emotional characters. *Progress in Artificial Intelligence*, 3808:127–140, 2005.

M. Dorigo and M. Colombetti. Robot shaping: Developing autonomous agents through learning. *Artificial Intelligence*, 71(2):321–370, 1994.

D. Dörner. *Bauplan für eine Seele [Blueprint for a Soul].* Reinbeck: Rowohlt, 1999.

M. El-Nasr, J. Yen, and T. Ioerger. Flame: Fuzzy logic adaptive model of emotions. *Autonomous Agents and Multi-Agent Systems*, 3:219–257, 2000.

C. Elliott. Research Problems in the Use of a Shallow Artificial Intelligence Model of Personality and Emotion. In *Proceedings of the 14th National Conference on Artificial Intelligence*, AAAI'94, pages 9–15, Menlo Park, California, 1994. AAAI Press.

P. Ellsworth and K. Scherer. *Handbook of the Affective Sciences*, chapter Appraisal processes in emotion, pages 572–595. Oxford University Press, New York and Oxford, 2003.

A. Falk and U. Fischbacher. A theory of reciprocity. *Games and Economic Behavior*, 54(2): 293–315, Feb. 2006.

N. Ferns, P. Panangaden, and D. Precup. Metrics for finite Markov decision processes. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 162–169, 2004.

P. Fidelman and P. Stone. Learning ball acquisition a physical robot. In *2004 International Symposium on Robotics and Automation (ISRA).* Citeseer, 2004.

D. Fogel. An introduction to simulated evolutionary optimization. *IEEE Transactions on Neural Networks*, 5(1):3–14, 1994.

S. Franklin and A. Graesser. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. *Lecture Notes in Computer Science*, 1193:21–36, 1997.

N. Frijda and B. Mesquita. The analysis of emotions: Dimensions of variation. In *What develops in emotional development? Emotions, personality, and psychotherapy*, pages 273–295. Plenum Press, New York, NY, US, 1998.

S. Gadanho. *Reinforcement Learning in Autonomous Robots : An Empirical Investigation of the Role of Emotions*. PhD thesis, University of Edinburgh, 1999.

S. Gadanho and J. Hallam. Robot Learning Driven by Emotions. *Adaptive Behavior*, 9(1):42–64, 2001.

P. Gebhard. ALMA: a layered model of affect. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 29–36, New York, NY, USA, 2005. ACM.

H. Ginsburg and S. Opper. *Piaget's theory of intellectual development*. Englewood Cliffs, NJ: Prentice Hall, 3rd edition, 1988.

J. Goldsmith and M. Mundhenk. Competition adds complexity. In *Adv. Neural Information Proc. Systems*, volume 20, 2007.

J. Gratch, S. Marsella, N. Wang, and B. Stankovic. Assessing the validity of appraisal-based models of emotion. In *3rd International Conference on Affective Computing and Intelligent Interaction*, ACII 2009, pages 1–8, 2009.

P. Griffiths. Is Emotion a Natural Kind? In *Thinking about feeling: Contemporary philosophers on emotions*, chapter 15, pages 233–249. Oxford University Press, Oxford and New York, 2004.

C. Guestrin, R. Patrascu, and D. Schuurmans. Algorithm-directed exploration for model-based reinforcement learning in factored MDPs. In *Proceedings of the 19th International Conference on Machine Learning*, pages 235–242, 2002.

W. D. Hamilton. The genetical evolution of social behaviour. I. *Journal of theoretical biology*, 7 (1):1–16, July 1964.

J. Han, J. Pei, Y. Yin, and R. Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery*, 8:53–87, 2004.

T. Hester and P. Stone. Intrinsically motivated model learning for a developing curious agent. In *2012 IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, pages 1–6. IEEE, 2012.

T. Hester, M. Lopes, and P. Stone. Learning Exploration Strategies in Model-Based Reinforcement Learning. In *12th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS 2013, page 8, St. Paul, MN, USA, 2013. ACM.

P. Holland and M. Gallagher. Amygdala circuitry in attentional and representational processes. *Trends in Cognitive Sciences*, 3(2):65–73, 1999.

E. Hudlicka, C. Becker-Asano, S. Payr, K. Fischer, R. Ventura, I. Leite, A. Paiva, and C. von Scheve. Social interaction with robots and agents: Where do we stand, where do we go? In *3rd International Conference on Affective Computing and Intelligent Interaction*, ACII 2009, pages 1 –6, sept. 2009.

C. Isbell, M. Kearns, S. Singh, C. Shelton, P. Stone, and D. Kormann. Cobot in LambdaMOO: An adaptive social statistics agent. *Autonomous Agents and Multi-Agent Systems*, 13(3):327–354, May 2006.

A. Isen. Some ways in which positive affect influences decision making and problem solving. In *Handbook of Emotions*, chapter 10, pages 548–573. The Guilford Press, New York, NY, US, 3rd edition, 2008.

P. Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50, 1912.

N. Jong and P. Stone. State abstraction discovery from irrelevant state variables. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 752–757, 2005.

L. Kaelbling, M. Littman, and A. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

L. Kaelbling, M. Littman, and A. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, May 1998.

S. Kakade and J. Langford. Approximately Optimal Approximate Reinforcement Learning. In *Proceedings of the 19th International Conference on Machine Learning*, ICML 2002, pages 267–274, Sydney, Australia, 2002. Morgan Kaufmann.

F. Kaplan and P. Oudeyer. Motivational principles for visual know-how development. In *Proceedings of the 3rd International Workshop on Epigenetic Robotics*, pages 73–80, 2003.

F. Kaplan and P. Oudeyer. Intrinsically motivated machines. In M. Lungarella, F. Iida, J. Bongard, and R. Pfeifer, editors, *50 Years of Artificial Intelligence*, pages 304–315, 2007.

M. Kearns and D. Koller. Efficient reinforcement learning in factored MDPs. In *Proceedings of the 1999 International Joint Conference on Artificial Intelligence*, pages 740–747, 1999.

M. Kearns and S. Singh. Near-Optimal Reinforcement Learning in Polynomial Time. *Machine Learning*, 49(2-3):209–232, 2002.

E. Kensinger. Remembering emotional experiences: The contribution of valence and arousal. *Reviews in the Neurosciences*, 15:241–252, 2004.

S. Killcross, T. Robbins, and B. Everitt. Different types of fear-conditioned behaviour mediated by separate nuclei within amygdala. *Nature*, 388(6640):377–380, 1997.

N. Kohl and P. Stone. Policy gradient reinforcement learning for fast quadrupedal locomotion. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'04)*, pages 2619–2624, 2004.

G. Konidaris and A. Barto. An adaptive robot motivational system. In *Proceedings of the 9th International Conference on Simulation of Adaptive Behavior*, pages 346–356, 2006.

J. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, 1992.

J. Koza. *Genetic programming II: automatic discovery of reusable programs.* MIT Press, May 1994.

M. Kroon and S. Whiteson. Automatic feature selection for model-based reinforcement learning in factored MDPs. In *Proceedings of the 2009 International Conference on Machine Learning and Applications*, pages 324–330, 2009.

C. Kwok and D. Fox. Reinforcement learning for sensing strategies. In *Proceedings of the International Confrerence on Intelligent Robots and Systems (IROS 2004)*, pages 3158–3163. IEEE, 2004.

R. Lazarus. *Psychological stress and the coping process.* McGraw-Hill, New York, NY, US, 1966.

R. Lazarus. Relational meaning and discrete emotions. In *Appraisal processes in emotion: Theory, methods, research.*, pages 37–67. Oxford University Press, New York, NY, US, 2001.

J. LeDoux. Emotion circuits in the brain. *Annual review of neuroscience*, 23(1):155–184, 2000.

J. LeDoux. The amygdala. *Current Biology*, 17(20):868–874, 2007.

H. Leventhal and K. Scherer. The relationship of emotion to cognition: A functional approach to a semantic controversy. *Cognition & Emotion*, 1(1):3–28, 1987.

L. Li, T. Walsh, and M. Littman. Towards a unified theory of state abstraction for MDPs. In *Proceedings of the 9th International Symposium on Artificial Intelligence and Mathematics*, pages 531–539, 2006.

A. Lipkus. A proof of the triangle inequality for the tanimoto distance. *J. Mathematical Chemistry*, 26(1):263–265, 1999.

C. L. Lisetti and P. Gmytrasiewicz. Can a rational agent afford to be affectless? a formal approach. *Applied Artificial Intelligence*, 16(7-8):577–609, 2002.

M. Littman. Memoryless policies: Theoretical limitations and practical results. In *From Animals to Animats 3: Proceedings of the 3rd International Conference on Simulation of Adaptive Behavior*, pages 238–247, 1994.

M. Littman, A. Cassandra, and L. Kaelbling. Learning policies for partially observable environments: Scaling up. In *Proceedings of the 12th International Conference on Machine Learning*, ICML'95, pages 362–370, San Francisco, CA, 1995. Morgan Kaufmann.

B. Liu, S. Singh, R. L. Lewis, and S. Qin. Optimal rewards in multiagent teams. In *Proc. 2nd Joint IEEE Int. Conf. on Development and Learning and Epigenetic Robotics*, pages 1–8. IEEE, Nov. 2012.

J. Loch and S. Singh. Using eligibility traces to find the best memoryless policy in partially observable markov decision processes. In *Proceedings of the 15th International Conference on Machine Learning*, ICML'98, pages 323–331. Morgan Kaufmann, 1998.

M. Lopes, T. Lang, M. Toussaint, and P.-Y. Oudeyer. Exploration in model-based reinforcement learning by empirically estimating learning progress. In *Advances in Neural Information Processing Systems 25*, pages 206–214. 2012.

P. Maes. Modeling adaptive autonomous agents. *Artificial life*, 1(1&2):135–162, 1994.

R. Marinier. *A computational unification of cognitive control, emotion, and learning.* Phd thesis, University of Michigan, Ann Arbor, MI, 2008.

S. Marsella and J. Gratch. Ema: A process model of appraisal dynamics. *Cognitive Systems Research*, 10(1):70 – 90, 2009.

S. Marsella, J. Gratch, and P. Petta. *Blueprint for Affective Computing*, chapter Computational models of emotion, pages 21–44. Oxford University Press, 2010.

M. Mataric. Reward functions for accelerated learning. In *Proceedings of the 11th International Conference on Machine Learning*, ICML'94, page 6, New Brunswick, NJ, USA, 1994.

J. Maynard Smith. *Evolution and the Theory of Games.* Cambridge University Press, 1982.

J. Maynard Smith and G. Price. The logic of animal conflict. *Nature*, 246(5427):15–18, 1973.

M. Minsky. *The society of mind.* Simon & Schuster, Inc., New York, NY, USA, 1986.

T. Mitchell. *Machine Learning.* McGraw-Hill, 1997.

J. Moody and M. Saffell. Learning to trade via direct reinforcement. *IEEE Transactions on Neural Networks*, 12(4):875–889, Jan. 2001.

A. Moore and C. Atkeson. Prioritized sweeping: Reinforcement learning with less data and less real-time. *Machine Learning*, 13:103–130, 1993.

M. Morgan and J. LeDoux. Differential Contribution of Dorsal and Ventral Medial Prefrontal Cortex to the Acquisition and Extinction of Conditioned Fear in Rats. *Behavioral Neuroscience*, 109(4):681–688, 1995.

M. Morgan, J. LeDoux, and J. Schulkin. Ventral medial prefrontal cortex and emotional perseveration: the memory for prior extinction training. *Behavioural Brain Research*, 146(1-2):121–130, 2003.

K. Nader, J. LeDoux, and G. Schafe. Fear memories require protein synthesis in the amygdala for reconsolidation after retrieval. *Nature*, 406(6797):722–726, 2000.

R. Nair, M. Tambe, and S. Marsella. The role of emotions in multiagent teamwork. In J.-M. Fellous and M. A. Arbib, editors, *Who needs emotions? The brain meets the robot*, chapter 11, pages 311–333. Oxford University Press, 2005.

N. Naqvi, B. Shiv, and A. Bechara. The role of emotion in decision making. *Current Directions in Psychological Science*, 15(5):260–264, 2006.

S. Nason and J. Laird. Soar-RL: integrating reinforcement learning with Soar. *Cognitive Systems Research*, 6(1):51–59, Mar. 2005.

A. Ng and S. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the 17th International Conference on Machine Learning*, ICML 2000, pages 663–670. Morgan Kaufmann, 2000.

A. Ng, D. Harada, and S. Russel. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the 16th International Conference on Machine Learning*, ICML'99, pages 278–287, 1999.

A. Ng, A. Coates, M. Diel, V. Ganapathi, J. Schulte, B. Tse, E. Berger, and E. Liang. Autonomous inverted helicopter flight via reinforcement learning. In *Experimental Robotics IX*, volume 21 of *Springer Tracts in Advanced Robotics*, pages 363–372. Springer Berlin / Heidelberg, 2006.

S. Niekum, A. Barto, and L. Spector. Genetic programming for reward function search. *IEEE Transactions on Autonomous Mental Development*, 2(2):83–90, June 2010.

K. Oatley and J. Jenkins. *Understanding Emotions.* Wiley-Blackwell, 2 edition, 2006.

A. Ortony, G. Clore, and A. Collins. *The cognitive structure of emotions.* Cambridge University Press, New York, NY, US, 1988.

P. Oudeyer, F. Kaplan, and V. Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation*, 11(2):265–286, 2007.

L. Panait and S. Luke. Cooperative Multi-Agent Learning: The State of the Art. *Autonomous Agents and Multi-Agent Systems*, 11(3):387–434, Nov. 2005.

M. Pantic and M. Bartlett. Machine analysis of facial expressions. In *Face Recognition*, pages 377–416. I-Tech Education and Publishing, Vienna, Austria, 2007.

I. Pavlov. *Conditioned reflex: An investigation of the physiological activity of the cerebral cortex*, volume 3. Oxford University Press: Humphrey Milford, 1927.

E. Phelps and J. LeDoux. Contributions of the Amygdala to Emotion Processing: From Animal Models to Human Behavior. *Neuron*, 48(2):175–187, 2005.

R. Picard. *Affective Computing.* MIT Press, 2000.

S. Proper and P. Tadepalli. Scaling model-based average-reward reinforcement learning for product delivery. In *Proceedings of the 17th European Conference on Machine Learning*, ECML 2006, pages 735–742. Springer, 2006.

M. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Wiley, 2009.

J. Randløv and P. Alstrøm. Learning to drive a bicycle using reinforcement learning and shaping. In *Proceedings of the 15th International Conference on Machine Learning*, ICML'98, page 9, Madison, Wisconsin, USA, 1998.

A. Rapoport and A. Chammah. *Prisoner's Dilemma.* University of Michigan Press, 1965.

S. Reilly and J. Bates. Building emotional agents. Technical Report CMU-CS-92-143, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1992.

R. Reisenzein. Emotions as metarepresentational states of mind: Naturalizing the belief–desire theory of emotion. *Cognitive Systems Research*, 10(1):6–20, Mar. 2009.

R. Rescorla and A. Wagner. A Theory of Pavlovian Conditioning: Variations in the Effectiveness of Reinforcement and Nonreinforcement. In *Classical Conditioning II: Current Research and Theory*, pages 64–99. Appleton-Century-Crofts, New York, USA, 1972.

C. Ribeiro and C. Szepesvári. Q-learning combined with spreading: Convergence and results. In *Proceedings of the ISRF-IEE International Conference on Intelligent and Cognitive Systems*, pages 32–36, 1996.

I. Roseman. A model of appraisal in the emotion system: Integrating theory, research, and applications. In *Appraisal processes in emotion: Theory, methods, research.*, pages 68–91. Oxford University Press, New York, NY, US, 2001.

I. Roseman and C. Smith. Appraisal theory: Overview, assumptions, varieties, controversies. In *Appraisal processes in emotion: Theory, methods, research.*, pages 3–19. Oxford University Press, New York, NY, US, Nov. 2001.

T. Rumbell, J. Barnden, S. Denham, and T. Wennekers. Emotions in autonomous agents: comparative analysis of mechanisms and functions. *Autonomous Agents and Multi-Agent Systems*, 25(1):1–45, Feb. 2011.

G. Rummery and M. Niranjan. On-line *q*-learning using connectionist systems. Technical Report CUED/F-INFENG/TR 166, Cambridge University Engineering Department, 1994.

P. Rusmevichientong, J. Salisbury, L. Truss, B. Van Roy, and P. Glynn. Opportunities and challenges in using online preference data for vehicle pricing: A case study at General Motors. *Journal of Revenue and Pricing Management*, 5(1):45–61, Apr. 2006.

S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, second edition, 2003.

R. Ryan and E. Deci. Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions. *Contemporary educational psychology*, 25(1):54–67, 2000.

M. Salichs and M. Malfaz. Using emotions on autonomous agents: The role of happiness, sadness and fear. In *Proceedings of the Annual Conference on Ambient Intelligence and Simulated Behavior*, pages 157–164, 2006.

M. Salichs and M. Malfaz. A New Approach to Modeling Emotions and Their Use on a Decision-Making System for Artificial Agents. *IEEE Transactions on Affective Computing*, 3(1):56–68, Jan. 2012.

A. Samuel. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 3(3):210–229, July 1959.

J. Schaeffer, M. Hlynka, and V. Jussila. Temporal difference learning applied to a high-performance game-playing program. In *Proceedings of the 17th international Joint Conference on Artificial intelligence*, pages 529–534. Morgan Kaufmann Publishers Inc., Aug. 2001.

M. Schembri, M. Mirolli, and G. Baldassarre. Evolving internal reinforcers for an intrinsically motivated reinforcement-learning robot. In *Proceedings of the 6th International Conference on Development and Learning*, pages 282–287, 2007.

K. Scherer. Appraisal considered as a process of multilevel sequential checking. In *Appraisal processes in emotion: Theory, methods, research.*, volume 92, chapter 5, page 120. Oxford University Press, New York, NY, US, 2001.

K. Scherer. The dynamic architecture of emotion: Evidence for the component process model. *Cognition & Emotion*, 23(7):1307–1351, Nov. 2009.

M. Scheutz. Useful roles of emotions in artificial agents: A case study from artificial life. In *Proceedings of the 19th national conference on artificial intelligence*, AAAI 2004, pages 42–48. AAAI Press, 2004.

J. Schmidhuber. Formal Theory of Creativity, Fun, and Intrinsic Motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, 2010.

A. Schorr. Appraisal The Evolution of an Idea. In *Appraisal processes in emotion: Theory, methods, research.*, pages 20–34. Oxford University Press, New York, NY, US, 2001.

R. Schuster and A. Perelberg. Why cooperate? An economic perspective is not enough. *Behavioural processes*, 66(3):261–277, July 2004.

O. Selfridge and R. Sutton. Training and tracking in robotics. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, IJCAI'85, pages 670–672, San Francisco, CA, USA, 1985. Morgan Kaufmann Publishers Inc.

S. Sen, M. Sekaran, and J. Hale. Learning to coordinate without sharing information. In *Proceedings of the 12th National Conference on Artificial Intelligence*, AAAI'94, pages 426–431, 1994.

P. Sequeira and C. Antunes. Real-time sensory pattern mining for autonomous agents. In *Proceedings of the 6th International Workshop on Agents and Data Mining Interaction*, ADMI-10, pages 71–83, 2010.

P. Sequeira, F. S. Melo, and A. Paiva. Emotion-based Intrinsic Motivation for Learning Agents. In *Proceedings of the 4th International Conference on Affective Computing and Intelligent Interaction*, ACII 2011, 2011a.

P. Sequeira, F. S. Melo, R. Prada, and A. Paiva. Emerging Social Awareness: Exploring Intrinsic Motivation in Multiagent Learning. In *Proceedings of the 1st Joint IEEE International Conference on Development and Learning and on Epigenetic Robotics*, ICDL-EPIROB 2011. IEEE, 2011b.

P. Sequeira, F. S. Melo, and A. Paiva. Learning by appraising: An emotion-based approach for intrinsic reward design. Technical report, GAIPS / INES-ID / IST, 2012.

P. Sequeira, F. S. Melo, and A. Paiva. An Associative State-Space Metric for Learning in Factored MDPs. In *Proceedings of the 16th Portuguese Conference on Artificial Intelligence (to appear)*, EPIA 2013, 2013.

M. Si, S. Marsella, and D. Pynadath. Modeling appraisal in theory of mind reasoning. *Autonomous Agents and Multi-Agent Systems*, 20(1):14–31, 2010.

K. Sigmund and M. Nowak. Evolutionary game theory. *Current Biology*, 9(14):R503–R505, July 1999.

H. Simon. Motivational and emotional controls of cognition. *Psychological Review*, 74(1):29–39, 1967.

S. Singh, T. Jaakkola, and M. Jordan. Learning without state-estimation in partially observable Markovian decision processes . In *Proceedings of the 11th International Conference on Machine Learning*, pages 284–292, 1994.

S. Singh, D. Litman, M. Kearns, and M. Walker. Optimizing dialogue management with reinforcement learning: Experiments with the NJFun system. *Journal of Artificial Intelligence Research*, 16(1):105–133, 2002.

S. Singh, R. Lewis, and A. Barto. Where do rewards come from? In *Proceedings of the 31st Annual Conference on Cognitive Science Society*, pages 2601–2606, 2009.

S. Singh, R. Lewis, A. Barto, and J. Sorg. Intrinsically motivated reinforcement learning: An evolutionary perspective. *IEEE Transactions on Autonomous Mental Development*, 2(2):70–82, 2010.

B. Skinner. *The behavior of organisms: an experimental analysis.* D. Appleton-Century Company, incorporated, 1938.

C. Smith and L. Kirby. Consequences require antecedents: Toward a process model of emotion elicitation. In *Feeling and thinking: The role of affect in social cognition.*, chapter 4, pages 83–106. Cambridge University Press, New York, NY, US, 2000.

C. Smith and L. Kirby. Putting appraisal in context: Toward a relational model of appraisal and emotion. *Cognition & Emotion*, 23(7):1352–1372, Nov. 2009.

J. Sorg, S. Singh, and R. Lewis. Internal rewards mitigate agent boundedness. In *Proceedings of the 27th International Conference on Machine Learning*, pages 1007–1014, 2010a.

J. Sorg, S. Singh, and R. Lewis. Reward Design via Online Gradient Ascent. *Advances in Neural Information Processing Systems 23*, 23:1–9, 2010b.

P. Stone and R. Sutton. Scaling reinforcement learning toward robocup soccer. In *Proceedings of the 18th International Conference on Machine Learning*, ICML'01, pages 537–544, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

A. Stout, G. Konidaris, and A. Barto. Intrinsically motivated reinforcement learning: A promising framework for developmental robot learning. In *Proceedings of the AAAI Symposium on Developmental Robotics*, 2005.

R. Sutton and A. Barto. Toward a Modern Theory of Adaptive Networks: Expectation and Prediction. *Psychological Review*, 88(2):135–170, 1981.

R. Sutton and A. Barto. A temporal-difference model of classical conditioning. In *Proceedings of the 9th Annual Conference on Cognitive Science Society*, pages 355–378, 1987.

R. Sutton and A. Barto. *Reinforcement Learning: An Introduction.* The MIT Press, Cambridge, MA, US, 1998.

C. Szepesvári. *Algorithms for Reinforcement Learning.* Morgan and Claypool, 2010.

G. Tesauro. Temporal difference learning and TD-Gammon. *Communications of the ACM*, 38(3): 58–68, Mar. 1995.

S. Thrun. Efficient exploration in reinforcement learning. Technical Report CMU-CS-92-102, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, 1992.

R. Trivers. The evolution of reciprocal altruism. *The Quarterly Review of Biology*, 46(1):35–57, 1971.

K. Tuyls and A. Nowé. Evolutionary game theory and multi-agent reinforcement learning. *Knowledge Engineering Review*, 20:63–90, 2005.

J. Velásquez. Modeling emotions and other motivations in synthetic agents. In *Proceedings of the 14th National Conference on Artificial Intelligence and 9th Conference on Innovative Applications of Artificial Intelligence*, AAAI'97/IAAI'97, pages 10–15. AAAI Press, 1997.

R. Ventura and C. Pinto-Ferreira. Responding efficiently to relevant stimuli using an emotion-based agent architecture. *Neurocomputing*, 72(13-15):2923–2930, 2009.

J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior.* Princeton University Press, 1944.

D. Walker and M. Davis. The role of amygdala glutamate receptors in fear learning, fear-potentiated startle, and extinction. *Pharmacology, biochemistry, and behavior*, 71(3):379–392, Mar. 2002.

C. Watkins. *Learning from delayed rewards.* PhD thesis, King's College, Cambridge University, 1989.

J. Watson. Psychology as the behaviorist views it. *Psychological Review*, 20(2):158–177, 1913.

E. Wiewiora. Potential-based shaping and Q-value initialization are equivalent. *Journal Of Artificial Intelligence Research*, 19:205–208, 2003.

M. Wooldridge. *An Introduction to Multiagent Systems.* John Wiley & Sons, Ltd., 2002.

X. Yan, P. Diaconis, P. Rusmevichientong, and B. Van Roy. Solitaire: Man versus machine. In *Advances in Neural Information Processing Systems 17*, pages 1553–1560. MIT Press, Cambridge, MA, 2005.