Emergence of Emotional Appraisal Signals in Reinforcement Learning Agents

Sequeira, P., F. S. Melo, and A. Paiva (2015). Emergence of Emotional Appraisal Signals in Reinforcement Learning Agents, Journal of Autonomous Agents and Multiagent Systems, 29(4):537-568.

Presented By: Grant Bosley and Zane Dush

Outline

- Background
- Identification of Optimal Sources of Information
  - Various scenarios
- Validation of Identified Sources
  - More scenarios
- Paper’s Conclusions
- Our Conclusions

Background

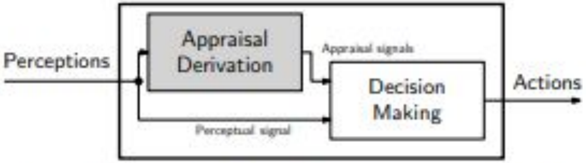


Fig. 1 General architecture of an emotional appraisal-based agent [23].[3]

Goal:

- Expand on the theory that emotional appraisal-like signals arise as natural candidates for sources of information to complement an agent’s perceptual capabilities and guide decision making.

Intrinsically motivated reinforced learning(IMRL) agents

- Use expanded reinforcement learning to overcome agents’ perceptual limitations (Partially Observable Markov Decision Processes)
- Provides framework to address Optimal Reward Problem (ORP)

Identification of Optimal Sources of Information

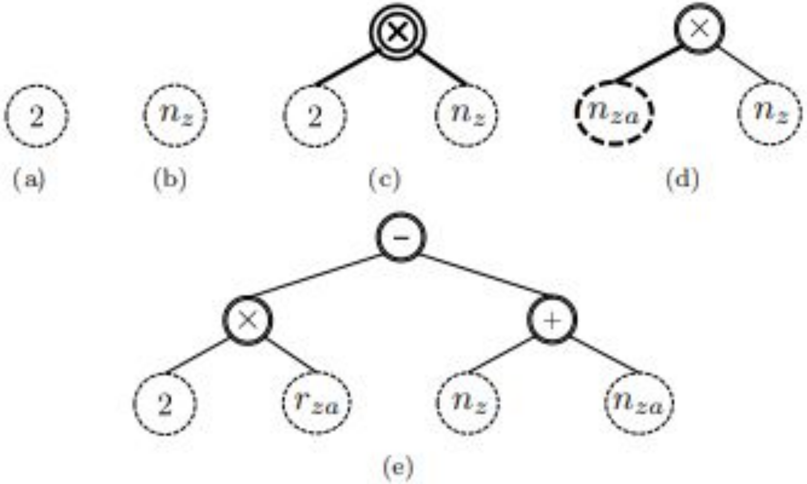
ORP using Genetic Programming(GP)

Evolutionary Procedure

Six scenarios:

- Hungry-Thirsty
- Lairs
- Moving-Preys
- Persistence
- Seasons
- Poisoned Prey

Genetic Programming



(a) : is a constant reward node

(b): is a node with variable reward function

(c): is a result of a crossover operation

The reward function of the tree is $r = 2n_z$

(d): is a mutation of ©

The new reward function is $r = n_{za}n_z$

(e): has a reward function of $r = 2r_{za}-(n_z+n_{za})$

Evolutionary Procedure

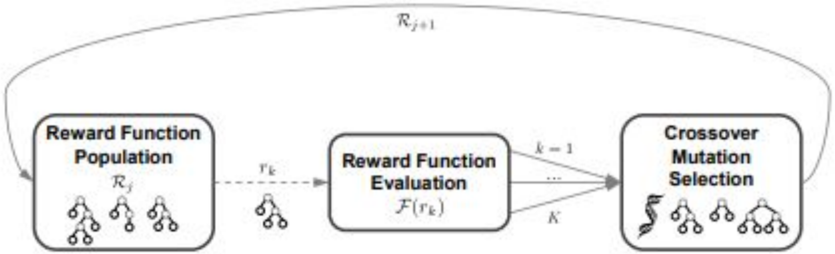


Fig. 5 The GP approach to the ORP, as proposed in [25]. In each generation j, a population

In each experiment in order to run the evolutionary procedure there are

50 independent initial populations

Each containing k = 100 elements

And run the evolutionary procedure for j = 50 generations for each population

Maintains the most fit elements

- The reward functions with highest fitness

Generates ten new random elements

The remaining 80 elements are generated either by mutating one element or through crossover.

Scenario Overall conditions

In all scenarios except Lairs the agent has four actions

- A = {N, S, E, W}, directional movement
- Prey is captured automatically, except in Lairs

In all scenarios except Lairs and Hungry-Thirsty the agent is only able to observe its own location (x:y) and whether it is collocated with a prey

Agents run prioritized sweeping with greedy exploration

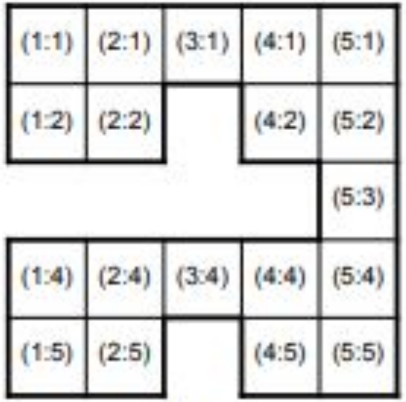
Hungry-Thirsty

The fitness of an agent is determined by the amount of food consumed

An agent can only eat if it is not thirsty

This can only be achieved by drinking

An agent becomes thirsty after drinking with a .2 probability at each time step



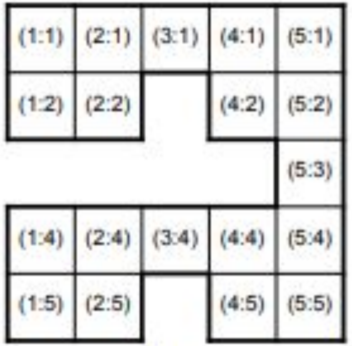
Lairs

The fitness of the agent is defined by the number of preys captured

The agent can perform six commands

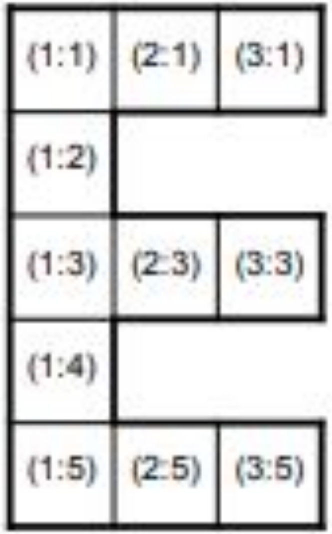
- N, S, E, W each corresponding to a direction to move in
- P, and C
    - P: Pulls prey from a lair, this agent now has only one time step to capture the prey
    - C: Captures a prey

Empty Lairs have a .1 probability of becoming filled at each time step



(1:1)	(2:1)	(3:1)	(4:1)	(5:1)
(1:2)	(2:2)		(4:2)	(5:2)
				(5:3)
(1:4)	(2:4)	(3:4)	(4:4)	(5:4)
(1:5)	(2:5)		(4:5)	(5:5)

Moving-Preys

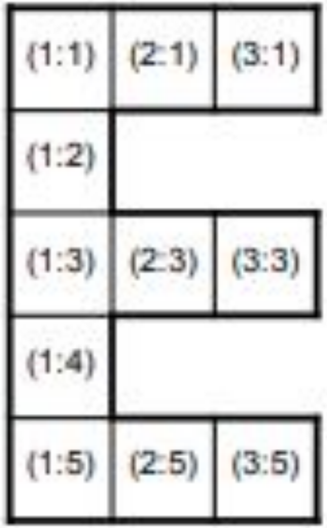


The agent’s fitness is again defined by the number of prey captured

At any given time step there is exactly one prey available in one of the end-of-corridor locations((3:1), (3:3), (3:5)).

Once a prey is captured it is removed from the scenario and another spawns at one of the two other locations

Persistence



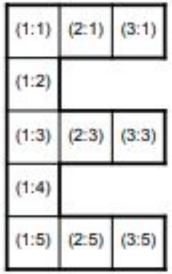
The scenario has two types of prey

- Hare: located at (3:1) and worth 1 point
- Rabbit: located at (3:5) and worth .01 point

The agent starts at (3:3) and when it captures a prey it gets reset to (3:3)

The environment also has a fence at (1:2)

- The fence is only active when moving up not down
- The fence takes Nt time steps to cross
- Each time it is crossed it the number of time steps required to cross it increases
- Nt = min{nt(fence) + 1; 30}



Seasons

Again the scenario has two types of prey

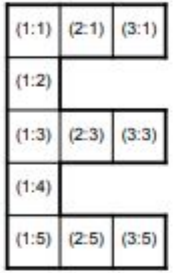
- Hare: located at (3:1) and worth 1 point
- Rabbit: located at (3:5) and worth .1 point

As with the Persistence scenario when a prey is captured the agent is reset to location(3:3)

There are two seasons that change every 5,000 time steps

- Initially selected from Hare or Rabbit season with equal probability
- Only one type of prey is available in each of the seasons
- In rabbit season every tenth rabbit the agent is negatively impacted, its fitness value is reduced by one

Poisoned Prey



(b)

This scenario is nearly identical to the Seasons scenario, except both rabbits a hares are available at every time step.

Rabbits have a value of .1

Healthy Hares have a value of 1

Poisoned Hares have a value of -1

The health status of hares changes every 5,000 time steps

Identified Sources

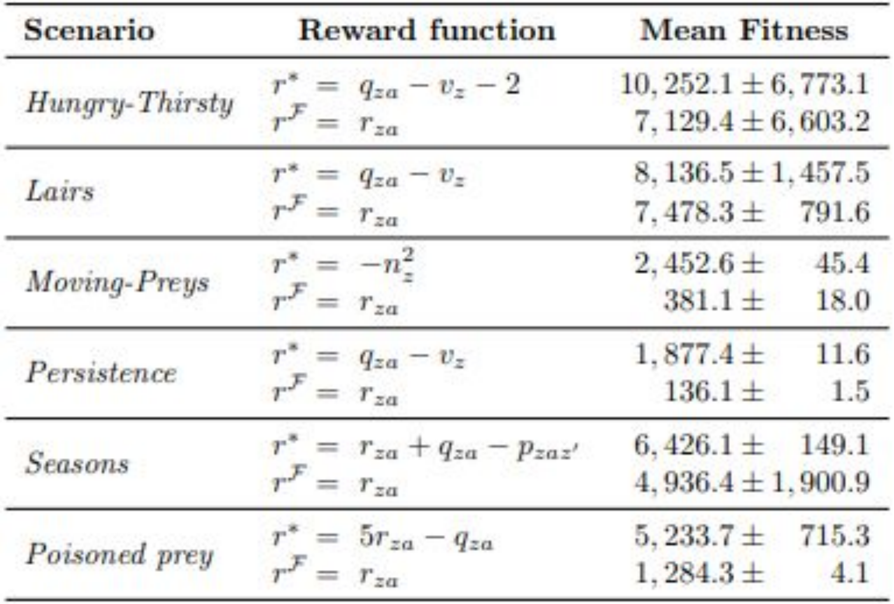
– φfit = r_{za} corresponds to the agent’s estimate of the fitness-based reward function.

– φrel = q_{za} corresponds to the estimated Q-function associated with r_{za}.

– φadv = $q_{za}-v_{z}$ corresponds to the estimated advantage function associated with r_{za}.

– φprd = $p_{zaz'}$, corresponds to the agent’s estimate of the transition probabilities.

– φfrq = $-n_{z}^{2}$ provides a (negative) measure of how novel z is given the agent’s observations.



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

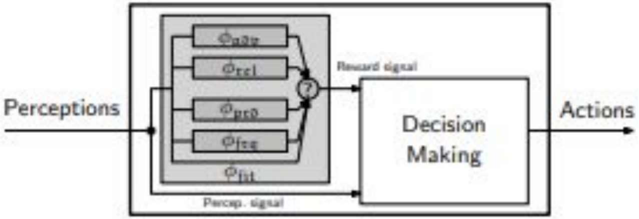
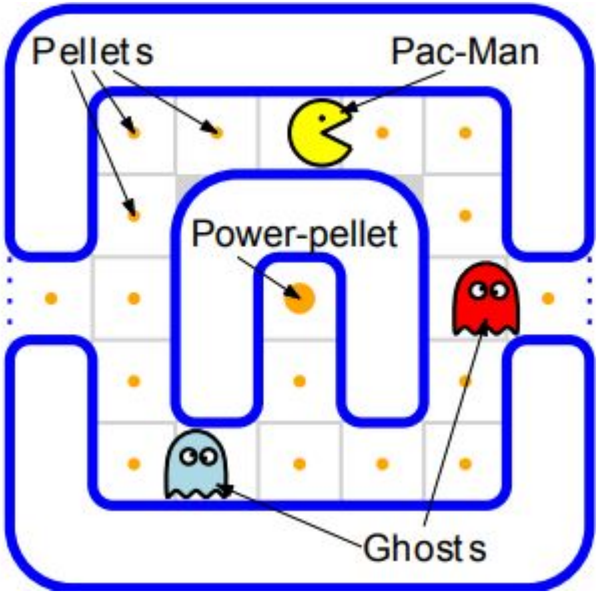


Fig. 7 Architecture for an agent using the identified sources of information.

Validation of Identified Sources

Four scenarios

- Power-pellet
- Eat-all-pellets
- Rewarding-pellets
- Pac-Man



Power-Pellets

Agent corresponds to Pac-Man and co-exists in the environment with two ghosts, the smart ghost and the keeper ghost. One pellet is available per run (the power-pellet), and is located in the center cell of the environment.

When Pac-Man consumes the power-pellet (by reaching its position), Pac-Man's fitness is increased by 0.8. Power-pellet enables Pac-Man to consume ghosts.

Run end criteria: – Pac-Man consumes both ghosts (which contributes to its fitness with a value of 1). – Pac-Man is captured by a ghost (which contributes to its fitness with a value of −1). – 20 time-steps have elapsed after the power-pellet was consumed.

Eat-all-pellets

Agent corresponds to Pac-Man and co-exists with only the smart ghost. Environment has 20 pellets total (one in each cell), which are consumed when Pac-Man visits the corresponding cell.

Consuming the power-pellet contributes to the fitness of the agent with a value of 0.5, but does not enable Pac-Man to consume the ghost.

Run end criteria: – Pac-Man consumes all 20 pellets (which contributes to its fitness with a value of 1). – Pac-Man is captured 3 times by the ghost before all pellets are consumed (which contributes to its fitness with a value of −0.5).

When the ghost captures Pac-Man, their positions are reset.

Rewarding-pellets

Agent corresponds to Pac-Man and co-exists with both the smart ghost and keeper ghost. Environment has 20 pellets (one in each cell), which are consumed when Pac-Man visits corresponding cell. Each consumed pellet contributes to the fitness of the agent with a value of 0.1, in the case of a regular pellet, or 0.8, in the case of the power-pellet (doesn’t enable Pac-Man to consume ghosts).

Run end criteria: – Pac-Man consumes all 20 pellets (which contributes to its fitness with a value of 1). – Pac-Man is captured by a ghost before all pellets are consumed (which contributes to its fitness with a value of −1).

Pac-Man

Agent corresponds to Pac-Man and co-exists with the smart ghost. Environment has 20 pellets as before. This time, consuming the power-pellet doesn’t contribute to the fitness of the agent, but does enable Pac-Man to consume the ghost.

Run end criteria: – Pac-Man consumes all 20 pellets (which contributes to its fitness with a value of 1). – Pac-Man is captured 3 times before all pellets are consumed (with no impact in fitness).

When the ghost captures Pac-Man, the fitness of the latter is decreased by a value of -0.1, and their positions are reset.

Validation Conclusions

Pac-Man agents that use only the fitness-based reward function are clearly inferior to those that use the additional sources of information determined previously.

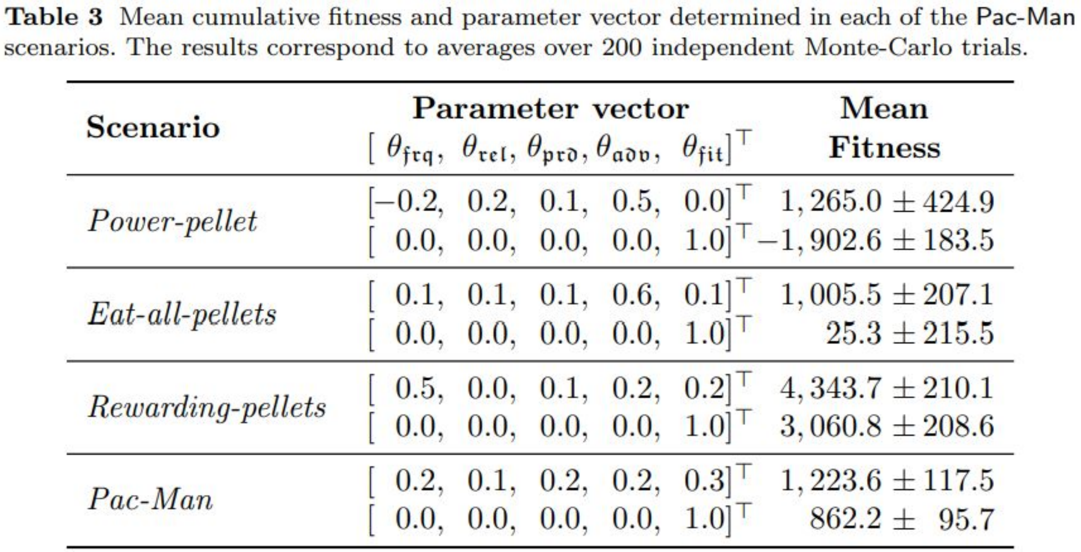


Table 3 Mean cumulative fitness and parameter vector determined in each of the Pac-Man scenarios. The results correspond to averages over 200 independent Monte-Carlo trials.

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

Paper’s Conclusions

Major emotional appraisal dimensions

- Novelty
- Intrinsic Pleasantness
- Motivational Bases
- Power and Coping
- Social Dimensions

Paper’s Conclusions

Reward features capture theme of the appraisal process

- Fitness
  - Φfit signals behaviors that directly enhance or reduce fitness.
- Relevance
  - Φrel denotes the impact of executing actions in some states for the agent’s fitness in the long-run.
- Advantage
  - Φadv denotes the (dis) advantage of executing actions in some states considering their future impact on fitness.
- Prediction
  - Φprd indicates how predictable the transition to some state is after the execution of an action in a previous state.
- Frequency
  - Φfrq indicates, this source of information punishes visits to states to which the agent is more accustomed to.

Paper’s Conclusions

Found that the kinds of evaluations they make about the agent’s relationship with its environment shared some properties with common dimensions of emotional appraisal.

Experiments show that the reward features 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).

Our Conclusions

Thought using Genetic Programming to determine the ORP was very good as it imitates biological evolution.

Seems well-supported that emotional appraisal signals emerge as a good complement to an agent’s perceptions (artificial or biological) when it comes to decision making.