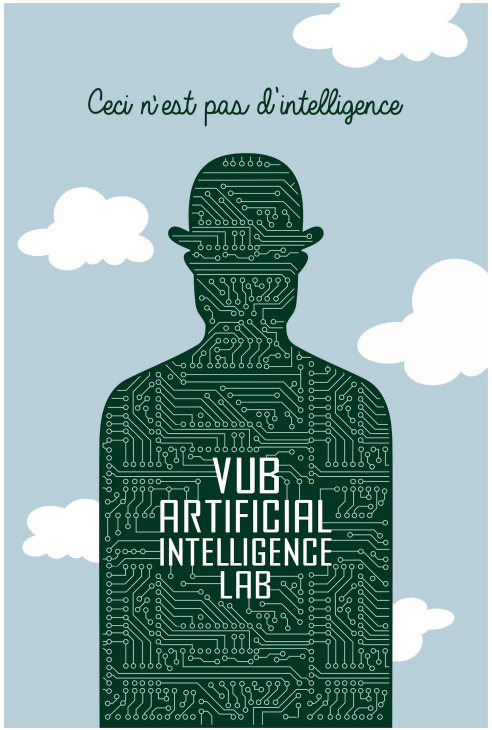


Handling Multiple Objectives in Single and Multi-Agent Reinforcement Learning

Ann Nowé (Vrije Universiteit Brussel)
Roxana Rădulescu (Utrecht University & Vrije Universiteit Brussel)

AAMAS Tutorial 2, Auckland, 2024
https://roxanaradulescu.com/#aamas2024

VUB ARTIFICIAL INTELLIGENCE RESEARCH GROUP
Utrecht University
@aibrussels
@rox_teo

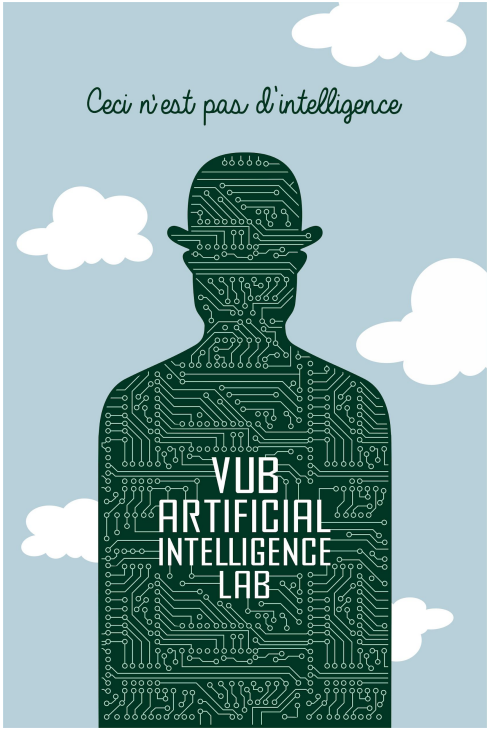
Nice to meet you!

https://forms.gle/kyNp4Mrmf1ZwKq1q6

Hi, I'm Ann!

- Full professor at Vrije Universiteit Brussel, Belgium
- Head of the AI lab at the VUB

- Research: Reinforcement Learning, multi-agent, multi-objective, trustworthiness



Hi, I'm Roxana!

- Assistant professor at Utrecht University, Netherlands
- Research: multi-objective reinforcement learning, multi-agent systems, multi-objective game theory

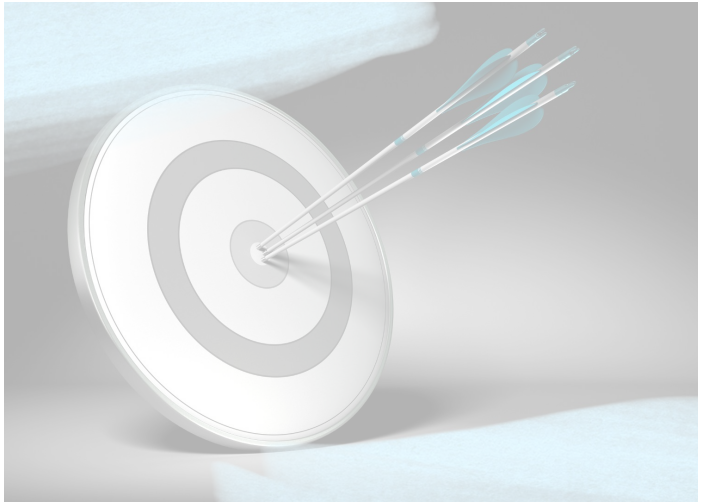
@rox_teo

http://roxanaradulescu.com

Tutorial Roadmap

Part 1 - Multi-objective decision making in single-agent systems

Motivation and core concepts



Why?

Multiple objectives

Because life is not simple

- What are your objectives for your current research project?
  - Publishing asap?
  - Quality of conference/journal?
  - Collaboration potential?
  - Flag-posting?
  - Increasing funding potential?
  - Finishing your PhD?

Because life is not simple

- What are your objectives for your current research project?
  - Publishing asap?
  - Quality of conference/journal?
  - Collaboration potential?
  - Flag-posting?
  - Increasing funding potential?
  - Finishing your PhD?
- How about your co-authors?

Scalar Reward Design Process

- Design/tweak scalar reward function
- (Re-)Train RL agent using new/updated reward function (may take hours or days)
- Evaluate the outcome (and try to figure out what went wrong!)

Scalar Reward Design Process

- **Design/tweak scalar reward function**
- **(Re-)Train RL agent using new/updated reward function (may take hours or days)**
- **Evaluate the outcome (and try to figure out what went wrong!)**

- Repeat until the desired agent behaviour is (finally) learned

- **Wasteful and time consuming** process - each trained agent must be discarded if the reward function changes

- Designers implicitly bake in trade-offs between different behaviours
    - **Should AI engineers make the decisions about these trade-offs?**

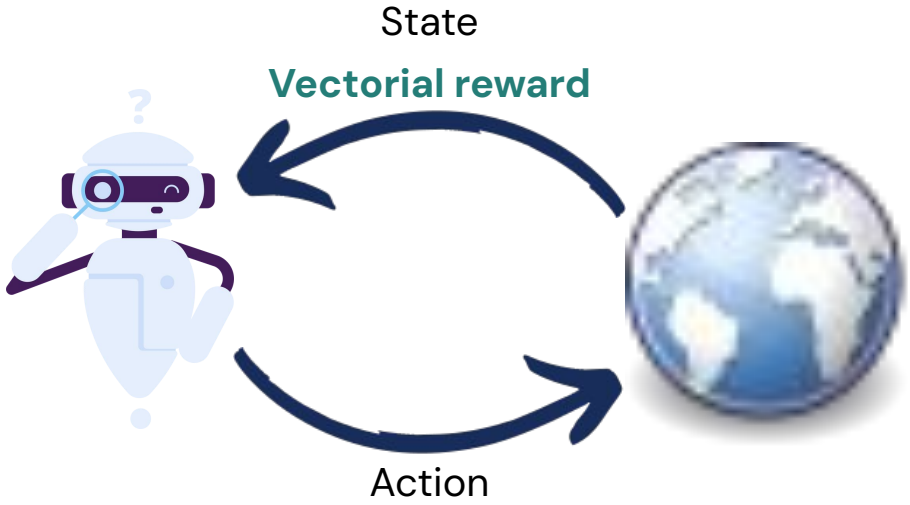
Scalar Reward Design Process

- **Design/tweak scalar reward function**
- **(Re-)Train RL agent using new/updated reward function (may take hours or days)**
- **Evaluate the outcome (and try to figure out what went wrong!)**

- GT Sophy - Super Human Racing AI Agent, Sony AI
- Objectives: high precision race car control, efficient racing tactics and manoeuvres, while respecting an imprecisely defined racing etiquette
- With enough time and computation, good results can be achieved

Multi-Objective Reinforcement Learning

- Vector-valued reward function
  - r = [r_objective1, r_objective2, ...]
- Length of the reward vector = number of objectives



Multi-Objective Reinforcement Learning

- Multi-Objective MDP $< S, A, T, \gamma, \mathbf{R} >$
  - Set of states
  - Set of actions
  - A vectorial reward function $\mathbf{R}: S \times A \times S \rightarrow \mathbb{R}^d$

    $d \geq 2$ objectives
  - Transition function (dynamics of the environment)
  - Discount factor $\gamma \in [0, 1]$

Value Functions and Policies

The agent behaves according to a policy:

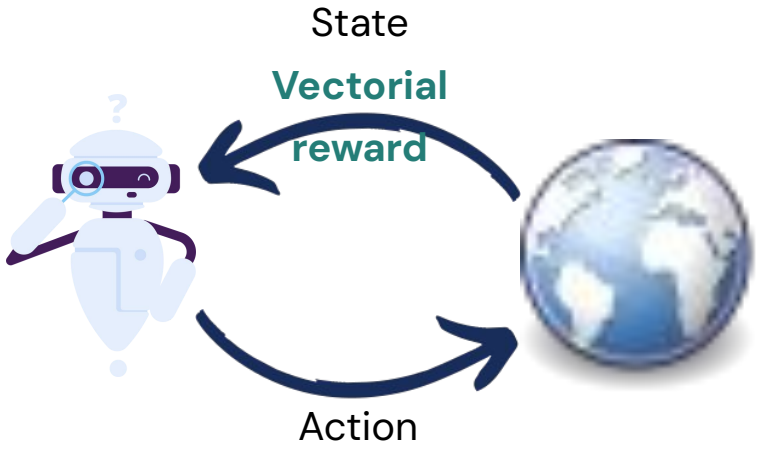
$$\pi : S \times A \to [0, 1]$$

The value function of a policy in a MOMDP:

$$\mathbf{V}^{\pi} = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \mathbf{r}_{k+1} \mid \pi, \mu\right]$$

where

$$\mathbf{r}_{k+1} = \mathbf{R}(s_k, a_k, s_{k+1})$$



Value Functions and Policies

- Vectorial value functions now supply only a partial ordering, even for a given state:

$$V_i^{\pi}(s) > V_i^{\pi'}(s) \text{ but } V_j^{\pi}(s) < V_j^{\pi'}(s)$$

- We can no longer determine which values are optimal without additional information about how to prioritize the objectives

Utility Functions

A utility function, u, is used to represent a user's preferences over objectives

Utility function maps a vector reward to a scalar utility:

$$u : \mathbb{R}^d \rightarrow \mathbb{R}$$

u is generally assumed to be monotonically increasing:

$$\left(\forall o, V_o^{\pi} > V_o^{\pi'}\right) \implies u\left(\mathbf{V}^{\pi}\right) \geq u(\mathbf{V}^{\pi'})$$

Utility Functions - Examples

- **Linear utility function**:

$$u(\mathbf{V}^{\pi}) = \mathbf{w}^{\top}\mathbf{V}^{\pi}$$

- Each element **w** specifies how much one unit of value for the corresponding objective contributes to the scalarised value
- The elements of the weight vector are all positive real numbers and sum to 1

Utility Functions - Examples

- The **product utility function** - seeks to make the objective values as balanced as possible

$$u(\mathbf{V}^{\pi}) = \prod_{o=1}^{d} V_o^{\pi}$$

Solution Sets

Solution Sets

- The utility function is often **unknown**
- In multi-objective settings there can now be multiple possibly optimal value vectors $\mathbf{V}$
- We need to reason about **sets of possibly optimal value vectors and policies** when thinking about solutions to MORL problems

Undominated Set

- The most general set of solutions: the undominated set

- The undominated set, U, is the subset of all possible policies Π for which there exists a possible utility function u with a maximal scalarised value:

$$U(\Pi) = \left\{ \pi \in \Pi \;\middle|\; \exists u, \forall \pi' \in \Pi : u(\mathbf{V}^{\pi}) \geq u(\mathbf{V}^{\pi'}) \right\}$$

Pareto Coverage Set

If the utility function u is any monotonically increasing function, then the **Pareto Coverage Set** (PCS) coincides with the undominated set:

$$PCS(\Pi) = \{\pi \in \Pi \mid \nexists \pi' \in \Pi : \mathbf{V}^{\pi'} \succ_P \mathbf{V}^{\pi}\}$$

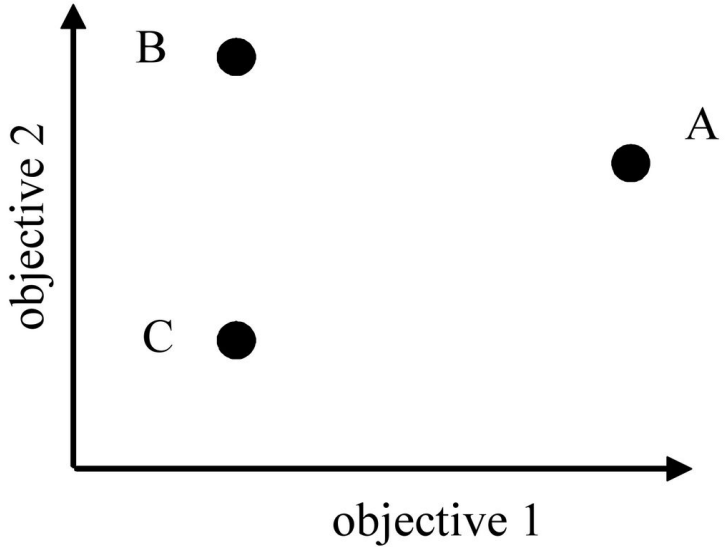
where $\succ_P$ is the Pareto dominance relationship:

$$\mathbf{V}^{\pi} \succ_P \mathbf{V}^{\pi'} \iff (\forall i : \mathbf{V}^{\pi}_i \geq \mathbf{V}^{\pi'}_i) \wedge (\exists i : \mathbf{V}^{\pi}_i > \mathbf{V}^{\pi'}_i)$$

Pareto Front

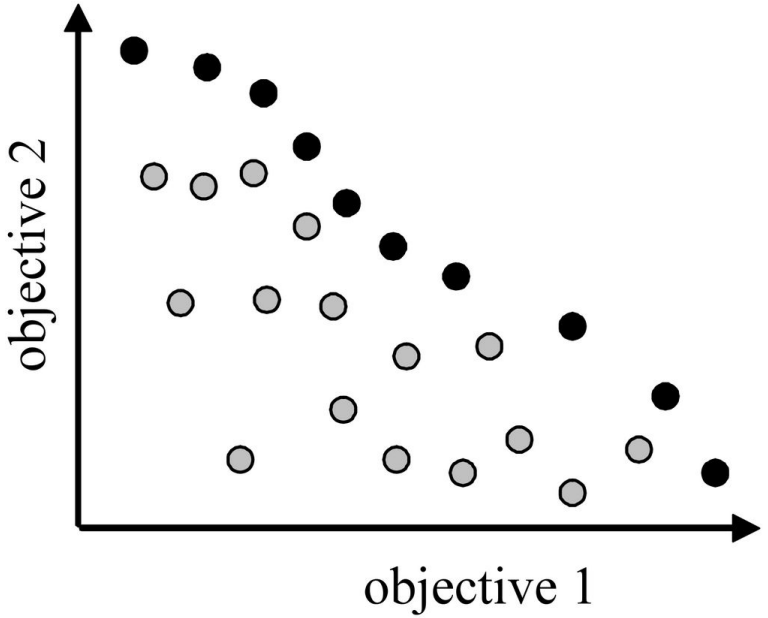
- The Pareto front (PF) is the set containing the value functions corresponding to the PCS policies

- Pareto dominance illustration, maximising objectives
- Solution A strongly dominates solution C
- Solution B weakly dominates solution C
- A and B are incomparable



Pareto Front

- Black points indicate solutions which form the Pareto front
- Grey solutions are dominated by at least one member of the Pareto front

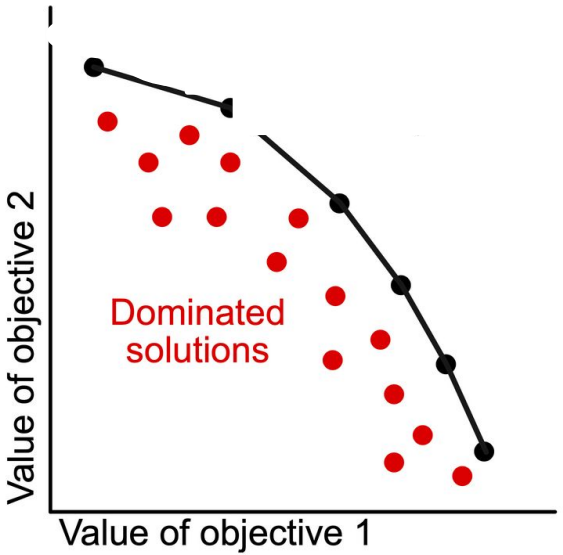


Convex Coverage Set (Convex Hull)

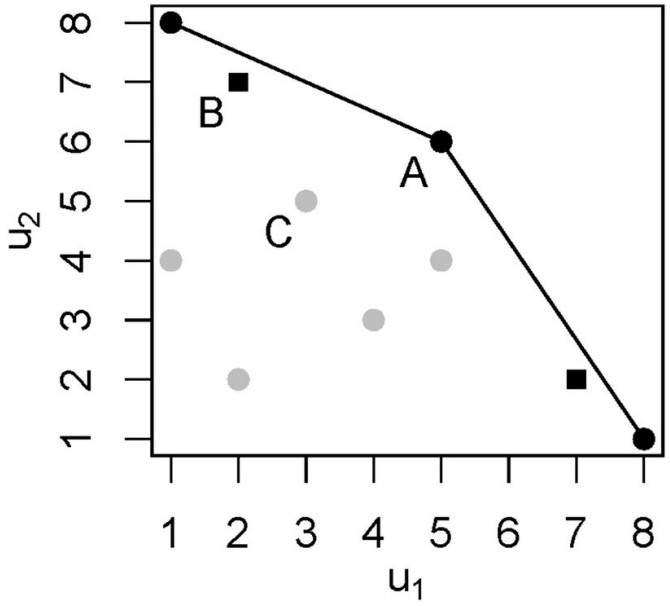
- The convex coverage set is the undominated set for non-decreasing linear utility functions

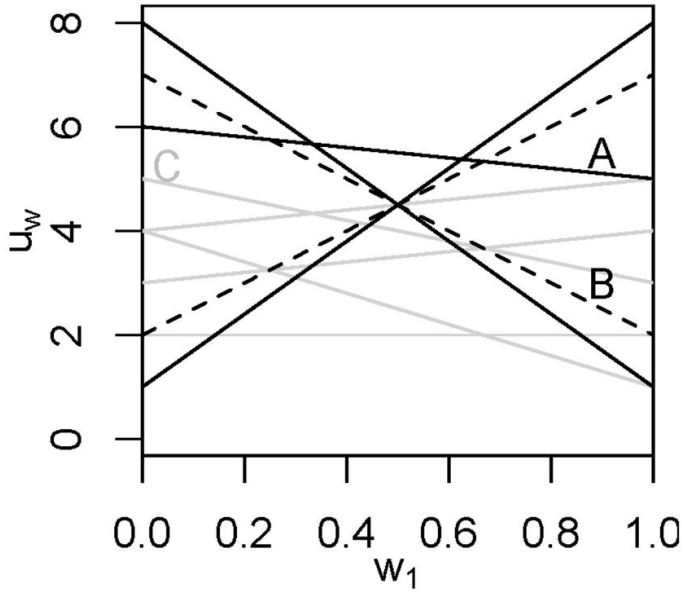
$$CCS(\Pi) = \{\pi \in \Pi \mid \exists \mathbf{w}, \forall \pi' \in \Pi : \mathbf{w}^\top \mathbf{V}^\pi \geq \mathbf{w}^\top \mathbf{V}^{\pi'}\}$$

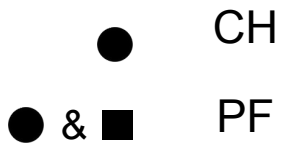
- The Convex Hull (CH) contains the value functions corresponding to the CCS policies



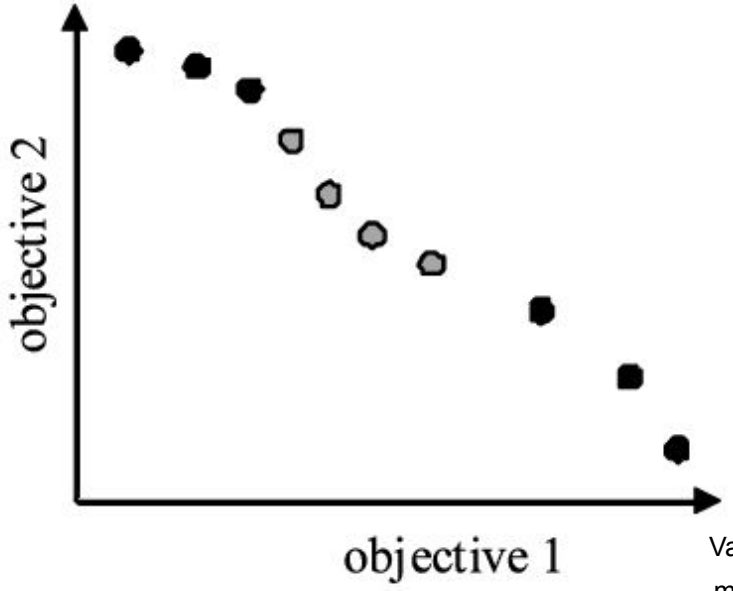
Convex Hull versus Pareto Front







Limitations of linear utilities



- Pareto front containing a concave region, indicated by the grey points
- Fundamental limitation of linear scalarisation: it cannot find policies which lie in non-convex regions of the Pareto front

Vamplew, P., Dazeley, R., Berry, A., Issabekov, R., & Dekker, E. (2011). Empirical evaluation methods for multiobjective reinforcement learning algorithms. Machine learning, 84, 51-80.

Axiomatic approach

- Defines the optimal solution set to be the Pareto front
- **Advantage**: retrieve a solution set containing an optimal policy for any possible monotonically increasing utility function, without any need to explicitly consider the details of those potential utility functions
- **Disadvantages**:
  - the set is typically large, and may be prohibitively expensive to retrieve
  - cannot exploit existing domain knowledge (e.g., in practical settings)

Utility-based approach

- Puts the user's utility at the center of the learning process
- Solution should be derived from utility (not axiomatically assumed)

- The properties of the user's utility function may drastically alter the desired solution, and what methods are available

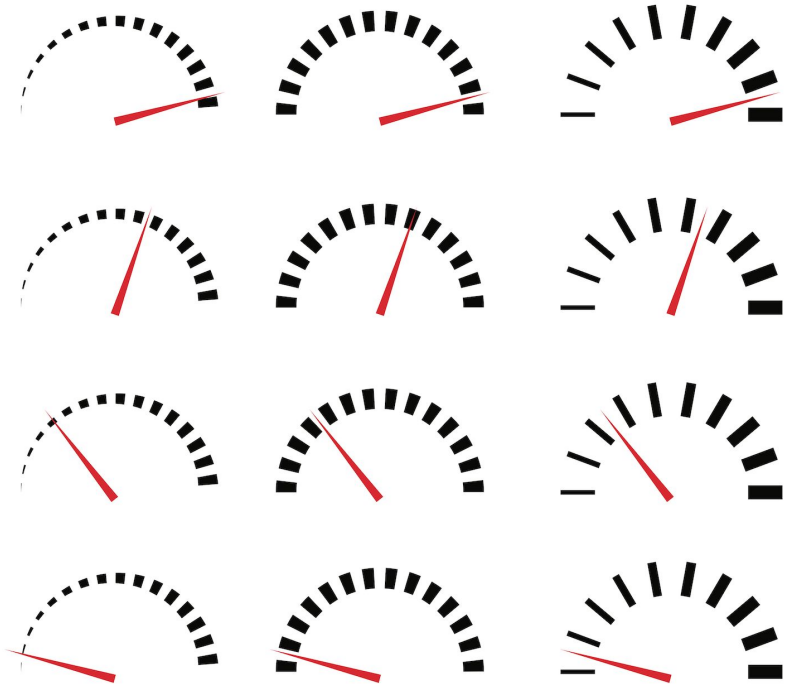
Utility-based approach

- Does not exclude or contradict the use of axiomatic methods
- When it is not possible to establish any constraints on the user's utility function, or other characteristics of the solution, prior to learning, we should still resort to axiomatic methods

Optimisation criteria

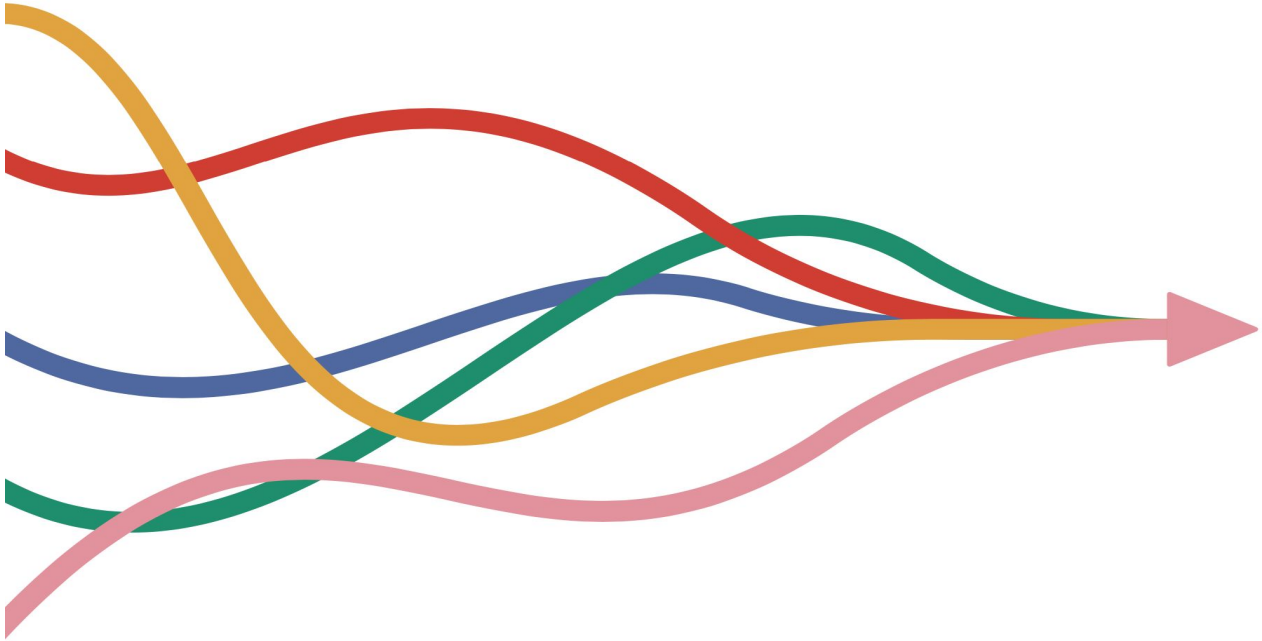
- Vectorial reward function
- Utility-based perspective

$$u_i : \mathbb{R}^d \rightarrow \mathbb{R}$$



Optimisation criteria

Optimisation criteria



Optimisation criteria

- Expected Scalarised Returns (ESR)
  - Calculate the expectation of the utility from the payoffs
  - Utility of an individual policy execution

Optimisation criteria

- Expected Scalarised Returns (ESR)
  - Calculate the expectation of the utility from the payoffs
  - Utility of an individual policy execution

- Scalarised Expected Returns (SER)
  - Calculate the utility of the expected payoff
  - Utility of the average payoff from several executions of the policy

Optimisation criteria

- Expected Scalarised Returns (ESR)

$$V_u^{\pi} = \mathbb{E}\left[u\left(\sum_{t=0}^{\infty}\gamma^t \mathbf{r}_t\right) \mid \pi, \mu_0\right]$$

- Scalarised Expected Returns (SER)

$$V_u^{\pi} = u\left(\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t \mathbf{r}_t \mid \pi, \mu_0\right]\right)$$

ESR

- The utility of a user is derived from a single execution of a policy
- Understudied in the RL literature
- ESR set is defined as the set of optimal solutions

Hayes, C. F., Verstraeten, T., Roijers, D. M., Howley, E., & Mannion, P. (2022). Expected scalarised returns dominance: a new solution concept for multi-objective decision making. Neural Computing and Applications, 1-21.

SER

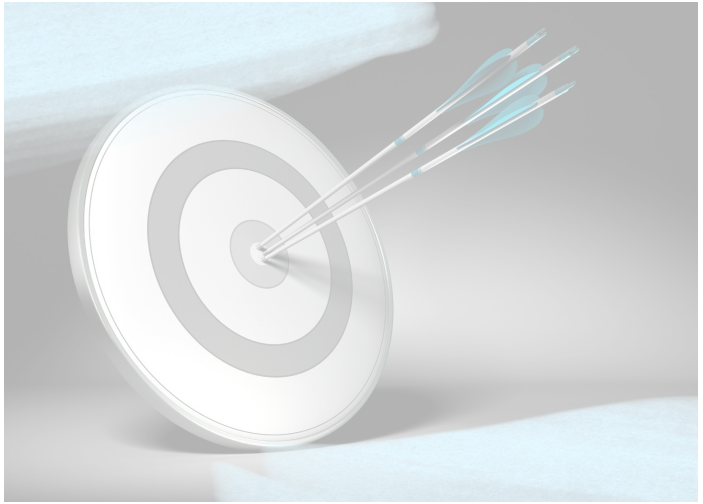
- The utility of a user is derived from multiple executions of a policy (i.e., user is concerned about achieving an optimal utility over multiple policy executions)
- Most commonly used optimisation criterion in multi-objective RL and planning
- Coverage set is defined as a set of optimal solutions for all possible utility functions

Optimisation criteria

- Note that:
- SER = ESR under linear scalarisation
- Which criterion should be chosen for optimisation depends on how the policies are used in practice

Part 2 - Multi-objective decision making in single-agent systems

Taxonomy, approaches, and tools



Taxonomy

- Categorizes problem classes based on the assumptions about the utility function, which policies the user allows, and whether one or multiple policies are required:
  - Single versus multiple policies
  - Linear versus monotonically increasing utility functions
  - Deterministic versus stochastic policies

Roijers, D. M., Vamplew, P., Whiteson, S., & Dazeley, R. (2013). A survey of multi-objective sequential decision-making. Journal of Artificial Intelligence Research, 48, 67-113.

Taxonomy

	single policy (known weights)		*multiple policies (unknown weights or decision support)*	
	deterministic	stochastic	deterministic	stochastic
linear scalarization	one deterministic stationary policy (1)		convex coverage set of deterministic stationary policies (2)	
monotonically increasing scalarization	one deterministic non-stationary policy (3)	one mixture policy of two or more deterministic stationary policies (4)	Pareto coverage set of deterministic non-stationary policies (5)	convex coverage set of deterministic stationary policies (6)

Roijers, D. M., Vamplew, P., Whiteson, S., & Dazeley, R. (2013). A survey of multi-objective sequential decision-making. Journal of Artificial Intelligence Research, 48, 67-113.

Deterministic versus non-stationary policies

Exercise:

White's Example (1982), 1 state, 3 actions

$$\mathbf{R}(a_1) = (3, 0), \mathbf{R}(a_2) = (0, 3), \mathbf{R}(a_3) = (1, 1)$$

1. Calculate the value functions under the corresponding deterministic policies: $\pi_i = a_i$

$$\mathbf{V}^{\pi_1} = (?, ?)$$
$$\mathbf{V}^{\pi_2} = (?, ?)$$
$$\mathbf{V}^{\pi_3} = (?, ?)$$

$$\mathbf{V}^{\pi} = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \mathbf{r}_{k+1} \mid \pi, \mu\right]$$

$$\sum_{t=0}^{\infty} \gamma^t r_{t+1} = \frac{r_{t+1}}{1-\gamma}$$

Deterministic versus non-stationary policies

Exercise:

White's Example (1982), 1 state, 3 actions

$$\mathbf{R}(a_1) = (3, 0), \mathbf{R}(a_2) = (0, 3), \mathbf{R}(a_3) = (1, 1)$$

1. Calculate the value functions under the corresponding deterministic policies: $\pi_i = a_i$

$$\mathbf{V}^{\pi_1} = (\frac{3}{1-\gamma}, 0)$$

$$\mathbf{V}^{\pi_2} = (0, \frac{3}{1-\gamma})$$

$$\mathbf{V}^{\pi_3} = (\frac{1}{1-\gamma}, \frac{1}{1-\gamma})$$

$$\mathbf{V}^{\pi} = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \mathbf{r}_{k+1} \mid \pi, \mu\right]$$

$$\sum_{t=0}^{\infty} \gamma^t r_{t+1} = \frac{r_{t+1}}{1-\gamma}$$

Deterministic versus non-stationary policies

Exercise:

White's Example (1982), 1 state, 3 actions

$$\mathbf{R}(a_1) = (3, 0), \mathbf{R}(a_2) = (0, 3), \mathbf{R}(a_3) = (1, 1)$$

2. Calculate the value function of a non-stationary policy alternating between a1 and a2, starting with a1:

$$\mathbf{V}^{\pi_{ns}} = (?, ?)$$

$$\mathbf{V}^{\pi} = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \mathbf{r}_{k+1} \mid \pi, \mu\right]$$

$$\sum_{t=0}^{\infty} \gamma^t r_{t+1} = \frac{r_{t+1}}{1-\gamma}$$

Deterministic versus non-stationary policies

Exercise:

White's Example (1982), 1 state, 3 actions

$$\mathbf{R}(a_1) = (3, 0), \mathbf{R}(a_2) = (0, 3), \mathbf{R}(a_3) = (1, 1)$$

2. Calculate the value function of a non-stationary policy alternating between a1 and a2, starting with a1:

$$\mathbf{V}^{\pi} = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \mathbf{r}_{k+1} \mid \pi, \mu\right]$$

$$\mathbf{V}^{\pi_{ns}} = \left(\frac{3}{1-\gamma^2}, \frac{3\gamma}{1-\gamma^2}\right)$$

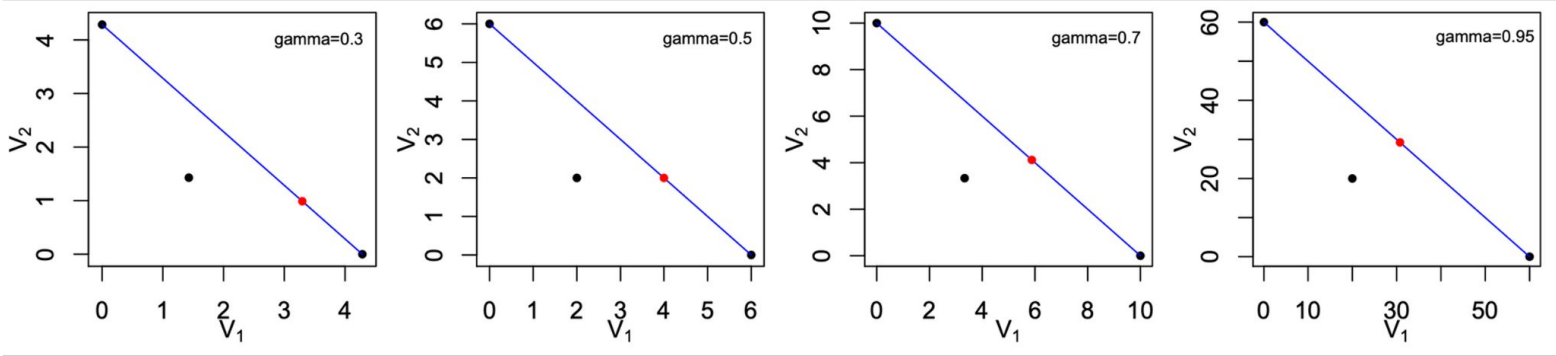
$$\sum_{t=0}^{\infty} \gamma^t r_{t+1} = \frac{r_{t+1}}{1-\gamma}$$

Deterministic versus non-stationary policies

Exercise:

$$\mathbf{V}^{\pi_1} = (\frac{3}{1-\gamma}, 0) \quad \mathbf{V}^{\pi_2} = (0, \frac{3}{1-\gamma}) \quad \mathbf{V}^{\pi_3} = (\frac{1}{1-\gamma}, \frac{1}{1-\gamma}) \quad \mathbf{V}^{\pi_{ns}} = (\frac{3}{1-\gamma^2}, \frac{3\gamma}{1-\gamma^2})$$

$$\pi_{ns} \succ_p \pi_3, \gamma \geq 0.5$$



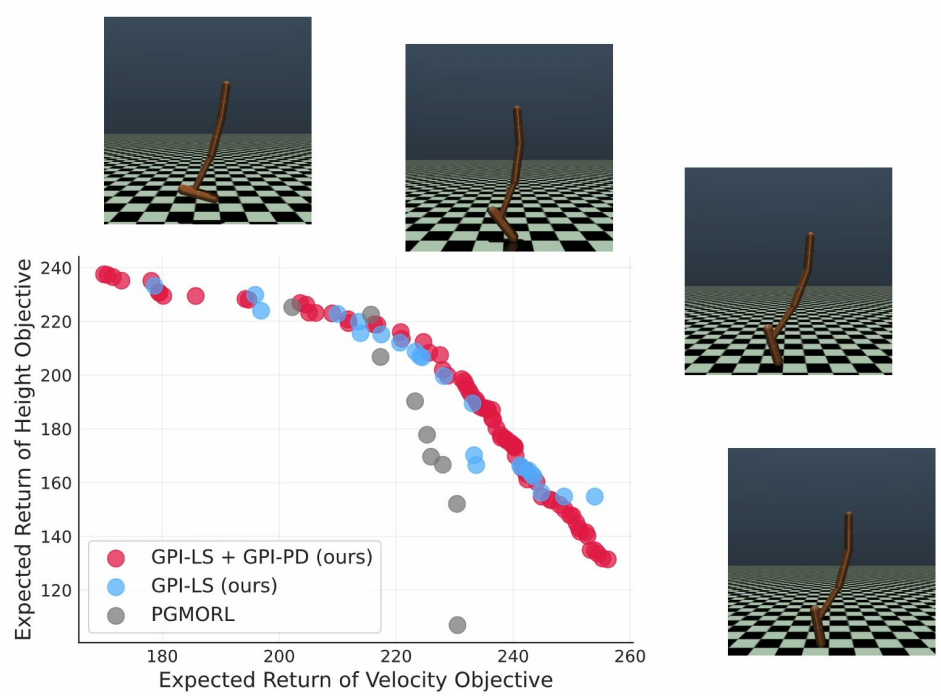
Whiteson & Roijers, Multi-Objective Decision Making, EASSS2016 tutorial

Inner Versus Outer Loop Methods

- In the **inner** loop approach, the required solution sets are produced by **changing the operators** required for optimization in single-objective problems (e.g., sums and maximisations), to operators that work on sets

- In **outer** loop approaches an (approximate) coverage set is built incrementally, by **solving scalarised instances** (i.e., re-use single-objective algorithms, but create a shell - the outer loop - around them)

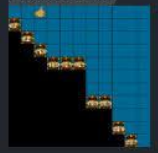
Benchmarks and Tools - MO-Gymnasium



Alegre et al. 2022. MO-Gym: A Library of Multi-Objective Reinforcement Learning Environments. In Proceedings of the 34th Benelux Conference on Artificial Intelligence BNAIC/Benelearn 2022.

Benchmarks and Tools - MO-Gymnasium

Environments

Env	Obs/Action spaces	Objectives	Description
`deep-sea-treasure-v0` 	Discrete / Discrete	`[treasure, time_penalty]`	Agent is a submarine that must collect a treasure while taking into account a time penalty. Treasures values taken from Yang et al. 2019.
`resource-gathering-v0` 	Discrete / Discrete	`[enemy, gold, gem]`	Agent must collect gold or gem. Enemies have a 10% chance of killing the agent. From Barret & Narayanan 2008.
`fruit-tree-v0` 	Discrete / Discrete	`[nutri1, ..., nutri6]`	Full binary tree of depth d=5,6 or 7. Every leaf contains a fruit with a value for the nutrients Protein, Carbs, Fats, Vitamins, Minerals and

Benchmarks and Tools - MORL-baselines

Implemented Algorithms

Name	Single/Multi-policy	ESR/SER	Observation space	Action space	Paper
GPI-LS + GPI-PD	Multi	SER	Continuous	Discrete / Continuous	Paper and Supplementary Materials
MORL/D	Multi	/	/	/	Paper
Envelope Q-Learning	Multi	SER	Continuous	Discrete	Paper
CAPQL	Multi	SER	Continuous	Continuous	Paper
PGMORL [1]	Multi	SER	Continuous	Continuous	Paper / Supplementary Materials
Pareto Conditioned Networks (PCN)	Multi	SER/ESR [2]	Continuous	Discrete / Continuous	Paper
Pareto Q-Learning	Multi	SER	Discrete	Discrete	Paper
MO Q learning	Single	SER	Discrete	Discrete	Paper
MPMOQLearning (outer loop MOQL)	Multi	SER	Discrete	Discrete	Paper
Optimistic Linear Support (OLS)	Multi	SER	/	/	Section 3.3 of the thesis

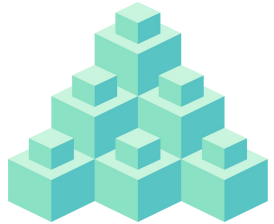
Felten, F., Alegre, L. N., Nowe, A., Bazzan, A., Talbi, E. G., Danoy, G., & C da Silva, B. (2024). A toolkit for reliable benchmarking and research in multi-objective reinforcement learning. Advances in Neural Information Processing Systems, 36.

Evaluation Metrics

- Axiomatic-based evaluation metrics
  - The hypervolume metric
  - The ε–metric
  - Cardinality
- Utility-based evaluation metrics
  - Expected utility metric (EUM)
  - Maximal utility loss (MUL)

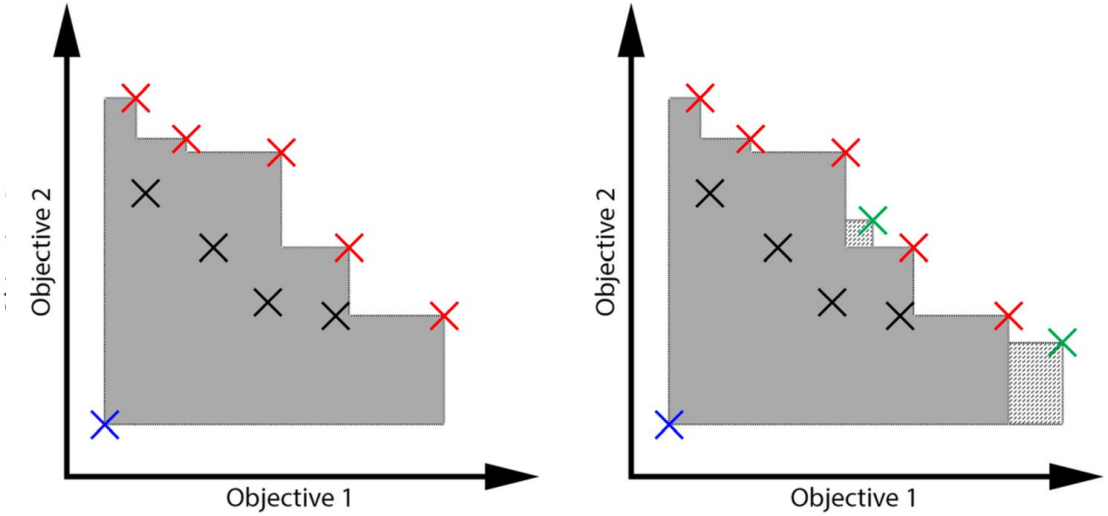
Hypervolume

- Measures the (hyper-)volume in value-space
  Pareto-dominated by the set of policies in an approximate coverage set



- Correlates with (but is not equal to) the spread of a set of undominated solutions over the possible multi-objective solution space

Hypervolume



- Left: The hypervolume for a 2-objective maximisation problem. Solutions in red form the undominated set, solutions in black are said to be dominated. The shaded area denotes the hypervolume of the undominated set with respect to the reference point (shown in blue).
- Right: The effect of adding two new points (shown in green) to the undominated set

Hypervolume

- Hypervolume values are difficult to interpret

- The benefit of a certain increase or decrease in hypervolume is not readily apparent to the end user:
  - adding a non-dominated solution at the extreme ends could lead to a large increase in the hypervolume, even if this additional solution is of little interest to the end user
  - adding a new solution close to other solutions in the non-dominated set can result in a minimal increase in hypervolume, even if the new solution is valuable to the end user

Expected utility metric

- Aims to directly evaluating an agent's ability to maximize user utility
- Defined as the expected utility for a user from this solution set, under some prior distribution over user utility functions
- Under the SER optimality criterion:

$$\mathrm{EUM} = \mathbb{E}_{P_u}\left[\max_{\pi \in \mathcal{S}} u(\mathbf{V}^{\pi})\right]$$

- Useful when we care about the agent's ability to do well across many different utility functions

Maximal utility loss

- Measures the maximal loss in utility that occurs when taking a policy from a given solution set, instead of the full set of possibly optimal solutions
- Under the SER optimality criterion:

$$\mathrm{MUL} = \max_{u \in \mathcal{U}} \left(\max_{\pi^* \in \mathcal{S}^*} u(\mathbf{V}^{\pi^*}) - \max_{\pi \in \mathcal{S}} u(\mathbf{V}^{\pi}) \right)$$

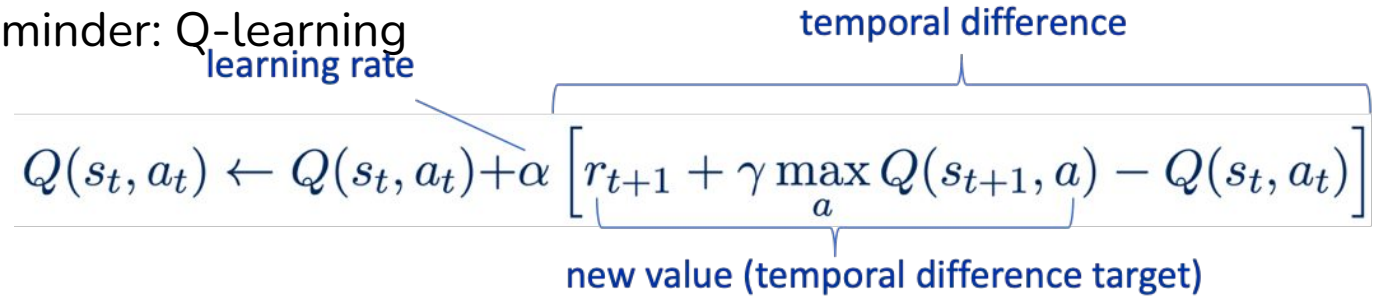
MORL approaches

Pareto Q-learning

- Van Moffaert, K., & Nowé, A. (2014)

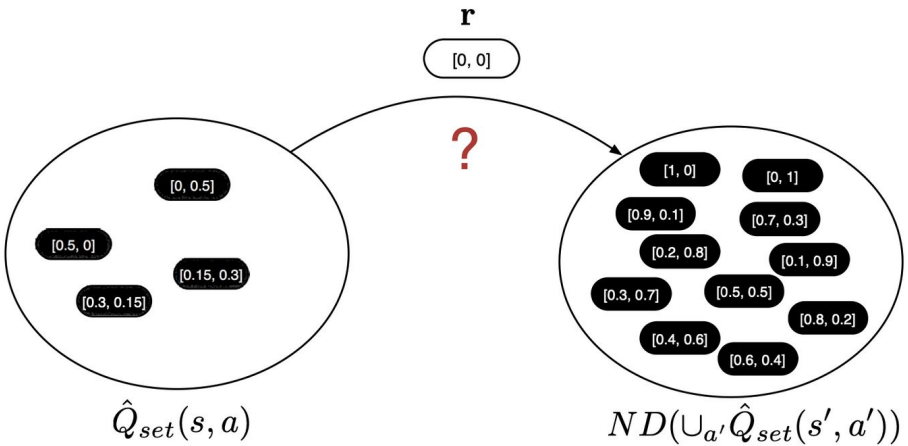
- Inner-loop method that modifies the workings of single-objective Q-learning to use set-operations and prune away dominated solutions

- Reminder: Q-learning



Pareto Q-learning

- We require set-based bootstrapping



Van Moffaert, K., & Nowé, A. (2014). Multi-objective reinforcement learning using sets of Pareto dominating policies. The Journal of Machine Learning Research, 15(1), 3483-3512.

Pareto Q-learning

- Proposed solution: learning immediate and future reward estimations separately

- The Q set of (s, a) can be calculated at run time by performing a vector-sum over the average immediate reward vector and the set of discounted Pareto dominating future rewards:

$$\hat{Q}_{set}(s,a) \leftarrow \overline{\mathfrak{R}}(s,a) \oplus \gamma ND_t(s,a)$$

Pareto Q-learning

Algorithm 4 Pareto Q-learning algorithm

1: Initialize $\hat{Q}_{set}(s,a)$'s as empty sets
2: **for** each episode t **do**
3: Initialize state s
4: **repeat**
5: Choose action a from s using a policy derived from the $\hat{Q}_{set}$'s
6: Take action a and observe state $s' \in S$ and reward vector $\mathbf{r} \in \mathbb{R}^m$
7: $\hat{Q}_{set}(s,a) \leftarrow \overline{\mathfrak{R}}(s,a) \oplus \gamma ND_t(s,a)$
8: $ND_t(s,a) \leftarrow ND(\cup_{a'} \hat{Q}_{set}(s',a'))$ ▷ Update ND policies of s' in s
9: $\overline{\mathfrak{R}}(s,a) \leftarrow \overline{\mathfrak{R}}(s,a) + \frac{\mathbf{r} - \overline{\mathfrak{R}}(s,a)}{n(s,a)}$ ▷ Update average immediate rewards
10: $s \leftarrow s'$ ▷ Proceed to next state
11: **until** s is terminal
12: **end for**

Pareto Q-learning

- How do we perform the action selection step?

Algorithm 5 Hypervolume Q_{set} evaluation

1: Retrieve current state s
2: evaluations $= \{\}$
3: **for** each action a **do**
4: $\quad hv_a \leftarrow HV(\hat{Q}_{set}(s,a))$
5: $\quad$ Append hv_a to evaluations ▷ Store hypervolume of the $\hat{Q}_{set}(s,a)$
6: **end for**
7: **return** evaluations

Pareto Deep Q-learning

Algorithm 1: PDQN

initialize replay-memory D
initialize reward estimator $\overline{R}$
initialize non-dominated estimator ND_t
initialize target non-dominated estimator $\hat{ND}_t = ND_t$
for *episode* = 1 **to** M **do**
  while *not terminal* **do**
    sample points p from $\mathbb{R}^{d-1}$
    $Q_{set}(s, ., p) \leftarrow \overline{R}(s, .) \oplus \gamma ND_t(s, ., p)$
    $hv \leftarrow$ hypervolume$(Q_{set}(s, ., p))$
    $a \leftarrow \epsilon -$ greedy(hv)
    execute a in environment, observe state s', reward r, terminal t
    add transition (s, a, r, s', t) to D
    sample minibatch $(s_i, a_i, r_i, s'_i, t_i)$ from D
    sample points p_i from $\mathbb{R}^{d-1}$
    $$y_i = \begin{cases} ND(\cup_{a' \in A} \hat{Q}_{set}(s'_i, a', p_i)) & \text{if not } t_i \\ r_i & \text{otherwise} \end{cases}$$
    update ND_t by performing gradient descent step on $(y_i - Q_{set}(s_i, a_i, p_i))^2$
    update $\overline{R}$ by performing gradient descent step on $(r_i - \overline{R}(s_i, a_i))^2$
    every C steps copy ND_t to $\hat{ND}_t$
  end
end

Reymond, M., & Nowé, A. (2019, May). Pareto-DQN: Approximating the Pareto front in complex multi-objective decision problems. In Proceedings of the adaptive and learning agents workshop (ALA-19) at AAMAS.

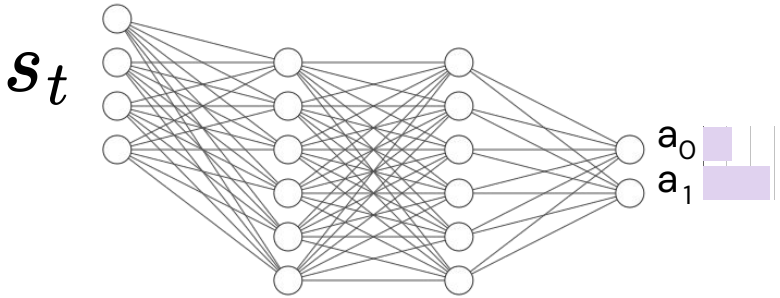
Pareto Conditioned Networks

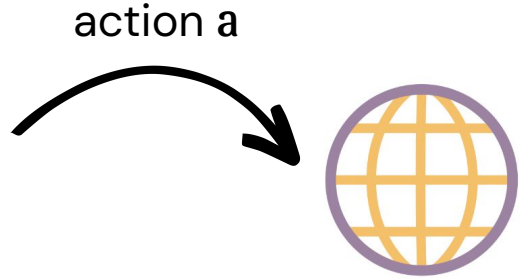
- Reymond, M., Bargiacchi, E., & Nowé, A. (2022, May). Pareto Conditioned Networks. In Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems(pp. 1110-1118).

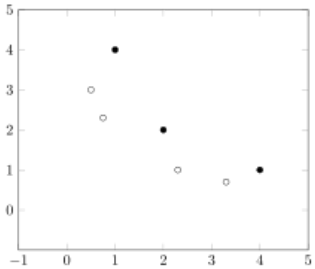
- Conditional networks for multi-policy embeddings

- The key idea is to use supervised learning techniques to improve the policy instead of resorting to temporal-difference learning

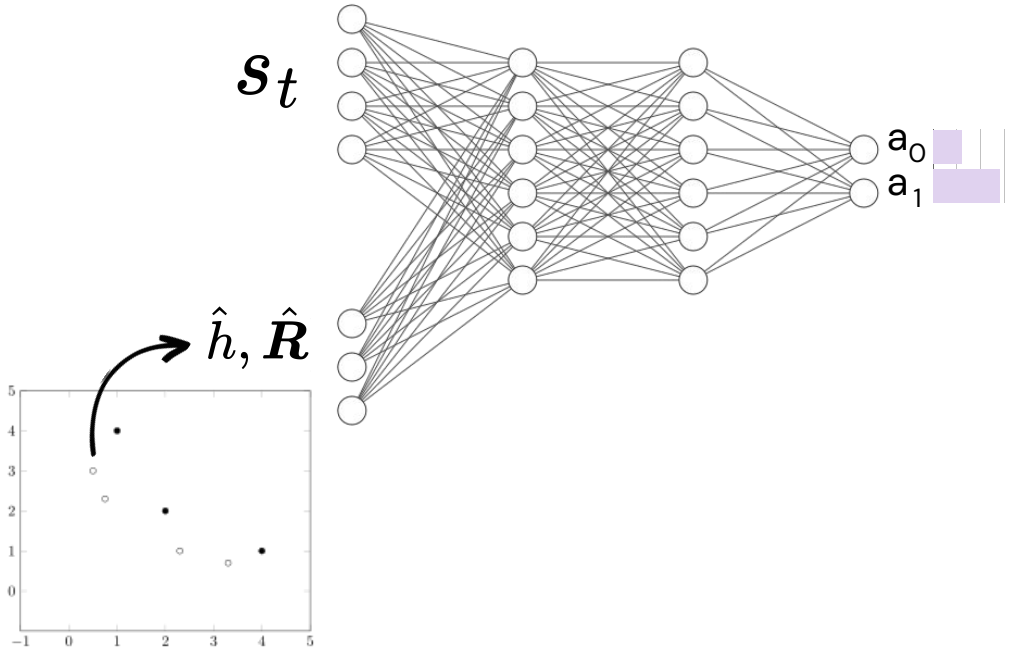
Pareto Conditioned Networks

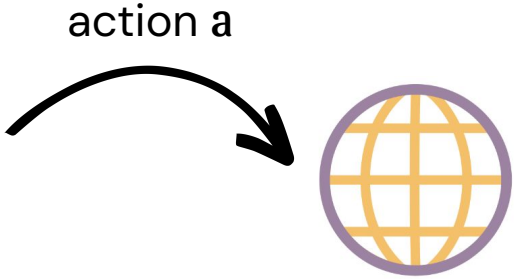




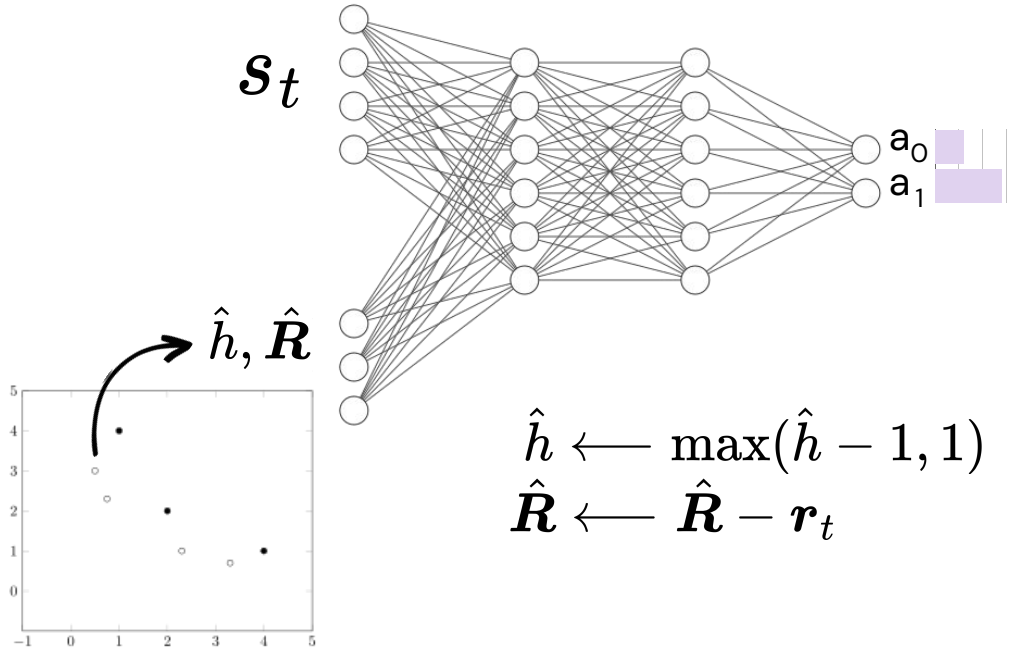


Pareto Conditioned Networks

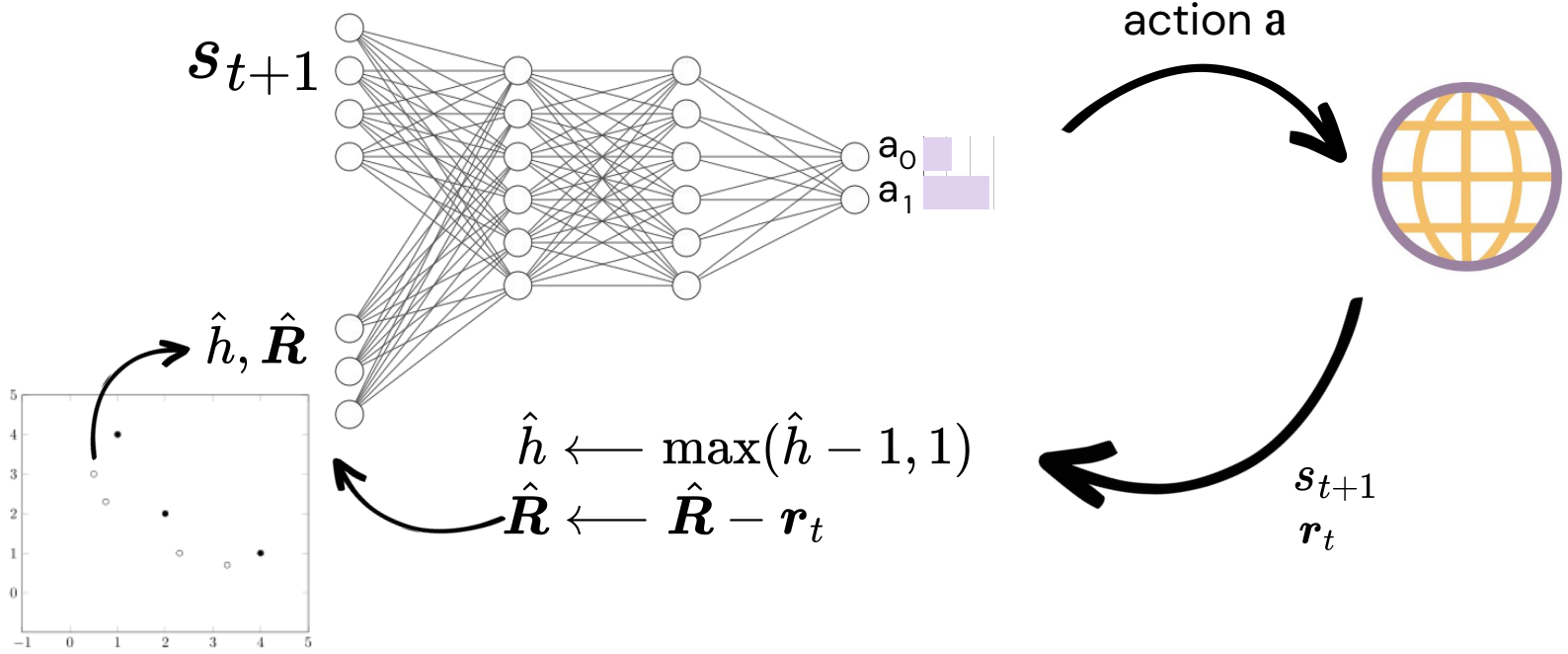




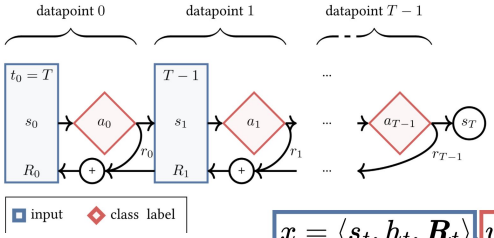
Pareto Conditioned Networks



Pareto Conditioned Networks



Pareto Conditioned Networks



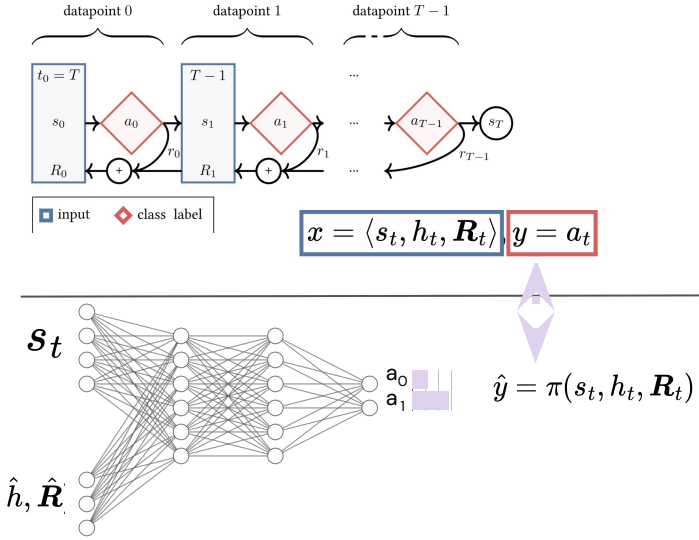
$x = \langle s_t, h_t, \boldsymbol{R}_t \rangle$ $y = a_t$

collect random trajectories

Require: PCN network π_θ with buffer $\mathcal{B}_{\text{PCN}}$.

for $n \in [N]$ **do**
  sample trajectory $\tau = \langle s_{0:T}, a_{0:T}, \boldsymbol{r}_{0:T} \rangle$ using random policy π_n
  for $\tau_i \in \tau$ **do**
    $\hat{h}, \hat{\boldsymbol{R}} = T - i, \sum_{t=i}^{T} \gamma^{t-i} \boldsymbol{r}_t$
    add $\tau_i, \hat{h}, \hat{\boldsymbol{R}}$ to $\mathcal{B}_{\text{PCN}}$
  end for
end for
while True **do**
  for $m \in [M]$ **do**
    update π_θ using $\mathcal{B}_{\text{PCN}}$
  end for
  prune $\mathcal{B}_{\text{PCN}}$ using $\{\boldsymbol{R}_\tau \mid \forall \tau \in \mathcal{B}_{\text{PCN}}\}$
  select $\hat{h}, \hat{\boldsymbol{R}}$ based on $\mathcal{B}_{\text{PCN}}$
  for $n \in [N]$ **do**
    sample $\tau = \langle s_{0:T}, a_{0:T}, \boldsymbol{r}_{0:T} \rangle$ using π_θ with $\hat{h}, \hat{\boldsymbol{R}}$
    for $\tau_i \in \tau$ **do**
      add $\tau_i, T - i, \boldsymbol{R}_i$ to $\mathcal{B}_{\text{PCN}}$
    end for
  end for
end while

Pareto Conditioned Networks

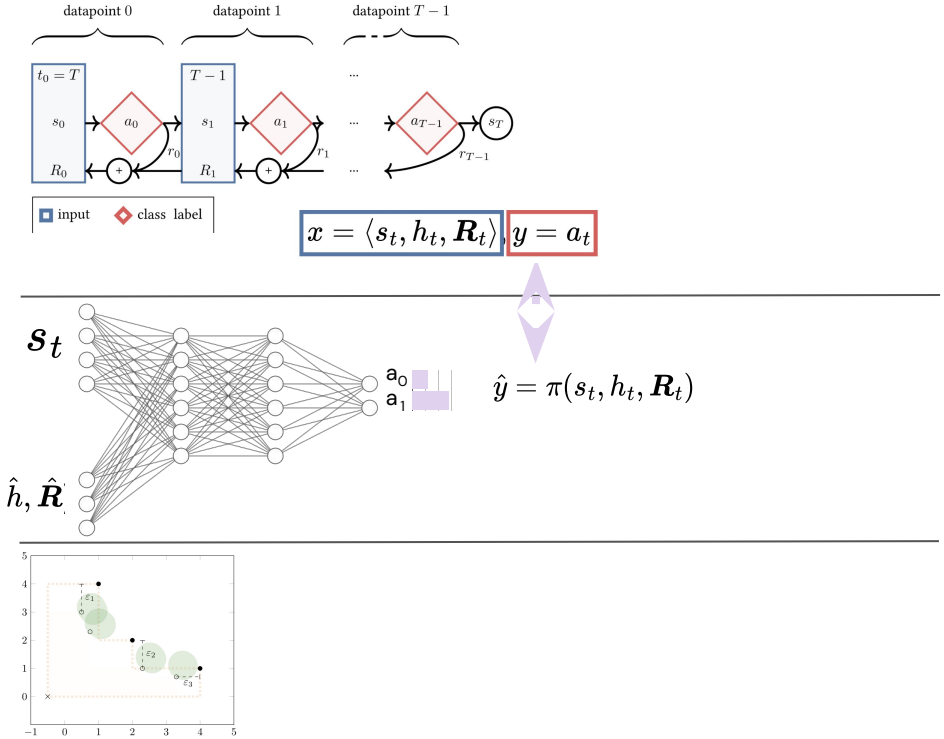


collect random trajectories
update network

Require: PCN network π_θ with buffer $\mathcal{B}_{\text{PCN}}$.

for $n \in [N]$ **do**
  sample trajectory $\tau = \langle s_{0:T}, a_{0:T}, \boldsymbol{r}_{0:T} \rangle$ using random policy π_n
  for $\tau_i \in \tau$ **do**
    $\hat{h}, \hat{\boldsymbol{R}} = T - i, \sum_{t=i}^{T} \gamma^{t-i} \boldsymbol{r}_t$
    add $\tau_i, \hat{h}, \hat{\boldsymbol{R}}$ to $\mathcal{B}_{\text{PCN}}$
  end for
end for
while True **do**
  for $m \in [M]$ **do**
    update π_θ using $\mathcal{B}_{\text{PCN}}$
  end for
  prune $\mathcal{B}_{\text{PCN}}$ using $\{\boldsymbol{R}_\tau \mid \forall \tau \in \mathcal{B}_{\text{PCN}}\}$
  select $\hat{h}, \hat{\boldsymbol{R}}$ based on $\mathcal{B}_{\text{PCN}}$
  for $n \in [N]$ **do**
    sample $\tau = \langle s_{0:T}, a_{0:T}, \boldsymbol{r}_{0:T} \rangle$ using π_θ with $\hat{h}, \hat{\boldsymbol{R}}$
    for $\tau_i \in \tau$ **do**
      add $\tau_i, T - i, \boldsymbol{R}_i$ to $\mathcal{B}_{\text{PCN}}$
    end for
  end for
end while

Pareto Conditioned Networks

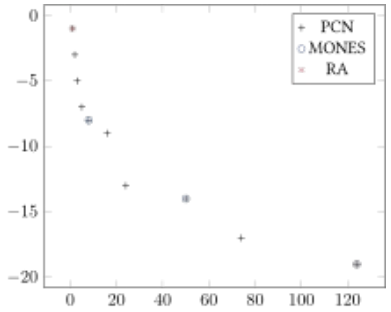


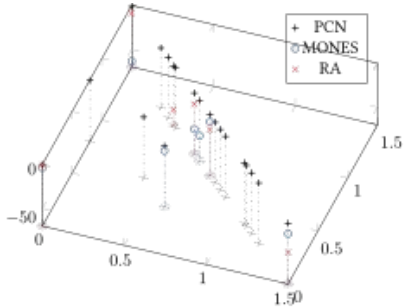
collect random trajectories
update network
explore on optimistic returns

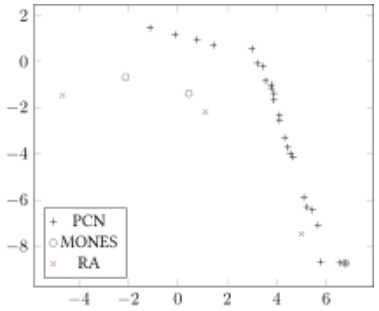
Require: PCN network π_θ with buffer $\mathcal{B}_{\text{PCN}}$.

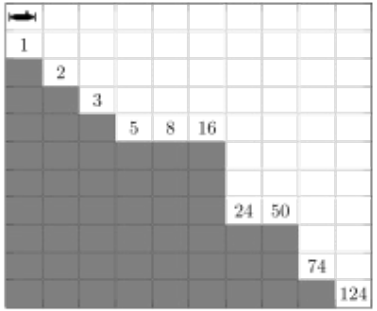
for $n \in [N]$ **do**
  sample trajectory $\tau = \langle s_{0:T}, a_{0:T}, \boldsymbol{r}_{0:T} \rangle$ using random policy π_n
  for $\tau_i \in \tau$ **do**
    $\hat{h}, \hat{\boldsymbol{R}} = T - i, \sum_{t=i}^{T} \gamma^{t-i} \boldsymbol{r}_t$
    add $\tau_i, \hat{h}, \hat{\boldsymbol{R}}$ to $\mathcal{B}_{\text{PCN}}$
  end for
end for
while True **do**
  for $m \in [M]$ **do**
    update π_θ using $\mathcal{B}_{\text{PCN}}$
  end for
  prune $\mathcal{B}_{\text{PCN}}$ using $\{\boldsymbol{R}_\tau \mid \forall \tau \in \mathcal{B}_{\text{PCN}}\}$
  select $h, \boldsymbol{R}$ based on $\mathcal{B}_{\text{PCN}}$
  for $n \in [N]$ **do**
    sample $\tau = \langle s_{0:T}, a_{0:T}, \boldsymbol{r}_{0:T} \rangle$ using π_θ with $\hat{h}, \hat{\boldsymbol{R}}$
    for $\tau_i \in \tau$ **do**
      add $\tau_i, T - i, \boldsymbol{R}_i$ to $\mathcal{B}_{\text{PCN}}$
    end for
  end for
end while

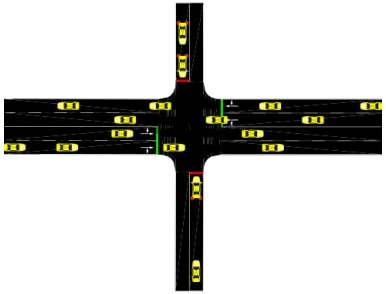
Pareto Conditioned Networks











GPI Linear Support

- Generalized Policy Improvement (GPI)
  - is the computation of a policy π' that improves over a set of policies $\pi \in \Pi$ given any new reward weights w

$$\pi^{GPI}(s;w) = \arg\max_{a \in \mathcal{A}} \max_{\pi \in \Pi} q_w^{\pi}(s,a)$$

- GPI Theorem (Barreto et al., 2017):

$$q_w^{GPI}(s,a) \geq \max_{\pi \in \Pi} q_w^{\pi}(s,a) \quad \textit{for any } w \in \mathcal{W}$$

GPI Linear Support

- Key idea: GPI Prioritization
  - Identifying the most promising preferences to train on

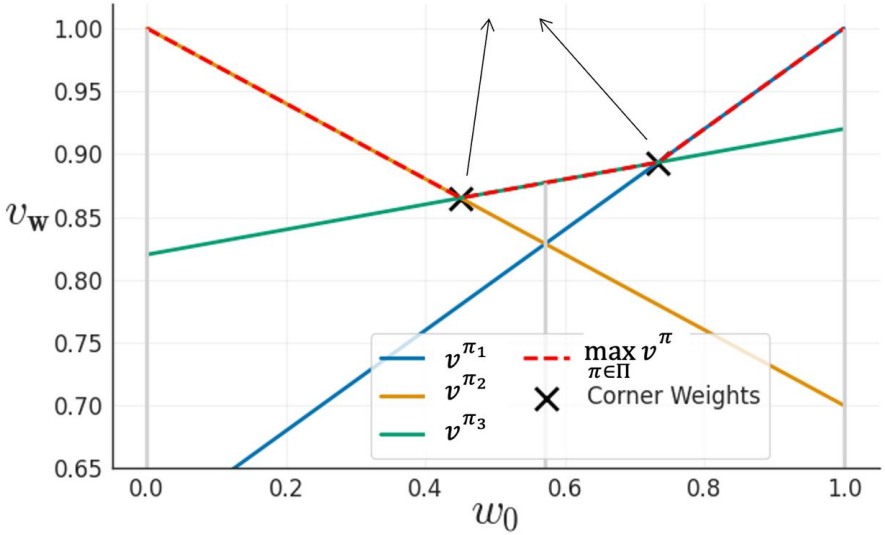
    ↦ focus on corner weights

  - Prioritize reward weights w.r.t. performance improvement given by GPI:

$$\arg\max_{\mathbf{w}\in\mathcal{W}_{\text{corner}}} \left(v_{\mathbf{w}}^{\text{GPI}} - \max_{\pi\in\Pi} v_{\mathbf{w}}^{\pi}\right)$$

GPI Linear Support

$$\arg\max_{\mathbf{w}\in\mathcal{W}_{\text{corner}}} \left(v^{\text{GPI}}_{\mathbf{w}} - \max_{\pi\in\Pi} v^{\pi}_{\mathbf{w}}\right)$$



- Maximum improvement is guaranteed to be in one of the corner weights
- Iteratively:
  - Selects the corner weight with a higher GPI priority
  - Learns an improved policy for the selected reward weights

GPI Linear Support

Algorithm 1: GPI Linear Support (GPI-LS)

Input: MOMDP M

$\pi_{\mathbf{w}}, \mathbf{v}^{\pi_{\mathbf{w}}} \leftarrow \text{NewPolicy}(\mathbf{w} = [1, 0, ..., 0]^\top)$

$\Pi \leftarrow \{\pi_{\mathbf{w}}\}, \mathcal{V} \leftarrow \{\mathbf{v}^{\pi_{\mathbf{w}}}\}, \mathcal{M} \leftarrow \{\}$

while *True* **do**

  $\mathcal{W}_{\text{corner}} \leftarrow \text{CornerWeights}(\mathcal{V}) \setminus \mathcal{M}$

  if $\mathcal{W}_{\text{corner}}$ *is empty* **then**

    return $\Pi, \mathcal{V}$; ▷ Found CCS (or ϵ-CCS)

  $\mathbf{w} \leftarrow \arg\max_{\mathbf{w} \in \mathcal{W}_{\text{corner}}} (v_{\mathbf{w}}^{\text{GPI}} - \max_{\pi \in \Pi} v_{\mathbf{w}}^{\pi})$

  $\pi_{\mathbf{w}}, \mathbf{v}^{\pi_{\mathbf{w}}}, \text{done} \leftarrow \text{NewPolicy}(\mathbf{w}, \Pi)$

  if done **then**

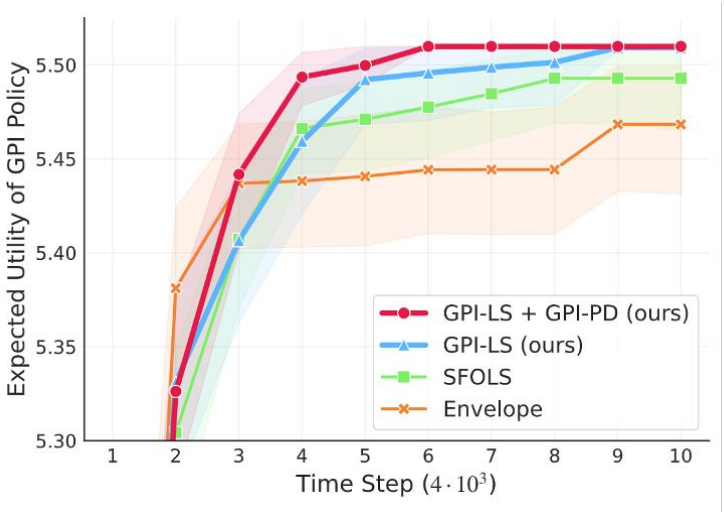
    Add $\mathbf{w}$ to $\mathcal{M}$; ▷ Adds $\mathbf{w}$ to support of partial CCS

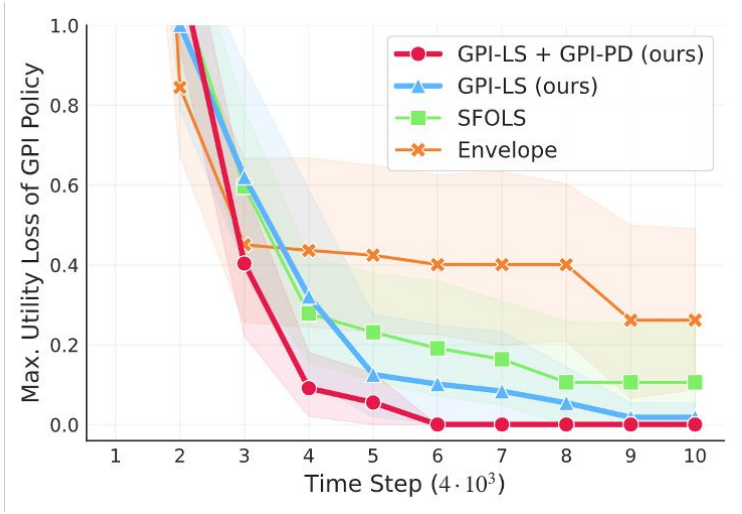
  Add $\pi_{\mathbf{w}}$ to Π and $\mathbf{v}^{\pi_{\mathbf{w}}}$ to $\mathcal{V}$

  $\Pi, \mathcal{V} \leftarrow \text{RemoveDominated}(\Pi, \mathcal{V})$

GPI Linear Support

Deep Sea treasure





Other approaches

- Källström, J., & Heintz, F. (2023, May). Model-Based Actor-Critic for Multi-Objective Reinforcement Learning with Dynamic Utility Functions. In Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems (pp. 2818-2820).
- Abels A, Roijers DM, Lenaerts T, Nowe A, Steckelmacher D. Dynamic weights in multi-objective deep reinforcement learning. In 36th International Conference on Machine Learning, ICML 2019 . Vol. 97. International Machine Learning Society (IMLS). 2019. p. 11-20. (36th International Conference on Machine Learning, ICML 2019).
- Hayes, C. F., Reymond, M., Roijers, D. M., Howley, E., & Mannion, P. (2023). Monte Carlo tree search algorithms for risk-aware and multi-objective reinforcement learning. Autonomous Agents and Multi-Agent Systems, 37(2), 26.
- Reymond, M., Hayes, C. F., Steckelmacher, D., Roijers, D. M., & Nowé, A. (2023). Actor-critic multi-objective reinforcement learning for non-linear utility functions. Autonomous Agents and Multi-Agent Systems, 37(2), 23.
- Röpke, W., Hayes, C. F., Mannion, P., Howley, E., Nowé, A., & Roijers, D. M. (2023). Distributional Multi-Objective Decision Making. IJCAI 2023.

Connections to other problems - POMDPs

- If one assumes **linear utility** functions, **POMDPs are a superclass of MOMDPs**
- Intuitively, imagine there would be a "true objective" and the linear weights of the utility function would form a "belief" over what the true objective would be
- MOMDPs and POMDPs have different interpretations
- Theoretical results and methods from POMDPs can be transferred/adapted for MOMDPs (e.g., Optimistic Linear Support - approximate the CCS)

Multi-objective as multi-agent problems

- **Objectives are not agents**
  - Cast single-agent multi-objective problems as multi-agent problems, with each agent representing a competing objective
  - Either through voting rules or Nash equilibria a policy is selected

Q: What do you think is the potential issue with such approaches?

Multi-objective as multi-agent problems

- Such mechanisms offer no guarantees with respect to the user's utility and it is unclear if these "compromise solutions" represents desired trade-offs or not

- Note that objectives can be more or less important and may have non-linear interactions in the utility function

- This is different than determining trade-offs between the individual payoffs of agents

Multi-objective as multi-agent problems

- But **altruistic agents can see other agents as objectives**
  - An altruistic agent could see the utility of other agents as objectives
  - Modelling other agents as objectives enables explicitly imposing fairness between these objectives, i.e., the utilities of the agents

- Lorenz dominance ordering - a refinement of Pareto dominance, adding a predilection towards a more balanced distribution of values over the objectives

Human-aligned agents

- How to ensure that the decisions and behaviour of autonomous agents are **safe, trustworthy, aligned, interpretable, fair, and unbiased**

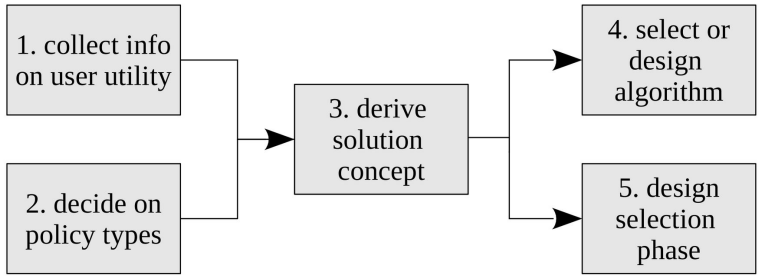
- Since these are additional considerations beyond maximising the agent's primary reward, there is a clear link to multi-objective approaches

Modelling and dealing w/

Multiple objectives

How to deal with MO problems

- Collect available information about user utility.
- Decide which policies (e.g., stochastic vs deterministic) are allowed.
- Derive the optimal solution concept from the resulting information of the first two points.
- Select or design an algorithm that fits the solution concept.
- When multiple policies are required for the solution, design a method for the user to select the desired policy among these optimal policies.



Part 3 - Multi-objective decision making in multi-agent systems

Core concepts, taxonomy, and results

@aibrussels
@rox_teo

Going to the conference

Two players

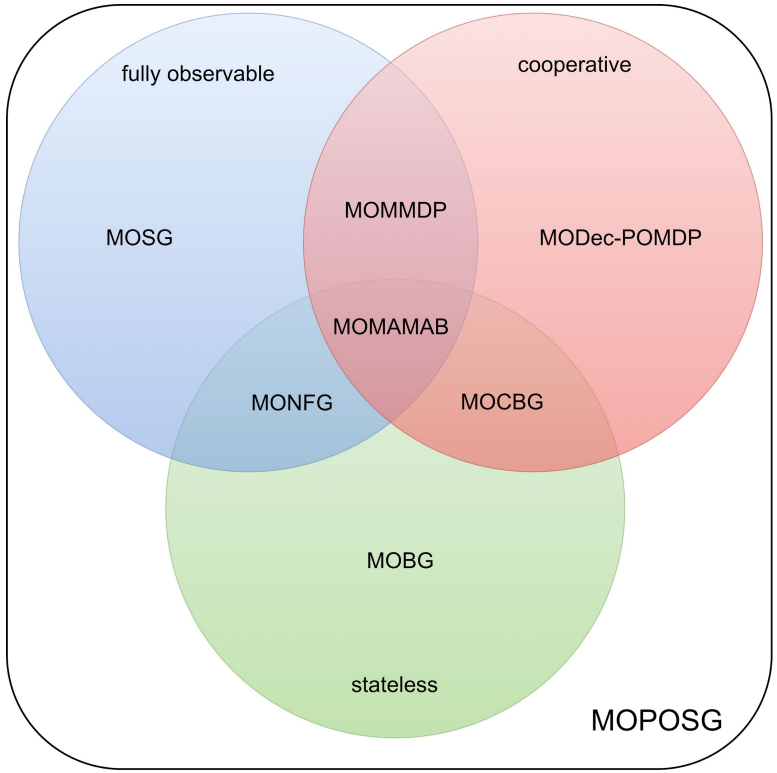
- rewards are public
- utility is private

MONFG

Why hard?

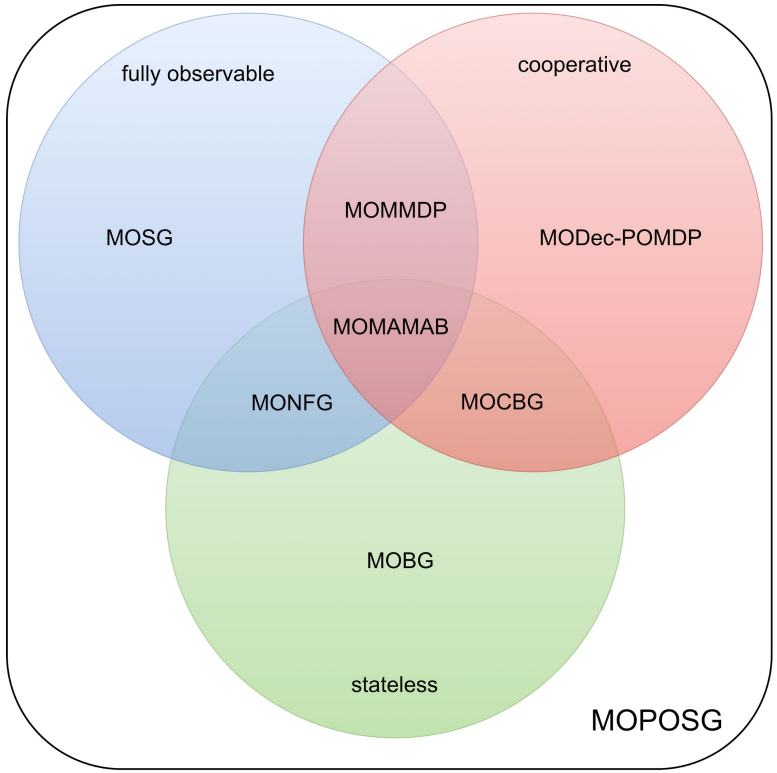
	Taxi	Tram	Walking
Taxi	(10€, 5min); (10€, 5min)	(20€, 5min); (2€, 15min)	(20€, 5min); (0€, 35min)
Tram	(2€, 15min); (20€, 5min)	(2€, 15min); (2€, 15min)	(2€, 15min); (0€, 35min)
Walking	(0€, 35min); (20€, 5min)	(0€, 35min); (2€, 15min)	(0€, 35min); (0€, 35min)

MOPOSG



Models:
On the basis of rewards (in objectives) and observations (about states).

MOPOSG



Models:
On the basis of rewards (in objectives) and observations (about states).

But utility is not yet modelled!

Because life really is not simple

- What are your objectives for your current research project?
  - Publishing asap?
  - Quality of conference/journal?
  - Collaboration potential?
  - Flag-posting?
  - Increasing funding potential?
  - Finishing your PhD?
- How about your co-authors?

Life is still not simple

- What are your objectives for your current research project?
  - Publishing asap?
  - Quality of conference/journal?
  - Collaboration potential?
  - Flag-posting?
  - Increasing funding potential?
  - Finishing your PhD?
- Setting?

Life is still not simple at all?

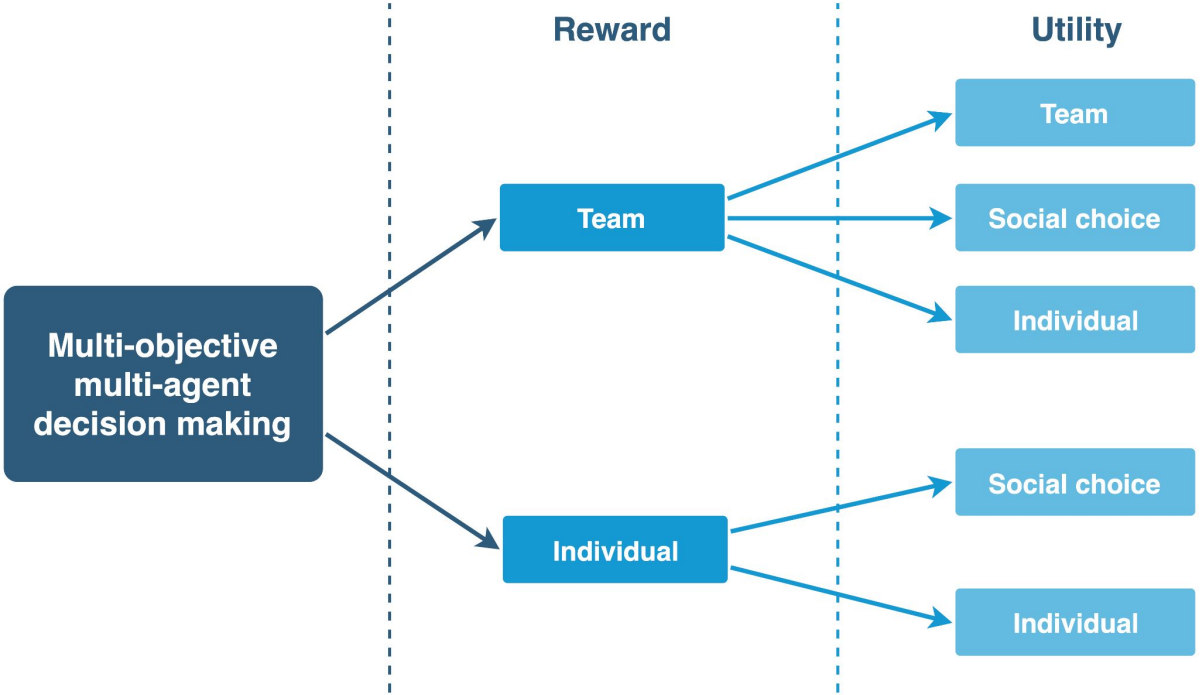
- What are your objectives for your current research project?
  - Publishing asap?
  - Quality of conference/journal?
  - Collaboration potential?
  - Flag-posting?
  - Increasing funding potential?
  - Finishing your PhD?
- Truly cooperative though?

Short break

Structuring the MOMADM field

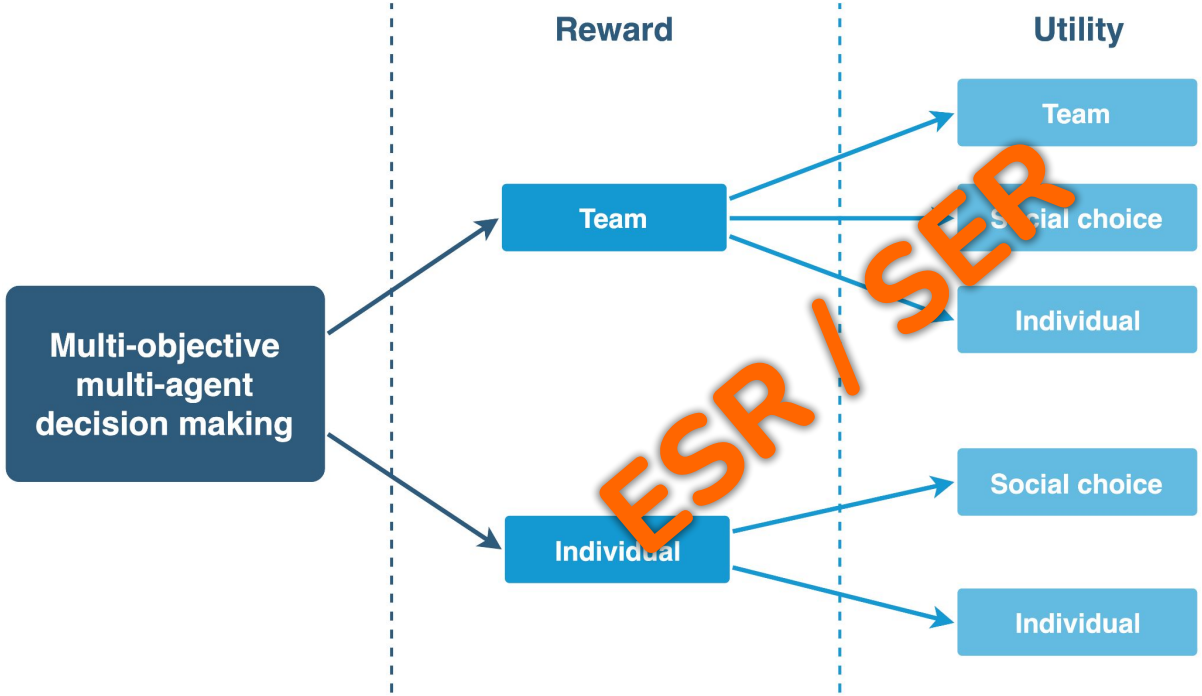
Taxonomy and solution concepts

Taxonomy



Rădulescu, R., Mannion, P., Roijers, D. M., & Nowé, A. (2020). Multi-objective multi-agent decision making: a utility-based analysis and survey. *Autonomous Agents and Multi-Agent Systems*, *34*(1), 1-52.

Taxonomy


Reward
Utility
Multi-objective multi-agent decision making
Team
Individual
Team
Social choice
Individual
Social choice
Individual
ESR / SER

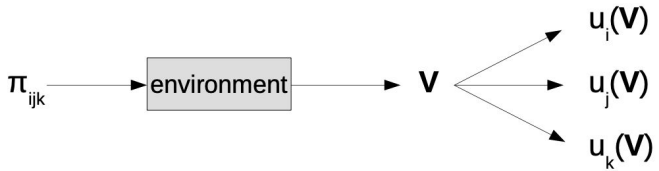
Examples - Team Reward

- Team utility
  - a company that aims to be environmentally responsible, while maximising profits

Examples - Team Reward

- Team utility

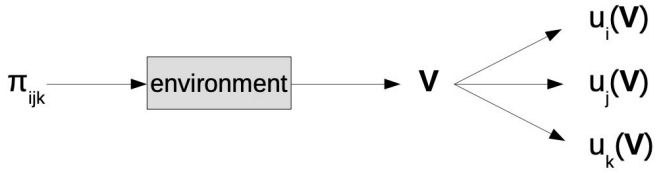
- Individual utility
  - Climate change policies, resource management



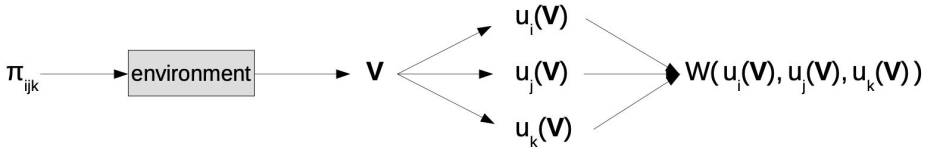
Examples - Team Reward

- Team utility

- Individual utility

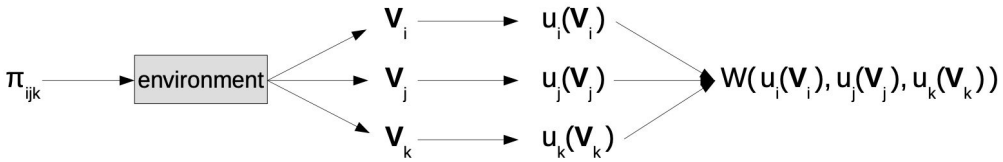


- Social Choice
  - urban planning/environmental management/social welfare policies



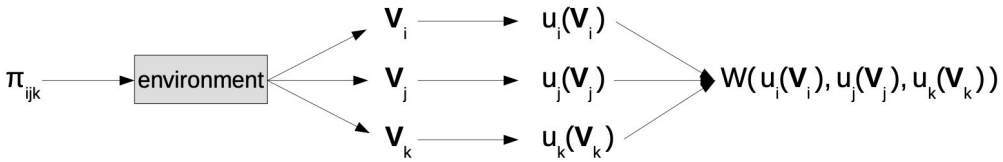
Examples - Individual Reward

- Social choice
  - Public tenders

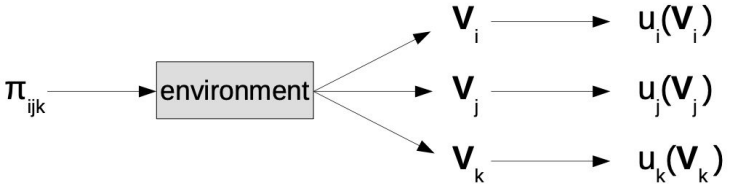


Examples - Individual Reward

- Social choice
  - Public tenders



- Individual utility
  - participating in city traffic, work commutes



Solution concepts

Reward \ Utility	Team	Social choice	Individual
Team	Coverage sets	Mechanism design	Coverage sets (+ Negotiation) Equilibria and stability concepts
Individual		Mechanism design	Equilibria and stability concepts Coverage Sets as best responses

Coverage sets

- Contain at least one optimal policy for each possible utility function

Reward \ Utility	Team	Social choice	Individual
Team	Coverage sets	Mechanism design	Coverage sets (+ Negotiation) Equilibria and stability concepts
Individual		Mechanism design	Equilibria and stability concepts Coverage Sets as best responses

- **TRTU**: rewards and derived utility is shared between agents, with one utility function selected during execution
- **TRIU**: agent can (contractually) agree which policy to execute
- **IRIU**: set of possible best responses to the behaviour of other agents

Coverage sets: negotiation

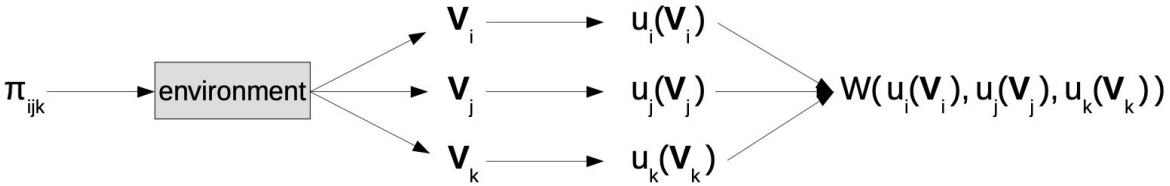
- Automated negotiation
  - Autonomous negotiating agents, representing their user's interests/preferences
  - Reach a compromise that satisfies all the involved parties
  - Pursue equity (i.e., fairness and justice)

- Baarslag, T., Kaisers, M., Gerding, E., Jonker, C. M., & Gratch, J. (2017). When will negotiation agents be able to represent us? The challenges and opportunities for autonomous negotiators. International Joint Conferences on Artificial Intelligence.
- Aydoğan, R., & Jonker, C. M. (2023). A Survey of Decision Support Mechanisms for Negotiation. In Recent Advances in Agent-Based Negotiation: Applications and Competition Challenges (pp. 30-51). Singapore: Springer Nature Singapore.

Social Welfare and Mechanism Design

- System perspective: what is a socially desirable outcome

Reward	Utility: Team	Utility: Social choice	Utility: Individual
Team	Coverage sets	Mechanism design	Coverage sets (+ Negotiation) Equilibria and stability concepts
Individual		Mechanism design	Equilibria and stability concepts Coverage Sets as best responses



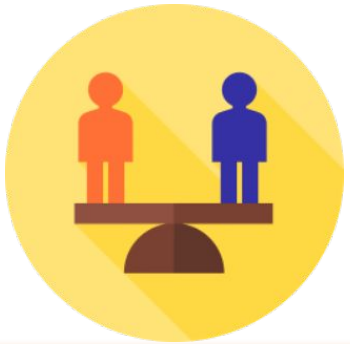
Design a system that forces agents to the truthful about their utilities and leads to optimal solution under W

Equilibria and stability concepts

- Stable outcomes from which self-interested agents have no incentive to deviate

Reward \ Utility	Team	Social choice	Individual
Team	Coverage sets	Mechanism design	Coverage sets (+ Negotiation) Equilibria and stability concepts
Individual		Mechanism design	Equilibria and stability concepts Coverage Sets as best responses

- Nash equilibria, correlated equilibria, cyclic equilibria, coalition formation



Nash Equilibrium

- No agent can improve their utility by unilaterally deviating from the joint strategy π^{NE}

- Nash equilibrium under SER:

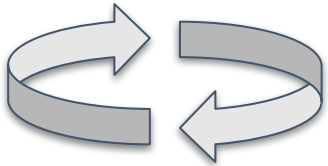
$$\mathbb{E}u_i\left[\mathbf{p}_i(\pi_i^{NE}, \pi_{-i}^{NE})\right] \geq \mathbb{E}u_i\left[\mathbf{p}_i(\pi_i, \pi_{-i}^{NE})\right]$$

- Nash equilibrium under ESR:

$$u_i\left[\mathbb{E}\mathbf{p}_i(\pi_i^{NE}, \pi_{-i}^{NE})\right] \geq u_i\left[\mathbb{E}\mathbf{p}_i(\pi_i, \pi_{-i}^{NE})\right]$$

Other solution concepts

- Cyclic Nash equilibria
  - No agent can improve their utility by unilaterally deviating from a joint cyclic strategy



- Correlated equilibria
  - No agent can improve their utility by unilaterally deviating from the recommendation of the correlated signal, given by an external mechanism

Multi-Objective Normal Form Games

- Introduced by Blackwell in 1956

- MONFG - tuple (N , A, **p**), with n ≥ 2 and C ≥ 2 objectives, where:
  - N = {1, ..., n} – set of players
  - $A = A_1 \times \cdots \times A_n$ – set of actions
  - $\mathbf{p} = (\mathbf{p}_1, \ldots, \mathbf{p}_n)$ – vectorial payoffs

Example - SER

$$u(p_1, p_2) = p_1 \cdot p_2$$

	A	B
A	(10, 2); (10, 2)	(0, 0); (0, 0)
B	(0, 0); (0, 0)	(2, 10); (2, 10)

Example - Nash equilibrium

$$u(p_1, p_2) = p_1 \cdot p_2$$

$$u(10, 2) = 10 \cdot 2 = 20$$

	A	B
A	(10, 2); (10, 2)	(0, 0); (0, 0)
B	(0, 0); (0, 0)	(2, 10); (2, 10)

Example - Cyclic Nash equilibrium

$$u(p_1, p_2) = p_1 \cdot p_2$$

$$u(\frac{10+2}{2}, \frac{2+10}{2}) = u(6,6) = 36$$

	A	B
A	(10, 2); (10, 2)	(0, 0); (0, 0)
B	(0, 0); (0, 0)	(2, 10); (2, 10)

- Joint cyclic strategy
  - Player 1: {A, B}
  - Player 2: {A, B}

Example - Correlated equilibrium

$$u(p_1, p_2) = p_1 \cdot p_2$$

$$u(\frac{10+2}{2}, \frac{2+10}{2}) = u(6,6) = 36$$

	A	B
A	(10, 2); (10, 2)	(0, 0); (0, 0)
B	(0, 0); (0, 0)	(2, 10); (2, 10)

- Correlated strategy σ
  - 50% (A, A)
  - 50% (B, B)

SOTA

Latest results and open challenges

(Im)balancing Act Game

- 2 players, 2 objective
- Same payoff vector for both players

	L	M	R
L	[4,0]	[3,1]	[2,2]
M	[3,1]	[2,2]	[1,3]
R	[2,2]	[1,3]	[0,4]

$$u_1([p_1, p_2]) = p_1^2 + p_2^2$$
$$u_2([p_1, p_2]) = p_1 \cdot p_2$$

ESR Equilibrium

- equilibrium 1: (0.75, 0, 0.25) and (0, 1, 0)
  expected utilities: 10 and 3
- equilibrium 2: (0.25, 0, 0.75) and (0, 1, 0)
  expected utilities: 10 and 3

ESR	L	M	R
L	16,0	10,3	8,4
M	10,3	8,4	10,3
R	8,4	10,3	16,0

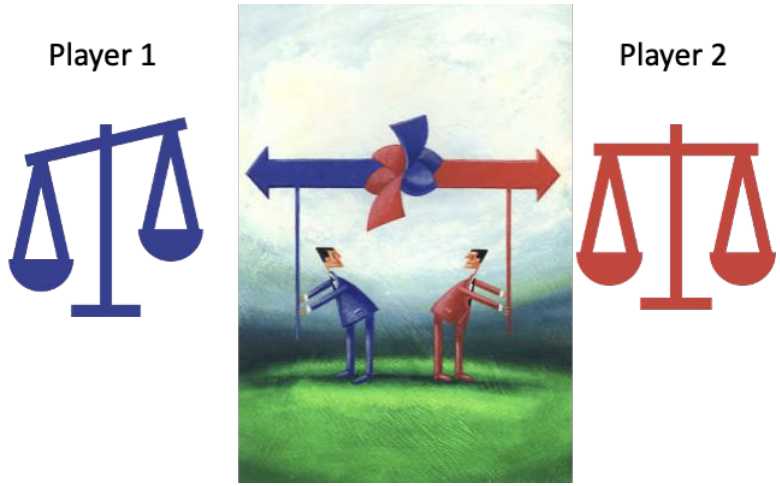
	L	M	R
L	[4,0]	[3,1]	[2,2]
M	[3,1]	[2,2]	[1,3]
R	[2,2]	[1,3]	[0,4]

$$u_1([p_1, p_2]) = p_1^2 + p_2^2$$
$$u_2([p_1, p_2]) = p_1 \cdot p_2$$

Rădulescu, R., Mannion, P., Zhang, Y., Roijers, D. M., & Nowé, A. (2020). A utility-based analysis of equilibria in multi-objective normal-form games. *The Knowledge Engineering Review*, *35*.

SER Equilibrium?

- In finite MONFGs, where each agent seeks to maximise the utility under **SER**, **Nash equilibria need not exist.**



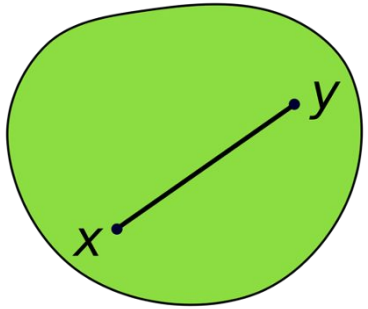
	L	M	R
L	[4,0]	[3,1]	[2,2]
M	[3,1]	[2,2]	[1,3]
R	[2,2]	[1,3]	[0,4]

$$u_1([p_1, p_2]) = p_1^2 + p_2^2$$
$$u_2([p_1, p_2]) = p_1 \cdot p_2$$

Rădulescu, R., Mannion, P., Zhang, Y., Roijers, D. M., & Nowé, A. (2020). A utility-based analysis of equilibria in multi-objective normal-form games. *The Knowledge Engineering Review*, *35*.

Bridging continuous games and MONFGs

- Continuous games:
  - Single objective
  - Infinite number of pure strategies
  - Continuous payoff functions
  - Benefit from many theoretical results
  - Algorithmically challenging



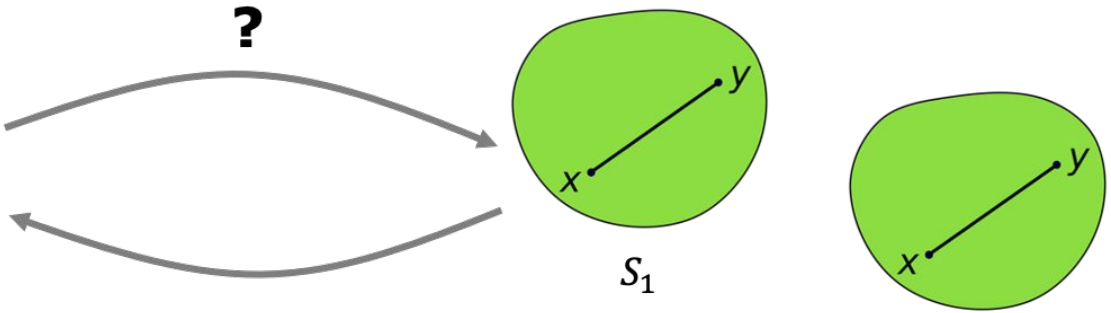
Assumption: convex strategy set

Röpke, W., Groenland, C., Rădulescu, R., Nowé, A., & Roijers, D. M. (2023). Bridging the Gap Between Single and Multi Objective Games. *AAMAS 2023.*

Bridging continuous games and MONFGs

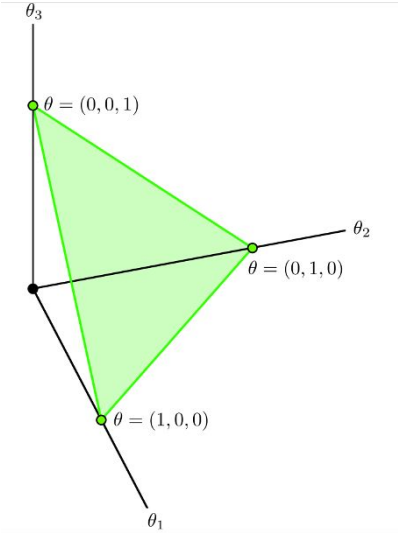
- Build mapping between MONFGs and continuous games
- Ensure that it preserves key dynamics
- Leverage the link for theoretical and algorithmic improvements

	A	B
A	(10, 2); (10, 2)	(2, 3); (2, 3)
B	(4, 2); (4, 2)	(6, 3); (6, 3)



Bridging continuous games and MONFGs

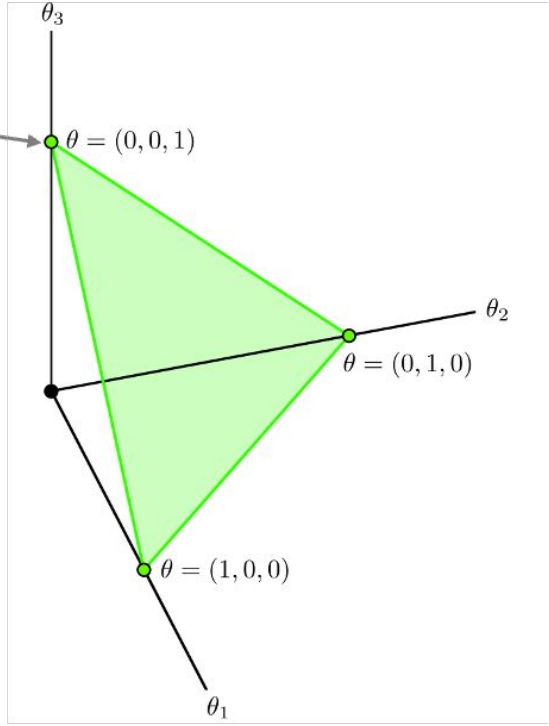
- Every mixed strategy in the MONFG becomes a pure strategy in the continuous game



Röpke, W., Groenland, C., Rădulescu, R., Nowé, A., & Roijers, D. M. (2023). Bridging the Gap Between Single and Multi Objective Games. *AAMAS 2023.*

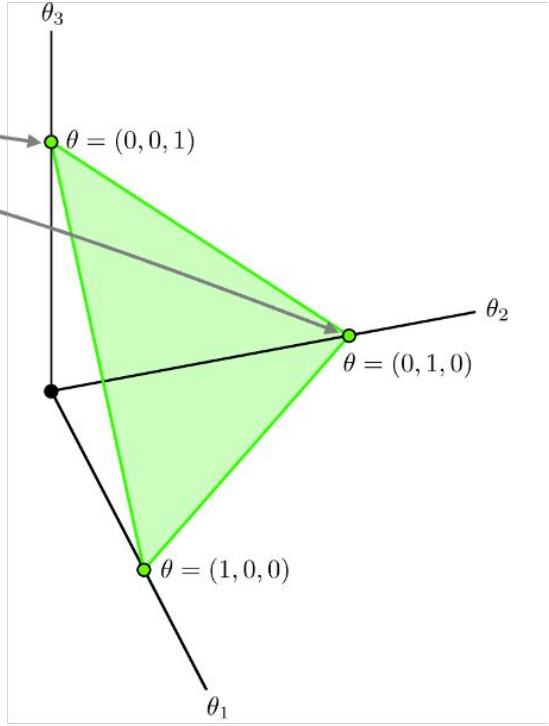
Bridging continuous games and MONFGs

	A	B	C
A	(4, 1); (4, 1)	(1, 2); (4, 2)	(2, 1); (1, 2)
B	(3, 1); (2, 3)	(3, 2); (6, 3)	(1, 2); (2, 1)
C	(1, 2); (2, 1)	(2, 1); (1, 2)	(1, 3); (1, 3)



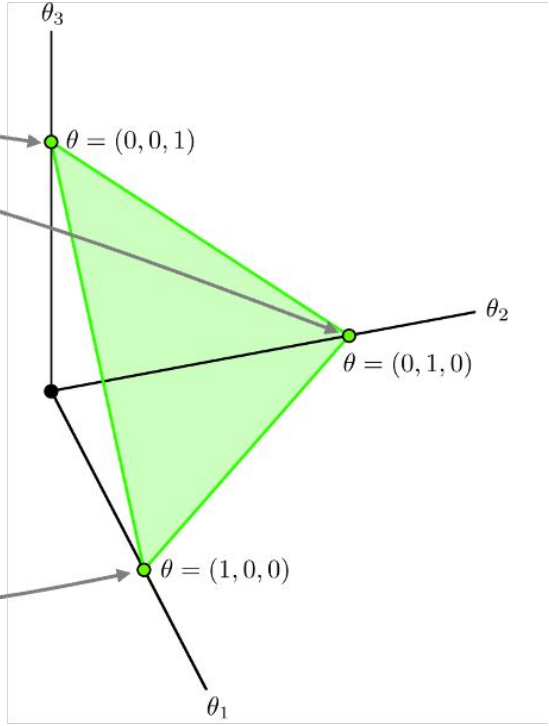
Bridging continuous games and MONFGs

	A	B	C
A	(4, 1); (4, 1)	(1, 2); (4, 2)	(2, 1); (1, 2)
B	(3, 1); (2, 3)	(3, 2); (6, 3)	(1, 2); (2, 1)
C	(1, 2); (2, 1)	(2, 1); (1, 2)	(1, 3); (1, 3)



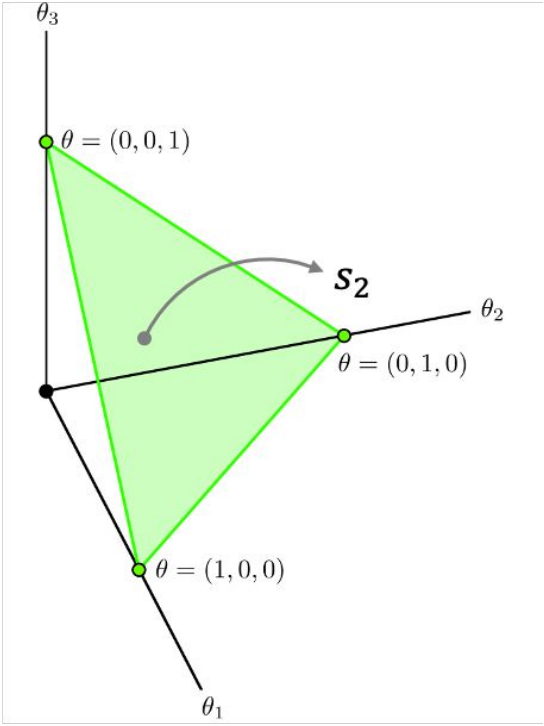
Bridging continuous games and MONFGs

	A	B	C
A	(4, 1); (4, 1)	(1, 2); (4, 2)	(2, 1); (1, 2)
B	(3, 1); (2, 3)	(3, 2); (6, 3)	(1, 2); (2, 1)
C	(1, 2); (2, 1)	(2, 1); (1, 2)	(1, 3); (1, 3)



Bridging continuous games and MONFGs

	$\frac{1}{4}$ A	$\frac{1}{4}$ B	$\frac{2}{4}$ C
A	(4, 1); (4, 1)	(1, 2); (4, 2)	(2, 1); (1, 2)
B	(3, 1); (2, 3)	(3, 2); (6, 3)	(1, 2); (2, 1)
C	(1, 2); (2, 1)	(2, 1); (1, 2)	(1, 3); (1, 3)

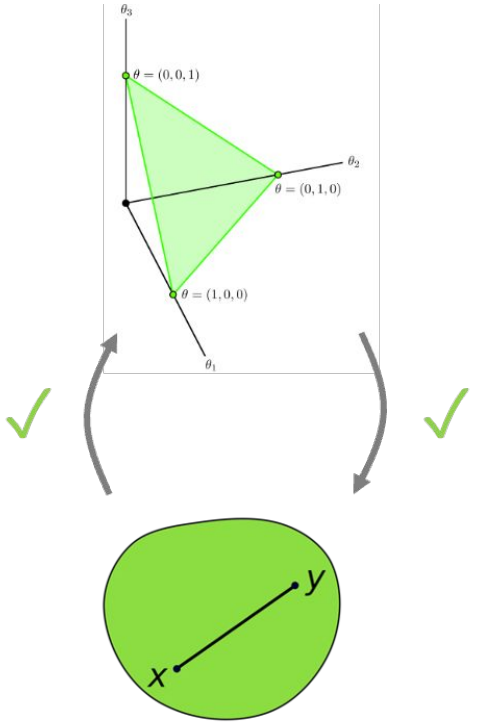


Theoretical insights

- Mixed strategy equilibria in the MONFG are pure strategy equilibria in the continuous game

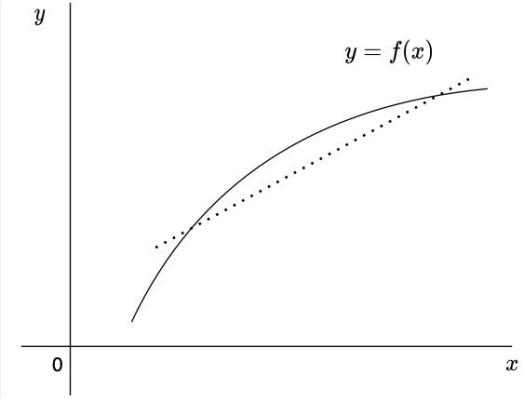
- Continuous games are not guaranteed to have a pure strategy Nash equilibrium

  - Nash equilibria are not guaranteed in MONFGs



NE Existence Guarantees

- Existence is guaranteed with (quasi)concave utility functions
  - Used in economics as well
  - Represents “well-behaved” preferences
- Intuition
  - MONFGs can be reduced to continuous games
  - In these game it is known that a pure strategy NE exists when assuming only quasiconcave utility functions
  - This equilibrium is also an equilibrium in the original MONFG



Röpke, W., Roijers, D. M., Nowé, A., & Rădulescu, R. (2022). **On nash equilibria in normal-form games with vectorial payoffs.** *Autonomous Agents and Multi-Agent Systems*, *36*(2), 53.

Non-existence

- We can show that no Nash equilibrium exists in this game
  - With **strict convex utility functions**

	A	B
A	(2, 0); (1, 0)	(1, 0); (0, 2)
B	(0, 1); (2, 0)	(0, 2); (0, 1)

$$u_1(p_1, p_2) = u_2(p_1, p_2) = p_1^2 + p_2^2$$

Relations between optimisation criteria

- **Mixed strategies**
  - **No relation** between both optimisation criteria **in general**

	A	B
A	(1, 0); (1, 0)	(0, 1); (0, 1)
B	(0, 1); (0, 1)	(-10, 0); (-10, 0)

Multi-objective reward vectors

$$u(x, y) = 0.1 * x + max(0, x) * max(0, y)$$

	A	B
A	0.1; 0.1	0; 0
B	0; 0	-0.1; -0.1

Scalarised utility for both agents

No sharing of number of equilibria or equilibria themselves

Relations between optimisation criteria

- **Pure strategies**
  - Pure strategy equilibrium under SER is also one under ESR
  - Bidirectional when assuming (quasi)convex utility functions

- We can extend this to **blended settings**
  - Pure strategy equilibrium under SER is also one in any blended setting
  - Bidirectional when assuming (quasi)convex utility functions

Additional work

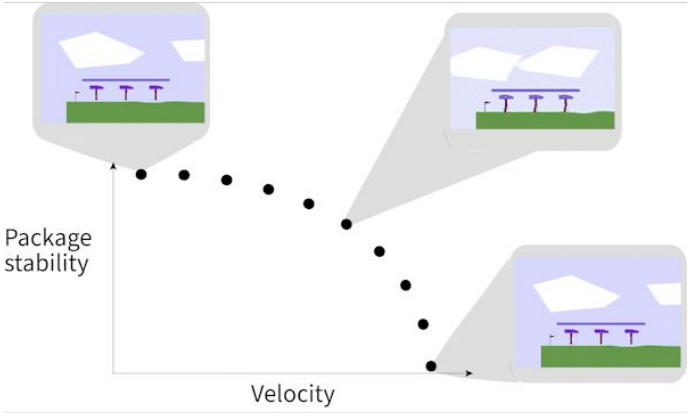
- Conor F. Hayes, Timothy Verstraeten, Diederik M. Roijers, Enda Howley, Patrick Mannion - Multi-Objective Coordination Graphs for the Expected Scalarised Returns with Generative Flow Models. *In European Workshop on Reinforcement Learning (EWRL 2022)*, Milan, 2022

- Rădulescu, R., Verstraeten, T., Zhang, Y., Mannion, P., Roijers, D. M., & Nowé, A. (2022). Opponent learning awareness and modelling in multi-objective normal form games. *Neural Computing and Applications*, 1-23.

- Röpke, W., Roijers, D. M., Nowé, A., & Rădulescu, R. (2022). Preference communication in multi-objective normal-form games. *Neural Computing and Applications*, 1-26.

MOMAland – MOMADM benchmarks

- Open source Python library for developing and comparing multi-objective multi-agent reinforcement learning algorithms



MOMAland

https://momaland.farama.org/

MOMAland – MOMADM benchmarks

Domain	# of agents	# of objectives	stochastic transitions?	full observability possible?	partial observability possible?	team rewards possible?	individual rewards possible?	discrete/continuous state (d/c)	discrete/continuous actions (d/c)
MO-BPD	2-n	2	✗[3]	✗	✓	✓	✓	d	d
MO-ItemGathering	2-n	2-d	✗[3]	✓	✗	✗	✓	d	d
MO-GemMining	2-n	2-d	✗[3]	-	-	✗	✓	-	d
MO-RouteChoice	2-n	2	✗[3]	-	-	✗	✓	-	d
MO-PistonBall	2-n	3	✗[3]	✗	✓	✗	✓	c	d/c
MO-MW-Stability	2-n	2	✗	✓	✓	✓	✓	c	c
CrazyRL/Surround	2-n	2	✗	✓	✗	✓	✓	c	c
CrazyRL/Escort	2-n	2	✗	✓	✗	✓	✓	c	c
CrazyRL/Catch	2-n	2	✓	✓	✗	✓	✓	c	c
MO-Breakthrough	2	1-4	✗	✓	✗	✗	✓	d	d
MO-Connect4	2	2-20	✗	✓	✗	✗	✓	d	d
MO-Ingenious	2-6	2-6	✗	✓	✓	✓	✓	d	d
MO-SameGame	1-5	2-10	✗[3]	✓	✗	✓	✓	d	d

MOMAland

https://momaland.farama.org/

Open questions

- Results for more complex (e.g., sequential, partially observable) settings
- Integrated pipelines for planning -> negotiation -> execution
- Utility modelling
- Strategic disclosure of utility information to the other agents

Multi-Objective Decision Making Workshop

- At ECAI 2024
- Santiago de Compostela, Spain
- Workshop: October 20
- Deadline: June 15
- Special Issue: NCAA (IF 6)

https://modem2024.vub.ac.be/

Thank you for listening

- Questions?

- Any additional questions drop us a message at:
  - ann.nowe@vub.be
  - r.t.radulescu@uu.nl

This tutorial was based (primarily) on

- Hayes, C. F., Rădulescu, R., Bargiacchi, E., Källström, J., Macfarlane, M., Reymond, M., ... & Roijers, D. M. (2022). A practical guide to multi-objective reinforcement learning and planning. Autonomous Agents and Multi-Agent Systems, 36(1), 26.
- Rădulescu, R., Mannion, P., Roijers, D. M., & Nowé, A. (2020). Multi-objective multi-agent decision making: a utility-based analysis and survey. Autonomous Agents and Multi-Agent Systems, 34(1), 1-52.
- Rădulescu, R., Mannion, P., Zhang, Y., Roijers, D. M., & Nowé, A. (2020). A utility-based analysis of equilibria in multi-objective normal-form games. The Knowledge Engineering Review, 35.
- Rădulescu, R. (2021). Decision Making in Multi-Objective Multi-Agent Systems: A Utility-Based Perspective. Brussels: Crazy Copy Center Productions.
- Roijers, D. M., Vamplew, P., Whiteson, S., & Dazeley, R. (2013). A survey of multi-objective sequential decision-making. Journal of Artificial Intelligence Research, 48, 67-113.
- Röpke, W., Roijers, D. M., Nowé, A., & Rădulescu, R. (2022). On Nash equilibria in normal-form games with vectorial payoffs. Autonomous Agents and Multi-Agent Systems, 36(2), 53.
- Röpke, W., Groenland, C., Rădulescu, R., Nowé, A., & Roijers, D. M. (2023). Bridging the Gap Between Single and Multi Objective Games. AAMAS 2023.