Multiobjective reinforcement learning survey

О чем пойдет речь

- Напоминание стандартной задачи RL.
- MORL. Отличие от стандартного RL. Для чего нужен MORL.
- Подходы к решению задачи MORL.
  - Single policy. Scalarization function.
  - Multi policy. Pareto frontier. Pareto Q-learning.
- Pareto Q-learning and single policy algorithm comparison.
  - Hypervolume Indicator.
  - The Deep sea treasure world.
- Thresholded lexicographic reinforcement learning.
- Stochastic mixture policy for episodic MORL.
- Convex hull value iteration.

- Напоминание стандартной задачи RL.
- MORL. Отличие от стандартного RL. Нужен ли MORL?
- Подходы к решению задачи MORL.
  - Single policy. Scalarization function.
  - Multi policy. Pareto frontier. Pareto Q-learning.
- Pareto Q-learning and single policy algorithm comparison
  - Hypervolume Indicator.
  - The Deep Sea Treasure world.
- Thresholded lexicographic reinforcement learning.
- Stochastic mixture policy for episodic MORL.
- Convex hull value iteration.

Напоминание стандартной задачи RL

$$MDP = (S, A, T, R, \gamma)$$

Хотим максимизировать R_t, где

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

State value function:

$$V^{\pi}(s) = E[R^t \mid \pi, s_t = s]$$

Напоминание стандартной задачи RL

Bellman equation

$$V^{\pi}(s) = \sum_{a} \pi(s,a) \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V^{\pi}(s')]$$

Bellman optimality equation

$$V^{*}(s) = \max_{a} \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V^{*}(s')]$$

- Напоминание стандартной задачи RL.
- MORL. Отличие от стандартного RL. Нужен ли MORL?
- Подходы к решению задачи MORL.
  - Single policy. Scalarization function.
  - Multi policy. Pareto frontier. Pareto Q-learning.
- Pareto Q-learning and single policy algorithm comparison
  - Hypervolume Indicator.
  - The Deep Sea Treasure world.
- Thresholded lexicographic reinforcement learning.
- Stochastic mixture policy for episodic MORL.
- Convex hull value iteration.

Отличие от стандартной постановки задачи

Единственное отличие заключается в том, что теперь наша функция награды в MDP это не скаляр, а вектор

$$R(s,a) = (R_1(s,a), R_2(s,a), \ldots, R_m(s,a))$$

Обычно компоненты вектора наград противоречивы, то есть улучшение одной компоненты приводит к ухудшению какой-либо другой компоненты

Примеры противоречивых наград

- Максимальная прибыль за кратчайшие сроки
- Польза препарата и его побочные действия
- Система управления трафиком должна минимизировать задержку и максимизировать пропускную способность
- Распределение времени, например, на учебные курсы
- ...

Функция скаляризации.

Функцией скаляризации будем называть такую функцию f, которая переводит вектор наград в скаляр.

Более формально:

$$V_w^\pi(s) = f(V^\pi(s), w)$$

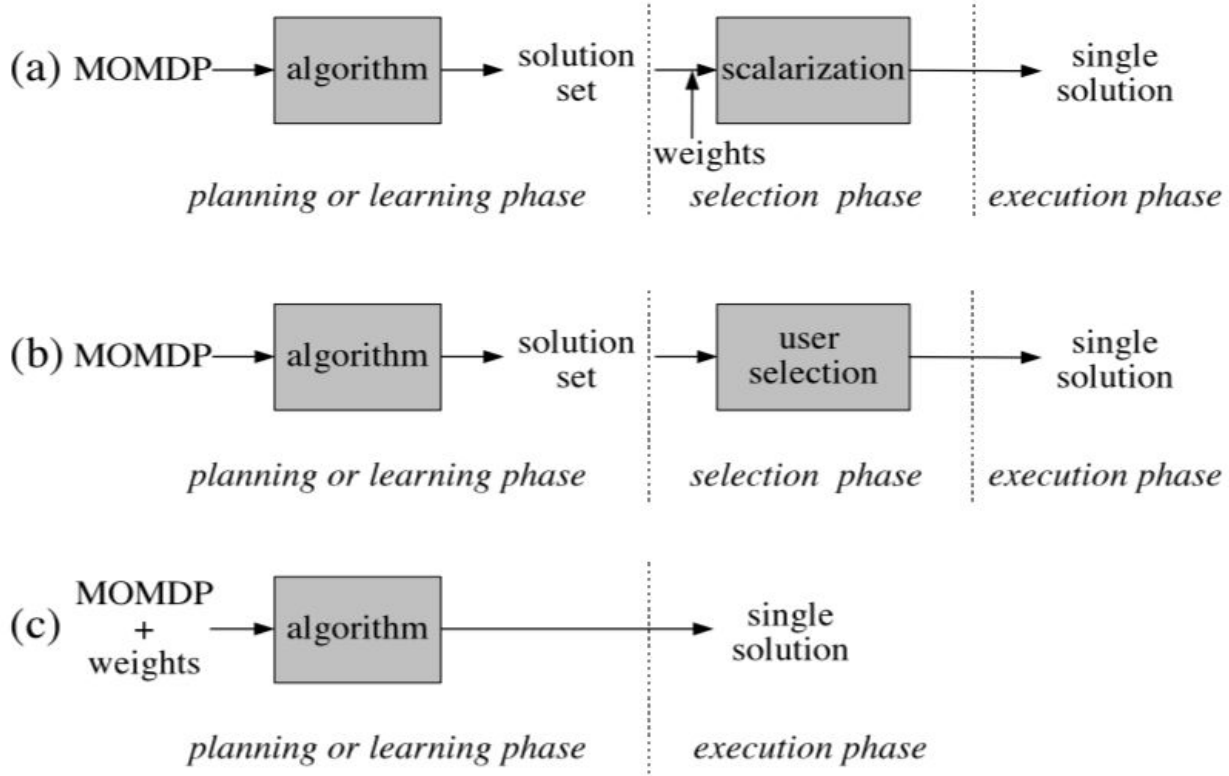
Где w - вектор весов, параметризующий f.

Зачем нужен MORL?

Чтобы понять, почему нам не хватает стандартной задачи RL, и MORL может быть очень полезен, рассмотрим следующие примеры.

- Предположим, что мы решаем задачу оптимизации для транспортной системы, пытаясь минимизировать время, которое требуется людям с окраин для того, чтобы добраться до работы, при этом мы также пытаемся сильно не навредить окружающей среде. В нашем штате компании есть экономисты, которые могут рассчитать убытки от загрязнения окружающей среды, а также рассчитать, какой убыток понесет компания от потери продуктивности рабочих во время их путешествия на работу. Более того соотношение убытков от этих двух компонент на открытом рынке постоянно меняется. Если наша модель транспортной системы достаточно сложная, то каждый раз пересчитывать оптимальное решение будет очень затратно. Именно здесь и может помочь MORL: мы изначально можем найти всевозможные оптимальные решения, не опираясь на фиксированное соотношение цен, быть может, потратив чуть больше ресурсов единожды, но зато затем при очередном изменении цен мы легко сможем выбрать оптимальное решение из множества имеющихся.
- Также может возникнуть ситуация, когда экономисты не могут явно перевести компоненты наград в денежный эквивалент. В этом случае мы можем также найти решения при всевозможных соотношениях на компоненты наград, а потом попросить шарящих в этой сфере дядек по этим решениям выбрать нужное.
- Возможен и третий случай, когда нам априори известно соотношение на компоненты, и оно не изменяется, но здесь может быть проблема с аддитивностью функции скаляризации.

Иллюстрация трех описанных сценариев.



Оптимальность в MOMDP

Теперь не совсем понятно, что означает оптимальная стратегия, ведь вполне возможна такая ситуация, что для двух рассматриваемых стратегий:

$$V_i^{\pi_1}(s) > V_i^{\pi_2}(s),\ V_j^{\pi_1}(s) < V_j^{\pi_2}(s)$$

Получается, что нам нужны какие-то априорные знания на компоненты наград, или же нам нужно учить сразу несколько оптимальных стратегий

В случае нескольких стратегий в качестве критерия оптимальности обычно рассматривают оптимальность по Парето:

$$\pi\ is\ Pareto\ optimal\ if\ \nexists \pi' : V^{\pi'}(s_0) \succ V^{\pi}(s_0)$$

- Напоминание стандартной задачи RL.
- MORL. Отличие от стандартного RL. Нужен ли MORL?
- Подходы к решению задачи MORL.
  - Single policy. Scalarization function.
  - Multi policy. Pareto frontier. Pareto Q-learning.
- Pareto Q-learning and single policy algorithm comparison
  - Hypervolume Indicator.
  - The Deep Sea Treasure world.
- Thresholded lexicographic reinforcement learning.
- Stochastic mixture policy for episodic MORL.
- Convex hull value iteration.

Single policy

Главная идея заключается в том, что мы сводим MORL к обычной задаче RL путем **скаляризации** функции награды

$$Q_w(s,a) = f(Q(s,a), w)$$

Например, линейная скаляризация

$$\hat{\mathbf{Q}}(s,a) = (\hat{Q}_1(s,a), \ldots, \hat{Q}_m(s,a))$$

$$\widehat{SQ}_{linear}(s,a) = \sum_{o=1}^{m} \mathbf{w}_o \cdot \hat{\mathbf{Q}}_o(s,a).$$

Scalarized multi-objective Q-learning

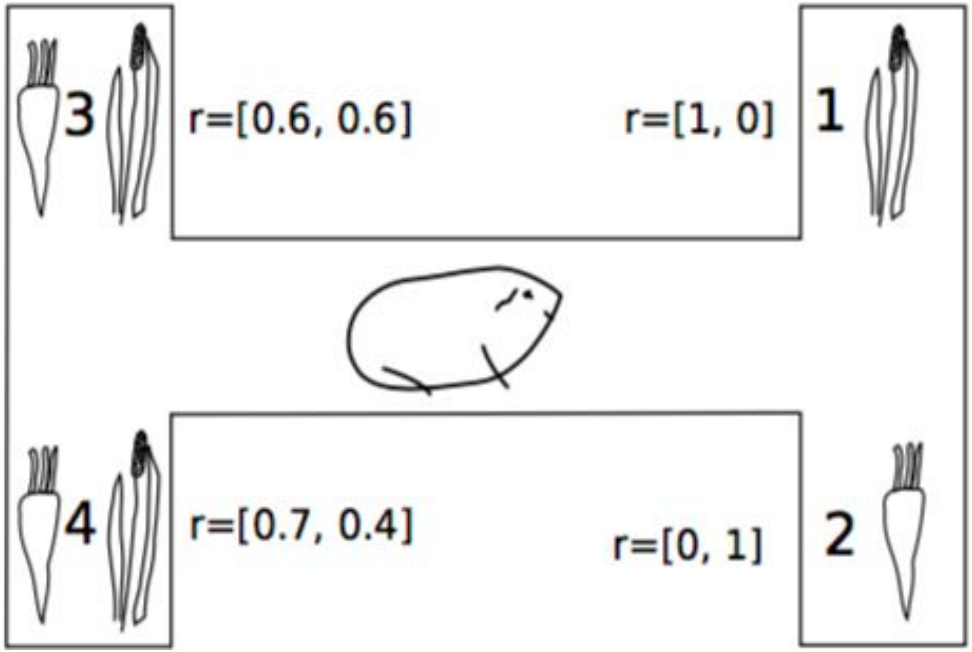
Initialize $\hat{Q}_o(s, a)$ arbitrarily
for each episode t **do**
    Initialize state s
    repeat
        Choose action a from s using the policy derived from $\widehat{SQ}$-values, e.g., *scal*-ϵ-greedy
        Take action a and observe state $s' \in S$ and reward vector $\mathbf{r} \in \mathbb{R}^m$
        $a' \leftarrow greedy(s')$ ▷ Call scal. greedy action selection
        for each objective o **do**
            $\hat{Q}_o(s, a) \leftarrow \hat{Q}_o(s, a) + \alpha_t(\mathbf{r}_o + \gamma\, \hat{Q}_o(s', a') - \hat{Q}_o(s, a))$
        end for

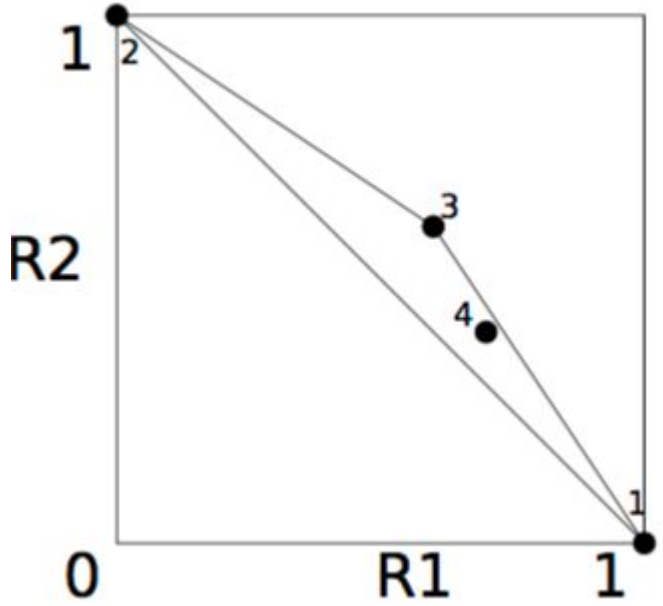
        $s \leftarrow s'$ ▷ Proceed to next state
    until s is terminal
end for

Недостатки single policy scalarization

- Находит только одну стратегию.
- Заранее нужны знания о соотношении важности компонент вектора наград.
- Найденная стратегия очень зависит от вектора весов, при малейшем изменении можем получить совсем другую стратегию.
- При использовании линейной скаляризации находит только стратегии, лежащие на выпуклой оболочке Парето фронта.
- При некоторых функциях скаляризации теряется аддитивность награды, поэтому мы не можем утверждать, что найденная стратегия является оптимальной, так как **Bellman optimality equation** справедливо только для аддитивной функции награды.

Guinea pig example





- Напоминание стандартной задачи RL.
- MORL. Отличие от стандартного RL. Нужен ли MORL?
- **Подходы к решению задачи MORL.**
  - Single policy. Scalarization function.
  - **Multi policy. Pareto frontier. Pareto Q-learning.**
- Pareto Q-learning and single policy algorithm comparison
  - Hypervolume Indicator.
  - The Deep Sea Treasure world.
- Thresholded lexicographic reinforcement learning.
- Stochastic mixture policy for episodic MORL.
- Convex hull value iteration.

Multi policy

В отличие от single policy данный подход не уменьшает размерность функции наград, пытаясь выучить сразу несколько политик, тем самым аппроксимируя Парето фронт.

Pareto Q-learning

$$V^{ND}(s') = ND(\cup_{a'} \hat{Q}_{set}(s', a'))$$

ND - оператор, который оставляет только множество non-dominated элементов.

Обновляем Q по такому правилу

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

Храним отдельно ***R(s, a)*** отдельно ***ND(s, a)***

Pareto Q-learning algorithm

Initialize $\hat{Q}_{set}(s,a)$'s as empty sets
for each episode t **do**
  Initialize state s
  repeat
    Choose action a from s using a policy derived from the $\hat{Q}_{set}$'s
    Take action a and observe state $s' \in S$ and reward vector $\mathbf{r} \in \mathbb{R}^m$

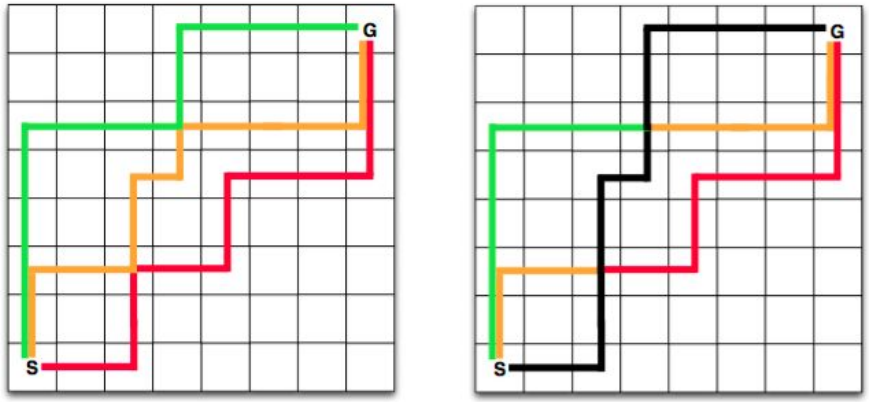
    $ND_t(s,a) \leftarrow ND(\cup_{a'} \hat{Q}_{set}(s',a'))$ ▷ Update ND policies of s' in s
    $\overline{\mathfrak{R}}(s,a) \leftarrow \overline{\mathfrak{R}}(s,a) + \frac{\mathbf{r} - \overline{\mathfrak{R}}(s,a)}{n(s,a)}$ ▷ Update average immediate rewards
    $s \leftarrow s'$ ▷ Proceed to next state
  until s is terminal
end for

Pareto Q-learning algorithm. Consistently tracking policy.

Допустим, что после того, как мы посчитали $Q(s, a)$, $R(s, a)$, $ND(s, a)$, нам поступила какая-то информация о том, как выбрать единственную самую лучшую стратегию из множества полученных, то есть мы получили некоторое соотношение на компоненты награды. Как тогда выбрать оптимальную стратегию?

Предположим для простоты, что у нас детерминированная среда, и рассмотрим некоторый пример, иллюстрирующий данную проблему.

Pareto Q-learning algorithm. Consistently tracking policy.



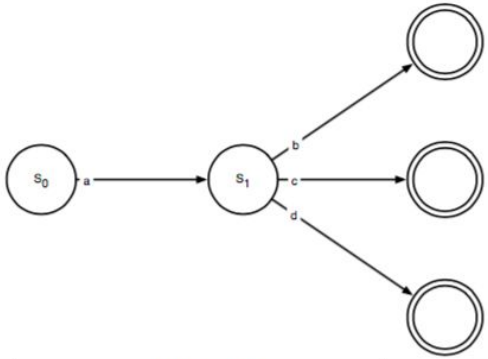
(a) (b)

Пусть мы нашли три стратегии, которые эффективны по Парето: красная, оранжевая и зеленая траектории на графике (a). Мы последовательно строим стратегию, черная линия на графике (b), каждый раз выбирая локально оптимальное действие в каждом из посещенных состояний. В итоге может случится, что полученная черная траектория не является глобально оптимальной стратегией по Парето. Как избежать такого случая?

Consistently tracking policy algorithm.

1: $target \leftarrow \mathbf{V}^{\pi}(s)$
2: **repeat**
3: **for** each a in A **do**
4: Retrieve $\overline{\mathfrak{R}}(s, a)$
5: Retrieve $ND_t(s, a)$
6: **for** each $\mathbf{Q}$ in $ND_t(s, a)$ **do**
7: **if** $\gamma\,\mathbf{Q} + \overline{\mathfrak{R}}(s, a) = target$ **then**
8: $s \leftarrow s' : T(s'|s, a) = 1$
9: $target \leftarrow \mathbf{Q}$
10: **end if**
11: **end for**
12: **end for**
13: **until** s is not terminal

Пример для Consistently tracking policy algorithm.



Action	$\overline{\mathfrak{R}}(s,a)$	$ND_t(s,a)$	$\hat{Q}_{set}(s,a)$
a	(0.2, 0.0)	((0.9, 0.5), (2.0, 0.4), (0.0, 0.6))	((1.1, 0.5), (2.2, 0.4) , (0.2, 0.6))
b	(0.9, 0.5)	()	(0.9, 0.5)
c	(2.0, 0.4)	()	(2.0, 0.4)
d	(0.0, 0.6)	()	(0.0, 0.6)

Хотим получить стратегию с наградой (2.2, 0.4). Первым шагом выбираем действие a, оно единственное, а вот второе действие выбирается следующим путем:

$$R(s', action) + ND(s', action) = (2.2, 0.4) - (0.2, 0.0) = (2.0, 0.4) \Rightarrow action = c$$

- Напоминание стандартной задачи RL.
- MORL. Отличие от стандартного RL. Нужен ли MORL?
- Подходы к решению задачи MORL.
  - Single policy. Scalarization function.
  - Multi policy. Pareto frontier. Pareto Q-learning.
- **Pareto Q-learning and single policy algorithm comparison**
  - **Hypervolume Indicator.**
  - The Deep Sea Treasure world.
- Thresholded lexicographic reinforcement learning.
- Stochastic mixture policy for episodic MORL.
- Convex hull value iteration.

Hypervolume indicator. Определение.

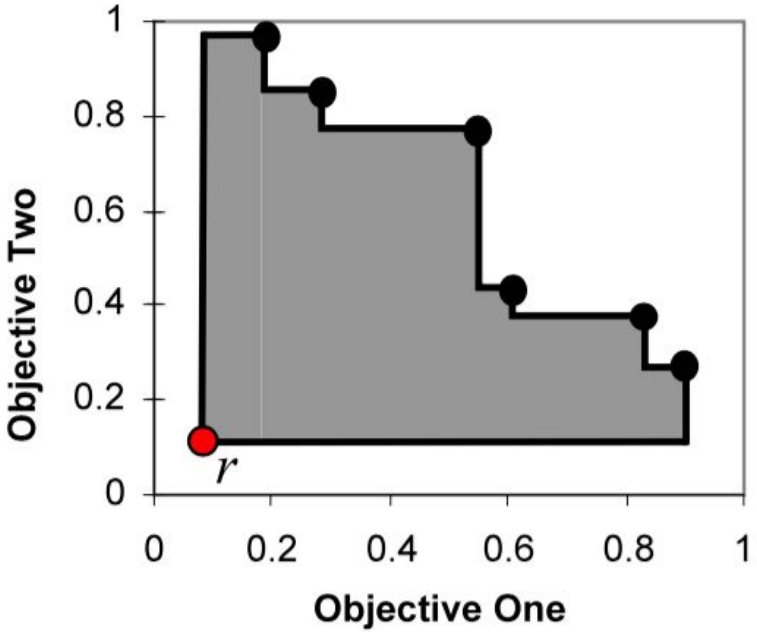
Формальное определение Hypervolume indicator с опорной точкой (0, …, 0) :

$$I_H^*(A) := \int_{(0,\ldots,0)}^{(1,\ldots,1)} \alpha_A(\boldsymbol{z}) d\boldsymbol{z}$$

$$\alpha_A(\boldsymbol{z}) := \begin{cases} 1 & if\ A \succeq \{\boldsymbol{z}\} \\ 0 & else \end{cases}$$

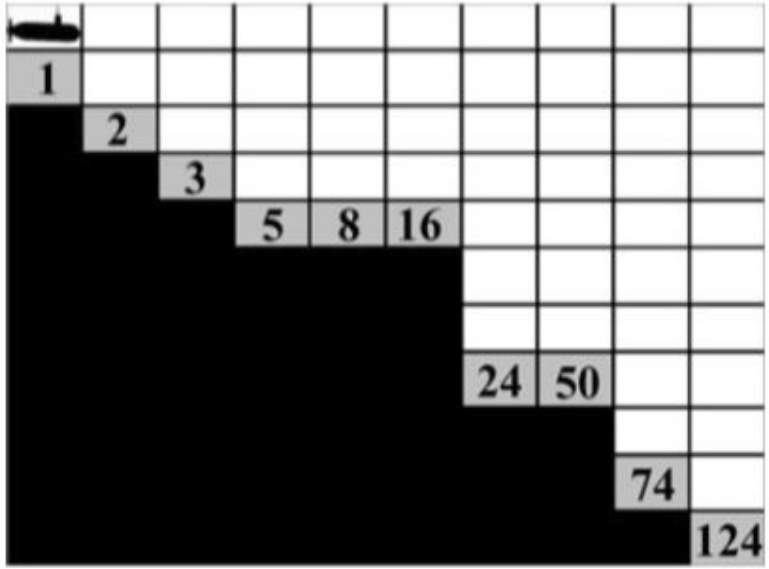
Важным свойство HV является тот факт, что если любой элемент множества B не строго доминируется хотя бы одним элементов множества A, но не наоборот, то HV это отображает: $I(A) \geq I(B)$

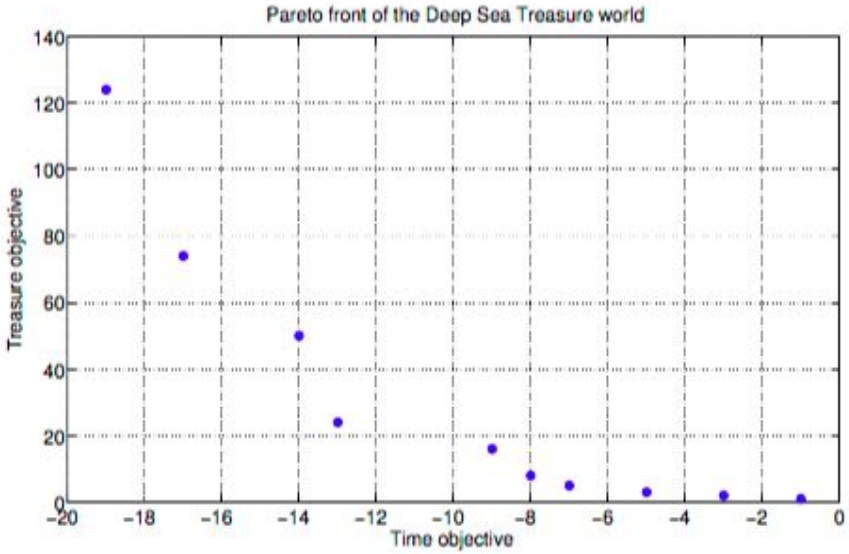
Hypervolume indicator. Двумерный случай.

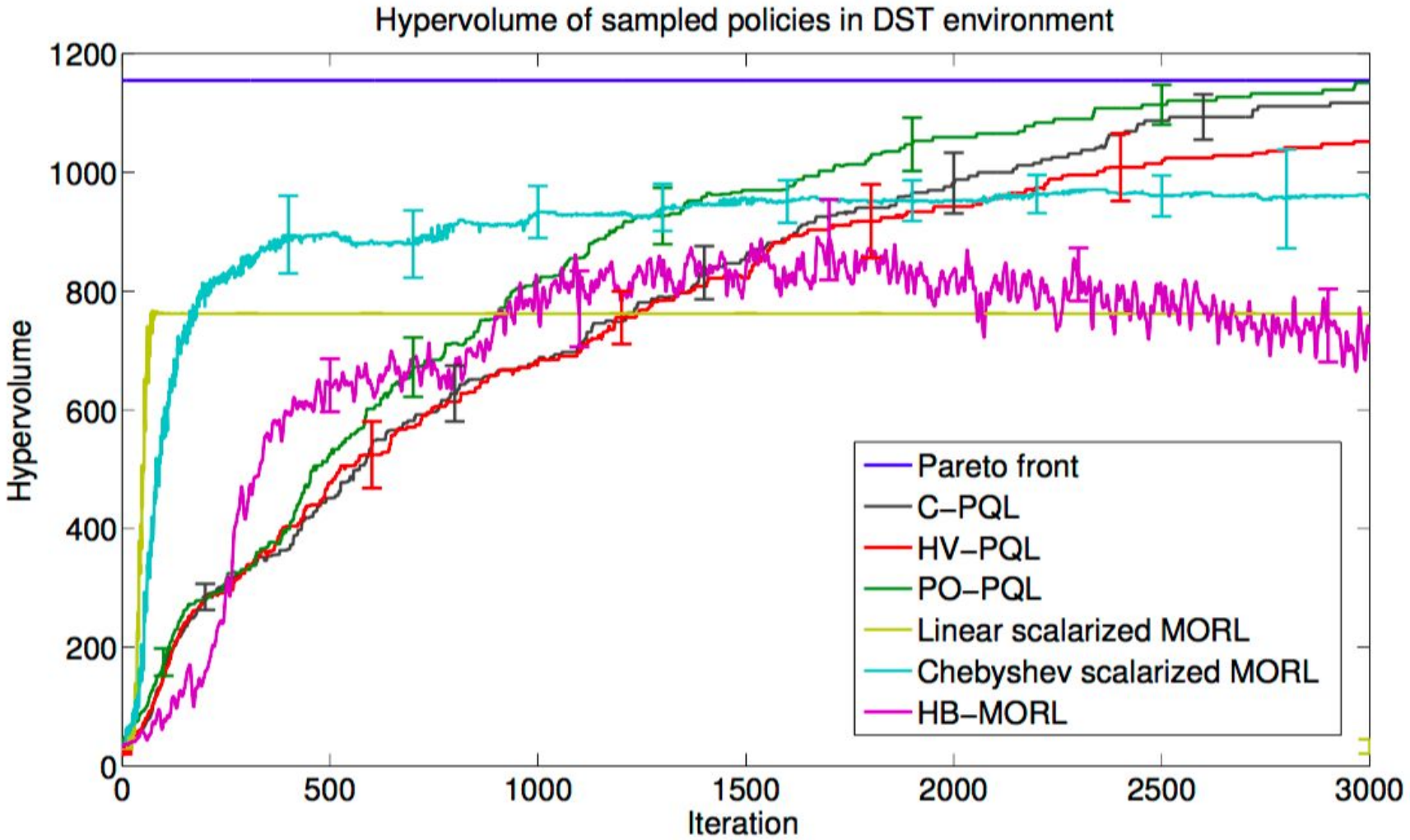


- Напоминание стандартной задачи RL.
- MORL. Отличие от стандартного RL. Нужен ли MORL?
- Подходы к решению задачи MORL.
  - Single policy. Scalarization function.
  - Multi policy. Pareto frontier. Pareto Q-learning.
- **Pareto Q-learning and single policy algorithm comparison**
  - Hypervolume Indicator.
  - **The Deep Sea Treasure world.**
- Thresholded lexicographic reinforcement learning.
- Stochastic mixture policy for episodic MORL.
- Convex hull value iteration.

Deep sea treasure world






Hypervolume of sampled policies in DST environment
Hypervolume
Iteration
0
200
400
600
800
1000
1200
0
500
1000
1500
2000
2500
3000
Pareto front
C−PQL
HV−PQL
PO−PQL
Linear scalarized MORL
Chebyshev scalarized MORL
HB−MORL

- Напоминание стандартной задачи RL.
- MORL. Отличие от стандартного RL. Нужен ли MORL?
- Подходы к решению задачи MORL.
  - Single policy. Scalarization function.
  - Multi policy. Pareto frontier. Pareto Q-learning.
- Pareto Q-learning and single policy algorithm comparison
  - Hypervolume Indicator.
  - The Deep Sea Treasure world.
- **Thresholded lexicographic reinforcement learning.**
- Stochastic mixture policy for episodic MORL.
- Convex hull value iteration.

Thresholded lexicographic reinforcement learning

Пусть мы опять находимся в ситуации, когда наш reward это вектор, но теперь перед нами стоит немного другая задача, а именно:

$$\begin{cases} R = (R_1, R_2, \cdots, R_n) \\ R_n \to max \\ \forall i \leq n-1 : R_i >= C_i \end{cases}$$

Thresholded lexicographic reinforcement learning algorithm

$$CQ_{s,a,j} \leftarrow \min(Q_{s,a,j}, C_j)$$

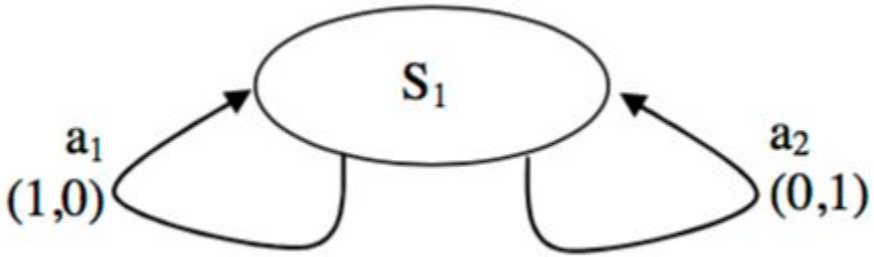
In state s, the greedy action a' is selected such that superior$(CQ_{s,a'}, CQ_{s,a}, 1)$ is true $\forall a \in A$ where superior$(CQ_{s,a'}, CQ_{s,a}, i)$ is recursively defined as:

```
if CQ_{s,a',i} > CQ_{s,a,i}
    return true
else if CQ_{s,a',i} = CQ_{s,a,i}
    if i = n
        return true
    else
        return superior(CQ_{s,a'}, CQ_{s,a}, i + 1)
else
    return false
```

- Напоминание стандартной задачи RL.
- MORL. Отличие от стандартного RL. Нужен ли MORL?
- Подходы к решению задачи MORL.
  - Single policy. Scalarization function.
  - Multi policy. Pareto frontier. Pareto Q-learning.
- Pareto Q-learning and single policy algorithm comparison
  - Hypervolume Indicator.
  - The Deep Sea Treasure world.
- Thresholded lexicographic reinforcement learning.
- Stochastic mixture policy for episodic MORL.
- Convex hull value iteration.

Stochastic mixture policy for episodic MORL

Проблема deterministic policies

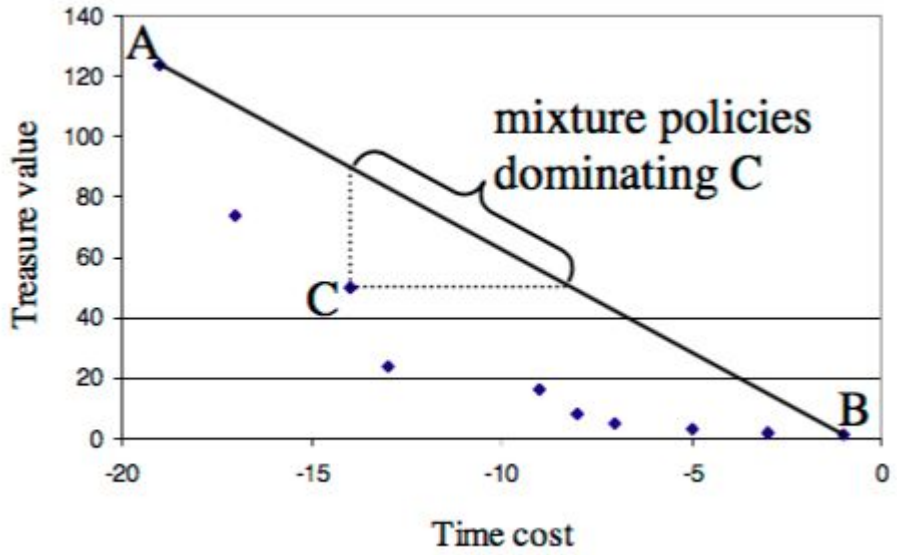


В такой среде с детерминированной политикой мы сможем получить только R = (1, 0) или R = (0, 1).

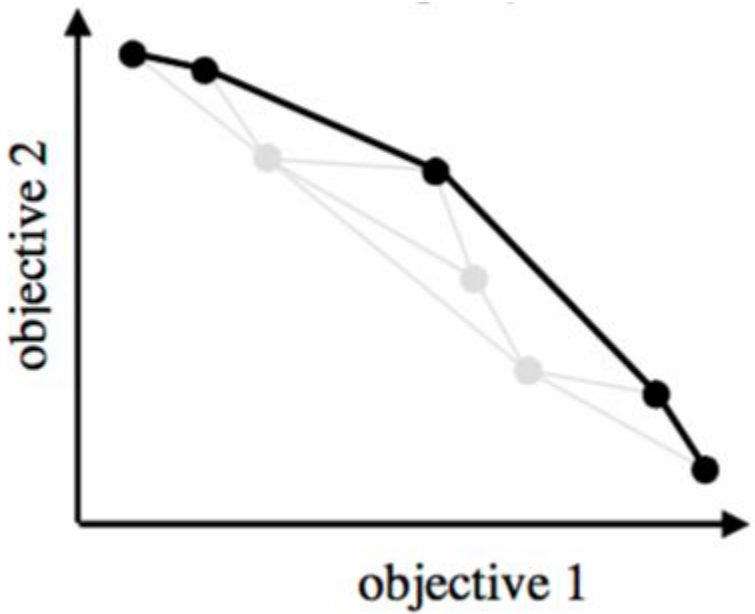
Если мы будем случайно с вероятностью p выбирать a1, варьируя p, мы сможем получить любой R = (p, 1-p).

Deep Sea Treasure task’s objective space

A, B - две политики, найденные, например, путем линейной скаляризации.



Наглядная иллюстрация Pareto front + mixture policies



Convex Hull Visualization and Barycentric Coefficients

$b_1, b_2, \ldots, b_n$ - Barycentric coefficients

$$\sum_{i=1}^{n} b_i = 1$$

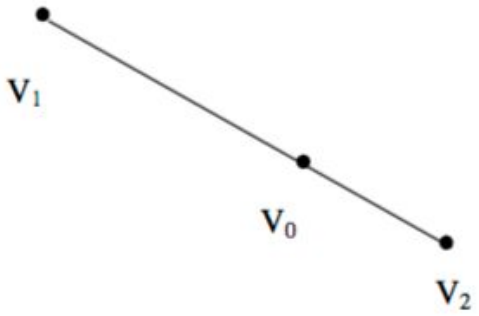
$$V_{0,j} = \sum_{1}^{n} b_i \cdot V_{i,j}$$

$V_{0,j}$ – jя компонента награды для mixture policy

Convex Hull Visualization and Barycentric Coefficients

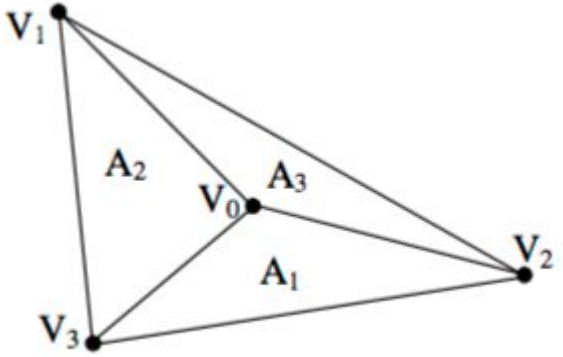
В двумерном случае:

$$b_i = \frac{\|V_i - V_0\|}{\|V_2 - V_1\|}$$



В трехмерном случае:

$$A_1 = \|(V_2 - V_0) \times (V_3 - V_0)\|$$
$$A_2 = \|(V_1 - V_0) \times (V_3 - V_0)\|$$
$$A_3 = \|(V_1 - V_0) \times (V_2 - V_0)\|$$
$$b_i = \frac{A_i}{\sum_{j=1}^{3} A_j}$$



- Напоминание стандартной задачи RL.
- MORL. Отличие от стандартного RL. Нужен ли MORL?
- Подходы к решению задачи MORL.
  - Single policy. Scalarization function.
  - Multi policy. Pareto frontier. Pareto Q-learning.
- Pareto Q-learning and single policy algorithm comparison
  - Hypervolume Indicator.
  - The Deep Sea Treasure world.
- Thresholded lexicographic reinforcement learning.
- Stochastic mixture policy for episodic MORL.
- Convex hull value iteration.

Convex hull value iteration

Хотим найти все policy, лежащие на выпуклой оболочке Парето фронта.

$$\forall w \; Q^*_w(s,a) = sup_\pi(w \cdot Q^\pi(s,a))$$

Convex hull value iteration

$\mathring{Q}(s,a)$ - точки выпуклой оболочки для пары *(s, a)*

- $\vec{u} + b\mathring{Q} \equiv \{\vec{u} + b\vec{q} : \vec{q} \in \mathring{Q}\}$
- $\mathring{Q} + \mathring{U} \equiv \text{hull}\{\vec{q} + \vec{u} : \vec{q} \in \mathring{Q}, \vec{u} \in \mathring{U}\}$
- $Q_{\vec{w}}(s,a) \equiv \max_{\vec{q} \in \mathring{Q}(s,a)} \vec{w} \cdot \vec{q}$

Convex hull value iteration algorithm

Initialize $\overset{\circ}{Q}(s,a)$ arbitrarily $\forall s, a$
while not converged **do**
  for all $s \in S$, $a \in A$ **do**
    $\overset{\circ}{Q}(s,a) \leftarrow \mathbb{E}[\overrightarrow{r}(s,a)$
      $+\gamma \,\text{hull} \bigcup_{a'} \overset{\circ}{Q}(s',a') | s, a]$
  end for
end while

return $\overset{\circ}{Q}$

Вопросы???


Hurricane

References

- *A Survey of Multi-Objective Sequential Decision-Making*
- *Multi-Objective Reinforcement Learning using Sets of Pareto Dominating Policies*
- *Learning All Optimal Policies with Multiple Criteria*
- *Constructing Stochastic Mixture Policies for Episodic Multiobjective Reinforcement Learning Tasks*


вжух
и конец презентации
memesmix.net