Appendix: Learning Fair Policies in Multi-Objective Deep Reinforcement Learning with Average and Discounted Rewards

A. Proofs of Theoretical Discussion

For better legibility, we first recall the equations and results that we need for our proofs.

$$\mathrm{GGF}_{\boldsymbol{w}}(\boldsymbol{v}) = \min_{\sigma \in \mathbb{S}_D} \boldsymbol{w}_\sigma^\intercal \boldsymbol{v} \tag{8}$$

where $\mathbb{S}_D$ is the symmetric group of degree D (i.e., set of permutations over $\{1, \dots, D\}$), σ is a permutation, and $\boldsymbol{w}_\sigma = (\boldsymbol{w}_{\sigma(1)}, \dots, \boldsymbol{w}_{\sigma(D)})$.

Theorem 3.5. *(Lamond & Puterman, 1989) For any MDP, any stationary policy π, and any $\gamma \in (\frac{\sigma(\boldsymbol{H}_{\boldsymbol{P}_\pi})}{\sigma(\boldsymbol{H}_{\boldsymbol{P}_\pi})+1}, 1)$,*

$$\boldsymbol{v}_\pi = \frac{1}{1-\gamma}\boldsymbol{g}_\pi + \frac{1}{\gamma}\sum_{n=0}^{\infty}\left(\frac{\gamma-1}{\gamma}\right)^n \boldsymbol{H}_{\boldsymbol{P}_\pi}^{n+1}\boldsymbol{r}_\pi \tag{10}$$

where $\boldsymbol{H}_{\boldsymbol{P}_\pi}$ is the Drazin inverse of $\boldsymbol{I} - \boldsymbol{P}_\pi$, which is given by $(\boldsymbol{I} - \boldsymbol{P}_\pi + \boldsymbol{P}_\pi^*)^{-1}(\boldsymbol{I} - \boldsymbol{P}_\pi^*)$, $\boldsymbol{P}_\pi^*$ is the Cesàro-limit of $\boldsymbol{P}_\pi^n$ for $n \to \infty$, and $\sigma(\boldsymbol{H}_{\boldsymbol{P}_\pi})$ is the spectral radius of matrix $\boldsymbol{H}_{\boldsymbol{P}_\pi}$.

We now present our proofs.

Lemma 3.8. *For any MOMDP, Problem (9) admits a solution that is a stationary stochastic Markov policy.*

Proof. For discounted rewards, a straightforward adaptation of the proof of Theorem 3.1 in (Altman, 1999) shows that the discounted occupation distribution of any policy can be obtained with that of a stationary stochastic Markov policy. As the evaluation of policies with GGF is completely determined by their discounted occupation distributions, the GGF of any policy can be obtained with that of a stationary stochastic Markov policy.

The situation is a bit more complicated for average rewards. We recall two results for single-objective MDPs that can straightforwardly be extended to multi-objective MDPs. Lemma 2.6 of Kallenberg (2003) states that:

$$\lim_{\gamma \uparrow 1}(1-\gamma)\boldsymbol{V}_\pi \geq \boldsymbol{G}_\pi \tag{11}$$

for any policy π. However, for such π, there exists a stationary policy π^+ such that for all γ close to one, $(1-\gamma)\boldsymbol{V}_{\pi^+} \geq (1-\gamma)\boldsymbol{V}_\pi$. Moreover, Corollary 2.5 of (Kallenberg, 2003) states that Inequality (11) becomes an equality for stationary policies. Therefore, the set of Pareto-optimal gains can be obtained using only stationary policies. □

Lemma 3.9. *For any weakly-communicating MOMDP, the GGF-average problem admits a solution that is a stationary stochastic Markov policy with constant gain.*

Proof. We start by recalling the following property of weakly communicating MDP. By Proposition 8.3.1 in (Puterman, 1994), an MDP is weakly communicating if and only if there exists a stationary stochastic policy $\overline{\pi}$ which induces a Markov chain with a single closed irreducible class $\mathcal{C}$ and a set of states $\mathcal{T}$ that is transient under all stationary policies.

Lemma 3.1 shows that we can focus on stationary stochastic Markov policies. Assume by contradiction that all such policies that are solutions of (9) with average rewards are such that their gains are not constant. Let π_1^* be such a policy. By assumption, there exist two states s_1 and s_2 such that $\mathrm{GGF}_{\boldsymbol{w}}(\boldsymbol{\mu}_{\pi_1^*,s_1}) > \mathrm{GGF}_{\boldsymbol{w}}(\boldsymbol{\mu}_{\pi_1^*,s_2})$. Denote $\mathcal{C}_{s_1}$ the set of states containing s_1 that is closed, irreducible, and recurrent with respect to π_1^*. We have necessarily $\mathcal{C}_{s_1} \subset \mathcal{C}$.

By the previous property of weakly-communicating MDPs, we can define a new policy π^+ such that $\forall s \in \mathcal{C}_{s_1}, \pi^+$ makes the same choices as π_1^*. For all the other states, π^+ makes the same choices as $\overline{\pi}$.

By definition, π^+ induces a Markov chain with a single closed irreducible class $\mathcal{C}_{s_1}$ and a set of transient states $\mathcal{S}\backslash\mathcal{C}_{s_1}$. The gain of policy π^+ is therefore constant and equal to $\boldsymbol{\mu}_{\pi_1^*,s_1}$, which contradicts our previous assumption. Therefore, the lemma holds. □

In the next theorem, π_γ^* is GGF-γ-optimal and π_1^* is GGF-average-optimal.

Theorem 3.6. *For any weakly-communicating MOMDP and any* $\gamma \in (\max(\frac{\sigma(\boldsymbol{H}_{\boldsymbol{P}_{\pi_\gamma^*}})}{\sigma(\boldsymbol{H}_{\boldsymbol{P}_{\pi_\gamma^*}})+1}, \frac{\sigma(\boldsymbol{H}_{\boldsymbol{P}_{\pi_1^*}})}{\sigma(\boldsymbol{H}_{\boldsymbol{P}_{\pi_1^*}})+1}), 1)$,

$$\mathrm{GGF}_{\boldsymbol{w}}(\boldsymbol{\mu}_{\pi_\gamma^*}) \geq \mathrm{GGF}_{\boldsymbol{w}}(\boldsymbol{\mu}_{\pi_1^*}) - \overline{\boldsymbol{R}}(1-\gamma)\left(\rho(\gamma, \sigma(\boldsymbol{H}_{\boldsymbol{P}_{\pi_1^*}})) + \rho(\gamma, \sigma(\boldsymbol{H}_{\boldsymbol{P}_{\pi_\gamma^*}}))\right)$$

where $\overline{\boldsymbol{R}} = \max_\pi \|\boldsymbol{R}_\pi\|_1$, $\sigma(\boldsymbol{M})$ *is the spectral radius of matrix* $\boldsymbol{M}$, *and* $\rho(\gamma, \sigma) = \frac{\sigma}{\gamma-(1-\gamma)\sigma}$.

Proof. We start with some notations. For any permutation σ, let $\mathcal{M}^\sigma$ be the MDP obtained from the initial MOMDP with the reward function defined by $\tilde{\boldsymbol{r}}_{a,s}^\sigma = \boldsymbol{w}_\sigma^\intercal \boldsymbol{R}_{a,s}$. Naturally, the MDP and MOMDP have the same policies. An element (e.g., optimal policy, value function, gain) corresponding specifically to $\mathcal{M}^\sigma$ will be marked with the tilde sign and exponent σ, e.g., $\tilde{\boldsymbol{v}}_\pi^\sigma$ (resp. $\tilde{\boldsymbol{g}}_\pi^\sigma$) is the value function (resp. gain) of policy π in $\mathcal{M}^\sigma$.

By (8), there exists σ such that $\mathrm{GGF}_{\boldsymbol{w}}(\boldsymbol{\mu}_{\pi_\gamma^*}) = \boldsymbol{w}_\sigma^\intercal \boldsymbol{\mu}_{\pi_\gamma^*}$. We now make two observations regarding π_γ^* and π_1^*. Regarding π_γ^*, we have:

$$\begin{aligned}
\boldsymbol{w}_\sigma^\intercal(\boldsymbol{d}_0\boldsymbol{V}_{\pi_\gamma^*})^\intercal &= \boldsymbol{d}_0\tilde{\boldsymbol{v}}_{\pi_\gamma^*}^\sigma && (12)\\
&\geq \mathrm{GGF}_{\boldsymbol{w}}\big((\boldsymbol{d}_0\boldsymbol{V}_{\pi_\gamma^*})^\intercal\big) && (13)\\
&\geq \mathrm{GGF}_{\boldsymbol{w}}\big((\boldsymbol{d}_0\boldsymbol{V}_{\pi_1^*})^\intercal\big) && (14)\\
&= \boldsymbol{w}_{\sigma'}^\intercal(\boldsymbol{d}_0\boldsymbol{V}_{\pi_1^*})^\intercal && (15)\\
&= \boldsymbol{d}_0\tilde{\boldsymbol{v}}_{\pi_1^*}^{\sigma'} && (16)
\end{aligned}$$

where (12) and (16) are obtained by linearity, (13) holds by (8), (14) is true by definition of the two policies, and (15) holds for some σ'. Regarding π_1^*, we have:

$$\mathrm{GGF}_{\boldsymbol{w}}(\boldsymbol{\mu}_{\pi_1^*}) \leq \boldsymbol{w}_{\sigma'}^\intercal \boldsymbol{\mu}_{\pi_1^*} = \boldsymbol{d}_0\tilde{\boldsymbol{g}}_{\pi_1^*}^{\sigma'} \tag{17}$$

where the inequality comes from (8) and the equality is obtained by linearity.

Using the first observation, we obtain:

$$\begin{aligned}
\boldsymbol{d}_0\tilde{\boldsymbol{g}}_{\pi_\gamma^*}^\sigma &= \boldsymbol{d}_0\big((1-\gamma)\tilde{\boldsymbol{v}}_{\pi_\gamma^*}^\sigma + \sum_{n=1}^\infty \left(\frac{\gamma-1}{\gamma}\right)^n \boldsymbol{H}_{\boldsymbol{P}_{\pi_\gamma^*}}^n \tilde{\boldsymbol{r}}_{\pi_\gamma^*}^\sigma\big) && (18)\\
&\geq (1-\gamma)\boldsymbol{d}_0\tilde{\boldsymbol{v}}_{\pi_1^*}^{\sigma'} + \sum_{n=1}^\infty \boldsymbol{d}_0\left(\frac{\gamma-1}{\gamma}\right)^n \boldsymbol{H}_{\boldsymbol{P}_{\pi_\gamma^*}}^n \tilde{\boldsymbol{r}}_{\pi_\gamma^*}^\sigma && (19)\\
&= \boldsymbol{d}_0\tilde{\boldsymbol{g}}_{\pi_1^*}^{\sigma'} - \sum_{n=1}^\infty \left(\frac{\gamma-1}{\gamma}\right)^n \boldsymbol{d}_0\boldsymbol{H}_{\boldsymbol{P}_{\pi_1^*}}^n \tilde{\boldsymbol{r}}_{\pi_1^*}^{\sigma'} + \sum_{n=1}^\infty \left(\frac{\gamma-1}{\gamma}\right)^n \boldsymbol{d}_0\boldsymbol{H}_{\boldsymbol{P}_{\pi_\gamma^*}}^n \tilde{\boldsymbol{r}}_{\pi_\gamma^*}^\sigma && (20)\\
&\geq \mathrm{GGF}_{\boldsymbol{w}}(\boldsymbol{\mu}_{\pi_1^*}) - \sum_{n=1}^\infty \left(\frac{\gamma-1}{\gamma}\right)^n \boldsymbol{d}_0\boldsymbol{H}_{\boldsymbol{P}_{\pi_1^*}}^n \tilde{\boldsymbol{r}}_{\pi_1^*}^{\sigma'} + \sum_{n=1}^\infty \left(\frac{\gamma-1}{\gamma}\right)^n \boldsymbol{d}_0\boldsymbol{H}_{\boldsymbol{P}_{\pi_\gamma^*}}^n \tilde{\boldsymbol{r}}_{\pi_\gamma^*}^\sigma && (21)
\end{aligned}$$

where (18) is obtained from (10) applied to $\tilde{\pi}_\gamma^*$ and rearranging terms, (19) comes from (13), (20) is obtained from (10) applied to π_1^*, and (21) holds because the gain of π_1^* is constant by Lemma 3.3.

We now show how the second term in the right-hand side of the inequality can be bounded:

$$\sum_{n=1}^{\infty}\left(\frac{\gamma-1}{\gamma}\right)^n \boldsymbol{d}_0 \boldsymbol{H}^n_{\boldsymbol{P}_{\pi_1^*}} \tilde{\boldsymbol{r}}^{\sigma'}_{\pi_1^*} \leq \sum_{n=1}^{\infty}\left(\frac{1-\gamma}{\gamma}\right)^n \|\boldsymbol{d}_0\|\|\boldsymbol{H}^n_{\boldsymbol{P}_{\pi_1^*}}\|_2\|\tilde{\boldsymbol{r}}^{\sigma'}_{\pi_1^*}\| \tag{22}$$

$$\leq \sum_{n=1}^{\infty}\left(\frac{1-\gamma}{\gamma}\right)^n \|\boldsymbol{H}_{\boldsymbol{P}_{\pi_1^*}}\|_2^n\|\tilde{\boldsymbol{r}}^{\sigma'}_{\pi_1^*}\| \tag{23}$$

$$\leq \sum_{n=1}^{\infty}\left(\frac{1-\gamma}{\gamma}\right)^n \|\boldsymbol{H}_{\boldsymbol{P}_{\pi_1^*}}\|_2^n\overline{\boldsymbol{R}} \tag{24}$$

$$= \sum_{n=1}^{\infty}\left(\frac{1-\gamma}{\gamma}\right)^n \sigma(\boldsymbol{H}_{\boldsymbol{P}_{\pi_1^*}})^n\overline{\boldsymbol{R}}$$

$$= \overline{\boldsymbol{R}}\left(\frac{1}{1-\frac{1-\gamma}{\gamma}\sigma(\boldsymbol{H}_{\boldsymbol{P}_{\pi_1^*}})}-1\right) \tag{25}$$

$$= \overline{\boldsymbol{R}}\frac{(1-\gamma)\sigma(\boldsymbol{H}_{\boldsymbol{P}_{\pi_1^*}})}{\gamma-(1-\gamma)\sigma(\boldsymbol{H}_{\boldsymbol{P}_{\pi_1^*}})}$$

where (22) is obtained by applying the Cauchy-Schwartz inequality, (23) uses $\|\boldsymbol{d}_0\| \leq 1$, (24) uses $\|\boldsymbol{w}\|_1 = 1$, and (25) holds by assumption ($\gamma > \sigma(\boldsymbol{H}_{\boldsymbol{P}_{\pi_1^*}})/(\sigma(\boldsymbol{H}_{\boldsymbol{P}_{\pi_1^*}})+1)$).

A similar bound can be found for the third term. Finally, the result follows by plugging the two bounds in (21) and the fact that $\mathrm{GGF}_{\boldsymbol{w}}(\boldsymbol{\mu}_{\pi^*_\gamma}) = \boldsymbol{w}_\sigma^\intercal \boldsymbol{\mu}_{\pi^*_\gamma} = \boldsymbol{d}_0 \tilde{\boldsymbol{g}}^{\sigma}_{\pi^*_\gamma}$, which holds by linearity. □

B. Descriptions of Experimental Domains

B.1. Conservation of two endangered species

This domain is based on the model introduced by Chadès et al. (2012) that describes the interaction of two endangered species, sea otters and its prey, northern abalones. In this problem, interventions can be taken in order to maintain the populations of the two species at relatively balanced levels.

The environment for this problem is in fact a partially observable MOMDP and similarly to (Chadès et al., 2012), we solve it as an MOMDP. The model can be summarized as follows[1] (motivation and more explanation can be found in (Chadès et al., 2012)):

- At time step t, a state consists of N_t^O (the population number of sea otters), $\boldsymbol{N}_t^A$ (a 10-dimensional vector indexed from 4 to 13, where $\boldsymbol{N}_{i,t}^A$ is the population number of abalones for age group i), and a 10×10 matrix representing the survival rate of abalones for different age and living area. In the model, 10 age groups are considered from 4 to 13: the enrollment age starts from age 4 and all the ages greater than 13 are pooled together into the 13 age group. The initial state is fixed to some stable abalone population numbers.
- An observation is defined as a pair $(n_t^O, n_t^A) \in \{1, ..., 21\} \times \{1, ..., 39\}$ where n_t^O (respectively n_t^A) is a discretization of N_t^O to represent sea otters (respectively $\sum_i \boldsymbol{N}_{i,t}^A$ to represent abalones).
- Five management actions are considered: *do nothing*, *introduce sea otters*, *enforce abalone antipoaching measures*, *control sea otters*, and *one-half antipoaching and one-half control sea otters*.
- The transition function is based on the population growth models of the two species taking into account factors such as poaching and predation (for abalones) or oil spills (for sea otters). The next state is computed in the following order : 1) apply abalone and sea otter growth models independently of the action, 2) potentially apply culling of sea otters according to the action, 3) remove abalone because of predation, 4) remove abalone because of poaching according to the action. The interaction between sea otters and abalone is modeled with a linear function response.
- The reward after performing action a in state s_t is a vector consisting of two components:

$$\boldsymbol{R}_{a,s_t} = \begin{bmatrix} JR_{so}(N_t^O) \\ JR_{aba}(\sum_{i=3}^{14} \boldsymbol{N}_{i,t}^A) \end{bmatrix}$$

[1] For simplicity, we use similar notations as in (Chadès et al., 2012) since there is no much risk of confusion with those used in our main paper.

where $s_t = (N_t, d_t^3, \ldots, d_t^{14})$, and the two functions JR_{so} and JR_{aba} are introduced to make a population number of sea otters and a density of northern abalones commensurable. Note that rewards do not depend on actions here. In (Chadès et al., 2012), a scalar reward was defined as the minimum of those two components in order to balance the two objectives.

B.2. Traffic Light Control

We also evaluate our method in the classic traffic light control problem. While the usual approach to this problem consists in minimizing the expected waiting times averaged over all lanes, we consider the expected waiting times of the four directions (north, east, south, or west) at the intersection separately. We use Simulation of Urban MObility (SUMO)[2] to simulate a single eight-lane intersection (see Figure 11). Depending on intersections, different numbers of traffic light phases can be considered. A traffic light phase specifies which lanes have the green light. We assume here that there are four phases NSL, $NSSR$, EWL, and $EWSR$. Phase NSL (North-South Left) corresponds to the case where the green light is given to cars in the left lanes of the roads coming from the north and south. The cars can only turn left in this phase. Phase $NSSR$ (North-South Straight and Right) allows cars in the right lanes for the north-south axis to go straight or turn right. Phases EWL and $EWSR$ are defined similarly for the east-west axis.

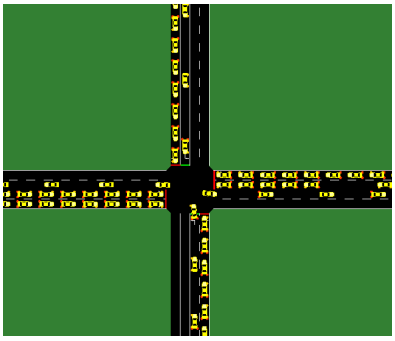

Figure 11. SUMO traffic controller simulation.

We formulate the MOMDP for this domain as follows:

- A state is a 20-dimensional vector containing the current traffic light phase (4 phases represented with a one-hot encoding) and for each lane, its total waiting time and density of cars stopped at the intersection (8×2).
- An action corresponds to a traffic light phase. Therefore, $\mathcal{A} = \{NSL, NSSR, EWL, EWSR\}$.
- The transition function depends on the current traffic light phase, how cars drive through the intersection, and how new traffic is generated.
  The duration of each phase is fixed (which corresponds to one time step in the RL problem). The green light time for each phase is 10 seconds while the yellow time is 4 (if there is a change to a new phase). For simplicity, all the vehicles have the same characteristics (e.g., car speed, acceleration, length) that are provided by default in SUMO. The phase duration and the car characteristics determine how many waiting cars can drive through the intersection.
  At each time step, for each lane, new vehicles enter the intersection according to a fixed Bernoulli distribution in each episode. The probabilities used for each lane are provided in Figure 12. They simulate a heavy traffic intersection.
- A reward is a vector of 4 components corresponding to the waiting times of each direction:

$$\boldsymbol{R}_{a,s} = \begin{bmatrix} -\sum_{j=1}^{2} W_j^1(s) \\ -\sum_{j=1}^{2} W_j^2(s) \\ -\sum_{j=1}^{2} W_j^3(s) \\ -\sum_{j=1}^{2} W_j^4(s) \end{bmatrix}$$

  where $W_j^i(s)$ is the total waiting time of all the cars of the i^{th} direction and j^{th} lane. The standard approach to this problem would define the scalar reward as the sum of those components.

[2] https://github.com/eclipse/sumo

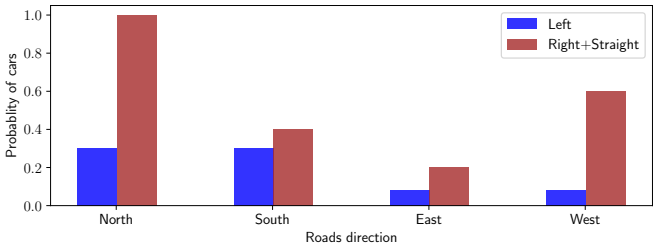

Figure 12. Probabilities of cars entering in each lane.

B.3. Data Center Traffic Control

In the Data Center (DC) traffic congestion control problem (Ruffy et al., 2019), a centralized controller manages a computer network that is shared by a certain number of hosts in order to optimize the bandwidths of each host. For the network topology, a fat-tree topology (see Figure 13 is considered, which has $D = 16$ hosts, 20 switches with $n = 4$ ports each, which leads to a total of 80 queues. For the experiments, we used Mininet[3] to simulate the network with the fat-tree topology using UDP as the underlying transport protocol and goben[4] to generate traffic and monitor/collect network information. In (Ruffy et al., 2019), a reward function was designed such that an RL agent would learn to maximize a sum of host bandwidths penalized by queue lengths (in order to avoid switch bufferbloats). Here, we instead aim at maximizing separately the bandwidth of each host penalized by queue lengths, while ensuring fairness.

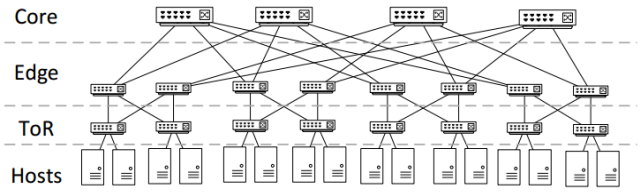

Figure 13. Network with a fat-tree topology from (Ruffy et al., 2019).

The MOMDP for this problem can be summarized as follows (for motivation and more explanation check (Ruffy et al., 2019)):

- A state is composed of a $n \times m$ matrix that stores the information statistics from the transport and lower layers, where n is the number of ports in a switch and m is the number of network features (i.e., the queue length, the derivative over time of the queue length, the number of packet drops and the queue length above the limit). A state also contains information about the current bandwidth allocation to the D hosts. A bandwidth is a value in $[0, 10]$.
- An action is a D-dimensional vector of bandwidth allocation to the hosts.
- Network traffic between hosts is generated according to an input file (see (Ruffy et al., 2019)).
- The D-dimensional reward vector is defined as follows:

$$\boldsymbol{R}_{\boldsymbol{a},s} = \boldsymbol{a} - 2 * \boldsymbol{a} * \max_i q_i(s)$$

where $\boldsymbol{a}$ is the vector action that represents the bandwidth allocation and $q_i(s)$ represents the i-th queue length. This definition is adapted from (Ruffy et al., 2019), which uses the average of the previous components to define a scalar reward.

[3] https://github.com/mininet/mininet
[4] https://github.com/udhos/goben

C. Hyperparameters

For the sake of reproducibility, all the hyperparameters for all the environments are reported in Tables 1, 3 and 2. In those tables, subscripts "sc", "tl", "dc" stand for our three experimental domains, respectively, species conservation, traffic lights control, and data center traffic control.

Table 1. Set of hyperparameters used during training with PPO and GGF-PPO

HYPERPARAMETER	VALUES$_{PPO}$	VALUES$_{GGF-PPO}$
γ	$0.99_{sc,tl,dc}$	$0.99_{sc,tl,dc}$
LEARNING RATE	$0.001_{sc}, 0.0005_{tl}, 0.0001_{dc}$	$0.00005_{sc,dc}, 0.0005_{tl}$
N_ENVS	$10_{sc,tl}, 1_{dc}$	$10_{sc,tl}, 1_{dc}$
N_STEPS PER UPDATE	$128_{sc,tl,dc}$	$128_{sc,tl,dc}$
CLIPRANGE	$0.2_{tl}, 0.1_{sc}, 0.5_{dc}$	$0.2_{tl}, 0.1_{sc}, 0.5_{dc}$
ACTOR NETWORK	$64 * 64_{sc,tl,dc}$	$64 * 64_{sc,tl,dc}$
CRITIC NETWORK	$64 * 64_{sc,tl,dc}$	$64 * 64_{sc,tl,dc}$
NETWORKS HIDDEN ACTIVATION	$TanH_{sc,tl,dc}$	$TanH_{sc,tl,dc}$
NETWORKS OUTPUT ACTIVATION	$Linear_{sc,tl,dc}$	$Linear_{sc,tl,dc}$
OPTIMIZER	$Adam_{sc,tl,dc}$	$Adam_{sc,tl,dc}$
ADAM EPSILON	$1e^{-5}_{sc,tl,dc}$	$1e^{-5}_{sc,tl,dc}$
ENT_COEF	$0.01_{sc,tl,dc}$	$0.01_{sc,tl,dc}$
VF_COEF	$0.5_{sc,tl,dc}$	$0.5_{sc,tl,dc}$

Table 2. Set of hyperparameters used during training with A2C and GGF-A2C

HYPERPARAMETER	VALUES$_{A2C}$	VALUES$_{GGF-A2C}$
γ	$0.99_{sc,tl,dc}$	$0.99_{sc,tl,dc}$
LEARNING RATE	$0.0001_{sc,tl,dc}$	$0.0001_{sc,tl}, 0.0005_{dc},$
N_ENVS	$10_{sc,tl}, 1_{dc}$	$10_{sc,tl}, 1_{dc}$
N_STEPS PER UPDATE	$10_{sc}, 5_{tl,dc}$	$30_{sc,tl,dc}$
ACTOR NETWORK	$64 * 64_{sc,tl,dc}$	$64 * 64_{sc,tl,dc}$
CRITIC NETWORK	$64 * 64_{sc,tl,dc}$	$64 * 64_{sc,tl,dc}$
NETWORKS HIDDEN ACTIVATION	$TanH_{sc,tl,dc}$	$TanH_{sc,tl,dc}$
NETWORKS OUTPUT ACTIVATION	$Linear_{sc,tl,dc}$	$Linear_{sc,tl,dc}$
OPTIMIZER	$RMSprop_{sc,tl,dc}$	$RMSprop_{sc,tl,dc}$
RMSPROP EPSILON	$1e^{-5}_{sc,tl,dc}$	$1e^{-5}_{sc,tl,dc}$
RMSPROP ALPHA	$0.99_{sc,tl,dc}$	$0.99_{sc,tl,dc}$
VALUE_FUNC_COEF	$0.25_{sc,tl,dc}$	$0.25_{sc,tl,dc}$
ENTROPY _COEF	$0.01_{sc,tl,dc}$	$0.01_{sc,tl,dc}$

Table 3. Set of hyperparameters used during training with DQN and GGF-DQN

HYPERPARAMETER	VALUES$_{DQN}$	VALUES$_{GGF-DQN}$
γ	$0.99_{sc,tl}$	$0.99_{sc,tl}$
LEARNING RATE	$0.0001_{sc}, 0.0005_{tl}$	0.005_{sc}, 0.0005_{tl}
BATCH_SIZE	64_{sc}, 128_{tl}	$128_{sc,tl}$
Q NETWORK	$64 * 64_{sc,tl}$	$64 * 64_{sc,tl}$
NETWORK HIDDEN ACTIVATION	$ReLU_{sc,tl}$	$ReLU_{sc,tl}$
NETWORK OUTPUT ACTIVATION	$ReLU_{sc,tl}$	$ReLU_{sc,tl}$
BUFFER_SIZE	$50000_{sc,tl}$	$50000_{sc,tl}$
EXPLORATION_FRACTION	$0.1_{sc,tl}$	$0.1_{sc,tl}$
PRIORITIZED_REPLAY	$False_{sc,tl}$	$False_{sc,tl}$
EXPLORATION_FRACTION	$0.1_{sc,tl}$	$0.1_{sc,tl}$
TARGET_NETWORK_UPDATE_FREQ	$500_{sc,tl}$	$500_{sc,tl}$
DUELING	$False_{sc,tl}$	$False_{sc,tl}$

D. Additional Experimental Results

We present here additional experimental results that were not included in the main document.

D.1. Conservation of two endangered species

In the SC domain we also perform some weight analysis. As it is a small problem with two objectives, it is easy to perform those experiments and clearly show how our approach yields more balanced or fairer solutions. In addition to the experiments that we have shown in the main paper, we also performed experiments where the GGF weights are decreasing faster. Concretely, GGF coefficients were defined as $\boldsymbol{w}_i = \frac{1}{2^i}$ from 0 to $D-1$, while in this set of experiments it is defined as $\boldsymbol{w}_i = \frac{1}{10^i}$ from 0 to $D-1$. Similar conclusions, which we detail next, can be drawn with both sets of weights.

Figure 14 shows the distributions of GGF score for the policies learned by DQN, A2C, PPO and their GGF counterparts. As expected, all the three GGF algorithms have higher GGF score than their original algorithms. Higher GGF scores means the solution is more balanced which can be validated from Figure 15.

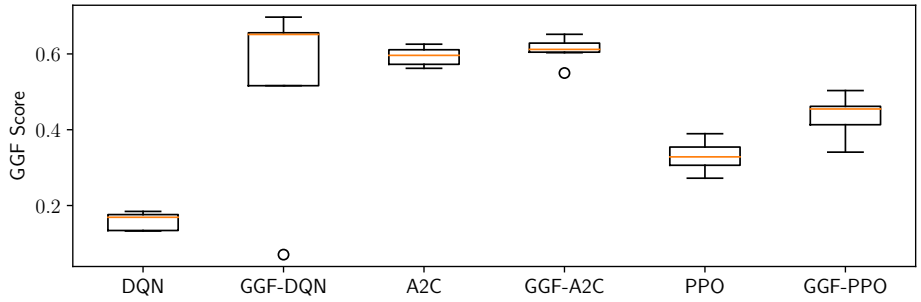

Figure 14. GGF scores of DQN, A2C, PPO and their GGF algorithms with $\boldsymbol{w}_i = \frac{1}{10^i}$ during the testing phase in the SC domain.

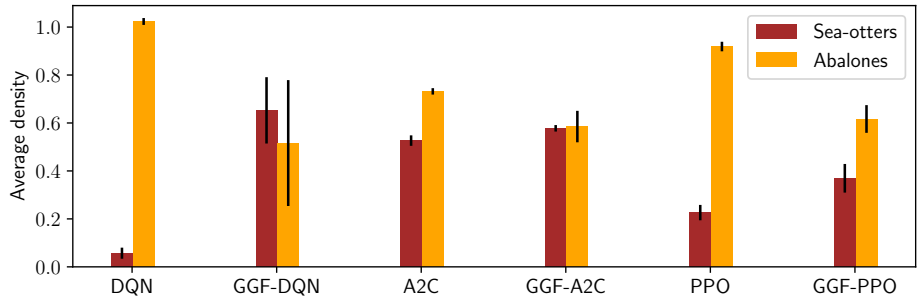

Figure 15. Individual densities for DQN, A2C, PPO and their GGF versions with $\boldsymbol{w}_i = \frac{1}{10^i}$ during the testing phase in the SC domain.

Similar to the case of $\boldsymbol{w}_i = \frac{1}{2^i}$, we also compared those GGF algorithms with faster decreasing weights in terms of their CV, minimum and maximum of densities (Figure 16). As explained before, standard RL algorithms generate unequal distributions of rewards while our adapted versions of DQN, A2C and PPO generate more balanced solutions. Again the CV of GGF algorithms is lower than their original algorithms which shows the less variations in their objectives.

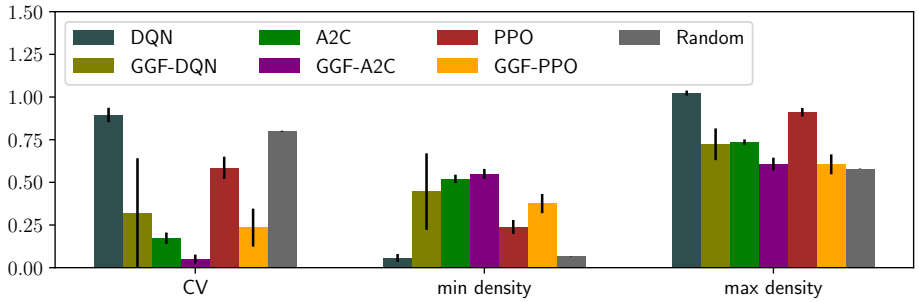

Figure 16. The performances of different RL algorithms and their GGF versions with $\boldsymbol{w}_i = \frac{1}{10^i}$ in the SC domain.

D.2. Traffic Light Control

To clearly demonstrate that our proposition yields more equitable solutions, we comapre PPO, A2C, DQN and their GGF counterparts in terms of waiting times per direction, which were estimated after training. As shown in Figure 19, the waiting times achieved by GGF-PPO is more balanced. In terms of average waiting times, DQN and GGF-DQN did not work very well. However, from the results in Figure 17, it is clear that the GGF version of DQN is fairer than the standard DQN.

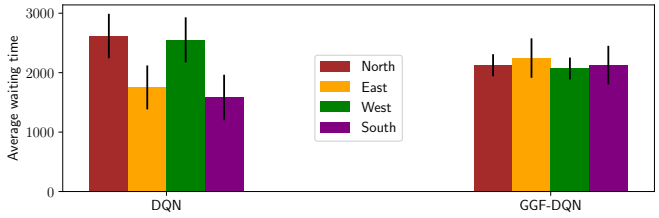

Figure 17. Individual average waiting times of DQN and GGF-DQN during the testing phase.

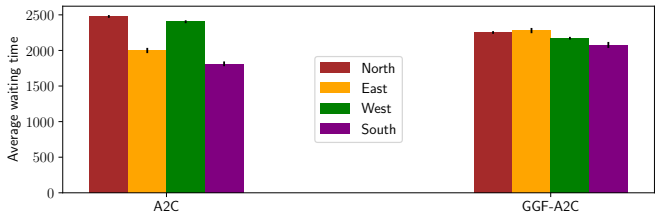

Figure 18. Individual average waiting times of A2C and GGF-A2C during the testing phase.

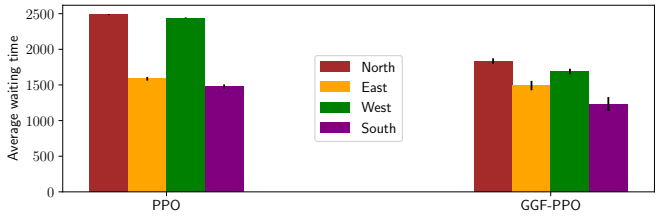

Figure 19. Individual average waiting times of PPO and GGF-PPO during the testing phase.

We also ran some additional experiments on a non-stationary environment. For this experiment, the traffic generation is not fixed and changes during the day. The problem becomes more challenging, but is much closer to a real environment.

There are many ways to add the variance in the traffic patterns. We defined 4 distributions (see Figure 20 for one lane) corresponding to four different periods of a day: morning, afternoon, evening, and night.

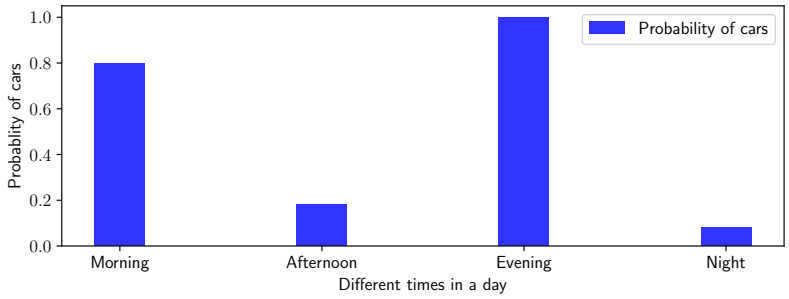

Figure 20. Probabilities of car entering at different times in a day for one lane.

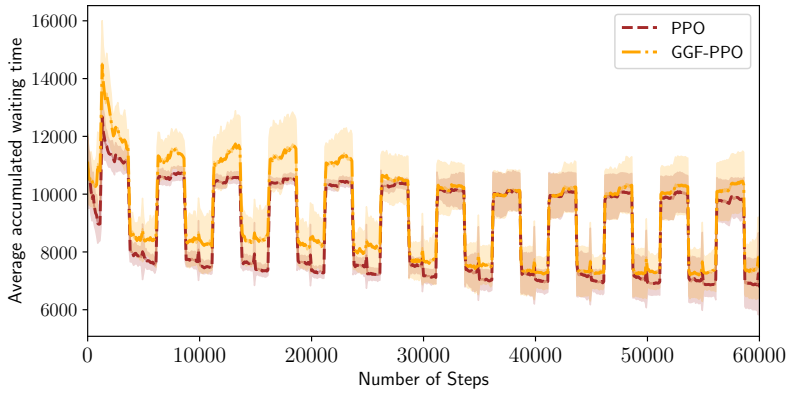

Figure 21. Average waiting times of PPO and GGF-PPO during learning phase in the non-stationary TL domain.

Figure 21 visualizes the average accumulated waiting time of PPO and GGF-PPO on this non-stationary environment. As expected, GGF-PPO performs worse than PPO on that metric. The ups and downs represents the different times in a day. However, GGF-PPO achieves a much higher GGF score than PPO (see Figure 22).

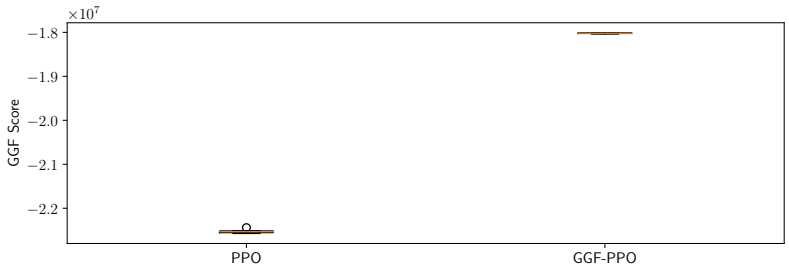

Figure 22. GGF scores of PPO and GGF-PPO during the testing phase in the non-stationary TL domain.

D.3. Data Center Traffic Control

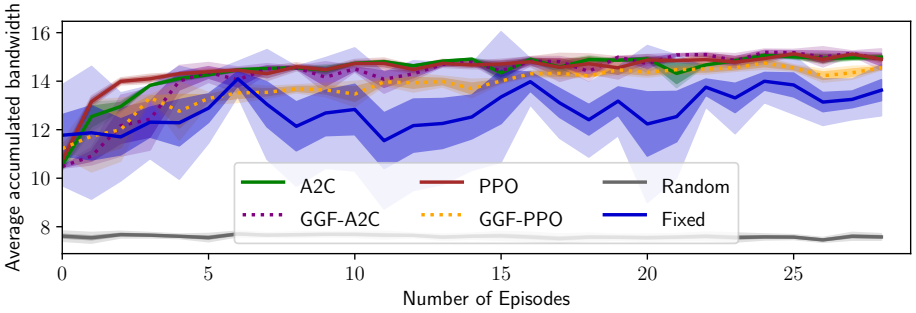

Figure 23. Episodic reward of PPO and GGF-PPO during the learning phase, and that of the Fixed and Random policies in the DC domain over 20 runs with different seeds.

Figure 23 illustrates the learning curves of PPO, A2C and their GGF counterparts in terms of episodic rewards. The performance of two policies (a fixed one and a random one) are added for comparison. The fixed policy always chooses the maximum bandwidth for each host. The random policy selects actions with a uniform distribution. We can see that PPO and GGF-PPO converge to a much higher reward than random and fixed policies. GGF-PPO's and GGF-A2C's average bandwidth in an episode is lower than PPO and A2C, this is because it is hard for a single policy to maximize rewards while ensuring fairness. However, GGF algorithms still performs better than the random and fixed policies and tries to get high rewards while allocating bandwidth equally to different hosts.

References

Altman, E. *Constrained Markov Decision Processes*. CRC Press, 1999.

Chadès, I., Curtis, J. M., and Martin, T. G. Setting realistic recovery targets for two interacting endangered species, sea otter and northern abalone. *Conservation Biology*, 26(6):1016–1025, 2012.

Kallenberg, L. Finite State and Action MDPs. In *Handbook of Markov Decision Processes*. 2003. doi: 10.1007/978-1-4615-0805-2_2.

Lamond, B. F. and Puterman, M. L. Generalized Inverses in Discrete Time Markov Decision Processes. *SIAM Journal on Matrix Analysis and Applications*, 10(1):118–134, jan 1989. ISSN 0895-4798. doi: 10.1137/0610009.

Puterman, M. *Markov decision processes: discrete stochastic dynamic programming*. Wiley, 1994.

Ruffy, F., Przystupa, M., and Beschastnikh, I. Iroko: A framework to prototype reinforcement learning for data center traffic control. In *Workshop on ML for Systems at NeurIPS*, 2019. URL http://arxiv.org/abs/1812.09975.