Exploring the behavioural spectrum with efficiency vs

Zafeiris Kokkinogenis [1,1], Margarida Silva [2], Jeremy Pitt [2], and Rosaldo Rossetti [2]

[1]University of Porto
[2]Affiliation not available

October 31, 2023

Abstract

The concept of fairness has been studied in philosophy and economics for thousands of years, so human actors in social systems have had plenty of time to "learn" what does, and does not, work. Yet, only recently. However, it is a relatively new question how software agents in a multi-agent system can use Reinforcement Learning models to develop an architecture that promotes equality or equity in the distribution of rewards to the agents within the system. Recent significant contributions have focused on optimising for efficiency based on the assumption that efficiency and fairness are opposites to be traded off against each other, but actually, the result of mixing fair and efficient policies is unknown in multi-agent reinforcement learning settings. In this work, we experiment with fair and efficient behaviours jointly, based on an extension of the state-of-the-art model in fairness SOTO that intertwines efficient and equitable recommendations. We analyse the fair versus efficient behavioural spectrum in the Matthew Effect and Traffic Light Control problems, finding some solutions that outperform the baseline SOTO and others that outperform a selfish baseline with comparable architectural design. We conclude it is possible to optimise for fairness and efficiency and this is important when computation of the reward distribution has to be paid for from the rewards themselves.

Exploring the behavioural spectrum with efficiency vs. fairness goals in Multi-Agent Reinforcement Learning

Margarida Silva[1*], **Zafeiris Kokkinogenis**[1†], **Jeremy Pitt**[2] and **Rosaldo J. F. Rossetti**[1]
[1]Artificial Intelligence and Computer Science Lab (LIACC), Department of Informatics Engineering, Faculty of Engineering, University of Porto, Porto, Portugal
[2]Department of Electrical and Electronic Engineering, Imperial College London, London
margarida.rpds@gmail.com, zafeiris.kokkinogenis@gmail.com, j.pitt@imperial.ac.uk, rossetti@fe.up.pt

Abstract

The concept of fairness has been studied in philosophy and economics for thousands of years, so human actors in social systems have had plenty of time to "learn" what does, and does not, work. Yet, only recently. However, it is a relatively new question how software agents in a multi-agent system can use Reinforcement Learning models to develop an architecture that promotes equality or equity in the distribution of rewards to the agents within the system. Recent significant contributions have focused on optimising for efficiency based on the assumption that efficiency and fairness are opposites to be traded off against each other, but actually, the result of mixing fair and efficient policies is unknown in multi-agent reinforcement learning settings. In this work, we experiment with fair and efficient behaviours jointly, based on an extension of the state-of-the-art model in fairness SOTO that intertwines efficient and equitable recommendations. We analyse the fair versus efficient behavioural spectrum in the Matthew Effect and Traffic Light Control problems, finding some solutions that outperform the baseline SOTO and others that outperform a selfish baseline with comparable architectural design. We conclude it is possible to optimise for fairness and efficiency.

1 Introduction

Throughout history, RL techniques have aimed to optimise the expected sum of rewards an agent gets for acting under its policy. More recently, fairness concerns have been brought into the Machine Learning literature, and fairness-aware line of algorithms have been emerging. In the multi-agent paradigm, some work attempts to optimise the equality in the distribution of rewards of the agents in the most efficient manner possible [Jiang and Lu, 2019; Zimmer *et al.*, 2021]. This literature approaches fairness as a goal for the individual agents' policies to optimise. However, there seems to be a gap in work considering fairness and efficiency holistically. Indeed, in a real-world scenario, it may be that neither the efficient nor the equality extreme goals are ideal, so it is important to study in-between solutions. The designer should be able to opt to sacrifice one of them, to a certain extent, for the other, and there is lack of literature to support such a decision. There is still no evidence of the outcome of mixing fair and efficient behaviours or even training these together in a MADRL system. While the relationship between fairness and efficiency is popularly seen as a trade-off [Pióro *et al.*, 2002], there is still a lack in evidence that is really the case in MADRL.

The main goal of this paper is to address the equality of rewards fairness issue in MADRL in an exploratory manner. By relaxing assumptions on which goal is intended for the system - efficiency or fairness - we aim to observe what solutions arise. We want to observe the impact of combining fair and efficient goals both in test and train.

We tackle this challenge by employing two main techniques. The first is heterogeneous testing, where the agents in the system act selfishly or fairly according to a probability, without updating their policies weights. This enables a direct mix in previously learned policies. The second is an extension to the SOTO model [Zimmer *et al.*, 2021] which enables a team-oriented policy to provide fair action insights to the self-oriented one. Because each policy recommends the other intertwined in this setting, we call this method Intertwined SOTO (I-SOTO). We also experiment with different settings for the training strategy that controls how SOTO's self- and team-oriented policies are trained. Doing so, we believe different solutions will be found in both extremes and heterogeneous intermediates from testing - specially in the I-SOTO case, where both policies share action recommendations. We develop our work under two main assumptions:

1. If selfish and fair policies are combined heterogeneously, a linear range of fair-efficient behaviours is generated
2. If SOTO's π^{IND} also receives recommendations from π^{SWF} - I-SOTO - it is possible to find solutions that are better in at least on of the goals (fairness or efficiency) without compromising the other.

This paper contributes to the state of the art on equality fairness in MADRL in a variety of manners. In a broad sense, our contribution mainly relies on presenting results for the ex-

*Contact Author
†Contact Author

ploratory attempts made in combining fair and efficient policies and training them jointly. The primary contributions of the paper: (1) we propose an extension that intertwines recommendations from the self- and team-oriented policies and assess it under different $\beta(e_r)$ training strategies; (2) we show that with I-SOTO it is possible to find solutions more fair and efficient than SOTO's fair and efficient baseline; (3) we provide an experimental set of results which may serve as support for a system's designer to choose which policy setup is most appropriate for their goals in terms of efficiency and fairness.

2 Related Work

As a social construct, fairness is inherently subjective [Lamertz, 2002]. Its notion has been extensively studied within various fields political science [Brams *et al.*, 1996] and economics [Moulin, 2003]. This led to the emergence of a variety of fairness considerations including impartiality, equity and equality, envy-freeness allocation, among others. Some of these were brought to other fields making use of applied mathematics such as Operations Research (OR), Artificial Intelligence (AI) and Machine Learning (ML).

With the recent increased presence of Machine Learning (ML) in real-life decision-making situations, fairness has also been gaining importance in such field [Mehrabi *et al.*, 2021].

More recently, the notion of fairness has been brought to MARL systems. A line of work focuses on applying equitable resource allocation [Luss, 2012] in the domain at study. Some approaches solve the problem with domain-specific knowledge towards solving a known issue in it[Elmalaki, 2021; Chen *et al.*, 2021]. Others consider resource allocation in a more general manner [Zhang and Shah, 2014] use the max-min egalitarian notion of social welfare, the lowest utility within the system. On another note [Claure *et al.*, 2019], tackle the multi-armed bandit domain by introducing constraints relative to the allocation process.

A second line of work focuses on including fairness concerns in the model functioning itself. [Siddique *et al.*, 2020] work on the multi-objective MDP problem. The approach taken was to make use of a social welfare function [Busa-Fekete *et al.*, 2017] as a way of ensuring each of these goals is being learned in a fair way, i.e., being given approximately equal opportunities to be learned. On another note, [Wang *et al.*, 2020] approach the multi-agent credit assignment problem using the notion of Shapley value, which approximates the impact of a single agent in a coalition of agents.

Finally, the line of work which we follow in this work, focuses on equality of the rewards between the set of agents within the MARL system. In competitive domains, [Hughes *et al.*, 2018] encode aversion for inequality, advantageous and disadvantageous, in the agents reward. While this work is successful in promoting cooperation in competitive environments, they do not make fairness considerations. In this work we focus on the cooperative MARL setting where the agents share the same system equality goal. However, making agents learn this is not trivial. Indeed, if the reward is set to be a global system property, all agents will receive the same reward signal which is not efficient. This problem is sometimes referred to as the *credit-assignment problem*.

To the best of our knowledge, there are only two approaches that attempt to work on this problem. Both of these seem to be approach this by developing architectures which combine insights of more than one policy. One of them, FEN [Jiang and Lu, 2019] is a hierarchical policy model consisting of a controller that ensembles several sub-policies. A key aspect is that only one of the sub-policies is trained with the traditional reward signal r_t while the remaining ones exploit an information-theoretic objective to explore alternative behaviours. The controller is optimised according to a *fair-efficient* reward $\hat{r}_t^i = \frac{\bar{u}_t/c}{\epsilon+\left|u_t^i/\bar{u}_t-1\right|}$. The results showed this model achieved fairer results than independent models in a variety of domains. The second one, SOTO [Zimmer *et al.*, 2021], provides agents with a self-oriented and a team-oriented policy. The first is trained for often in the beginning to provide efficient insights to the team-oriented policy. Results showed that it SOTO over-performed FEN both in fairness and efficiency.

There is indeed very little work approaching the reward equality problem in MADRL. In the existent literature, fairness is approached as an independent goal from efficiency.

3 Methodological Approach

Notation: we use calligraphic letters - $\mathcal{X}$ - to denote alphabet sets, upper-case letters - X - to denote random variables and constants, hats to denote approximations - $\hat{X}$ - and lower-case letters - x - to denote realizations. The methodology employed is an extension to SOTO [Zimmer *et al.*, 2021], so we dedicate section 3.1 to describing its intricacies.

3.1 SOTO

The SOTO architecture [Zimmer *et al.*, 2021], is, to the best of our knowledge, state of the art in the reward equality problem. It comprises a Self-Oriented Policy π^{IND} and Team-Oriented one π^{SWF}, trained for the selfish and the fair goals respectively. It is designed for the Dec-POMDP framework [Oliehoek and Amato, 2016], such that each of these policies is independently trained for each agent and receives the observations of the world as input. Moreover, the team-oriented policy additionally receives $\mathbf{J} = \{J_i(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}}\left[\sum_t \gamma^t r_{i,t}\right]\}$ - providing information on the wealth of other agents - and $\pi^{\text{IND}}(a|o)$ - the forwarded output self-oriented policy as an efficient recommendation. While the first inform the wealth state of other agents, the latter provides a self-oriented recommendation for efficiency to the team-oriented policy.

In the SOTO training procedure [1], at each batch of steps, throughout episodes, a policy is chosen - either self- or team-oriented - according to the value of a β variable. The agents act according to such policy for the length of the batch in time steps and, by the end of it, updates the corresponding policy's weights. As such, the training of each policy depends on the evolution of β throughout episodes $\beta(e_r)$, where $e_r = \frac{e}{E}$ is the episode ratio. The function chosen by the authors is

[1] For an in-depth explanation with pseudo-code we defer the reader to the original paper [Zimmer *et al.*, 2021]

$\beta(e_r) = \max(1 - 2e_r, 0)$, where e is the episode number and E is the number of the last episode.

Both policies are trained with Policy Gradients algorithms. However, the team-oriented policy is trained with a different advantage value than the self-oriented one - it is based on a social welfare function (SWF) with regards to the distribution of cumulative rewards $\mathbf{J}$. To choose which SWF to utilise, the authors respect three principles: Impartiality, Equity and Pareto-efficiency [Weng, 2019].The present two families of viable functions according to these criteria and end up utilising the Generalised Gini Function [Weymark, 1981] (GGF) $G_{\mathbf{w}}(\mathbf{u}) = \sum_{k\in[\mathcal{D}]} w_k u_k^{\uparrow}$ and the α-fairness [Mo and Walrand, 2000] SWF$S_U(\mathbf{u}) = \sum_{k\in[\mathcal{D}]} U(u_k)$, where $U_\alpha(x) = \frac{x^{1-\alpha}}{1-\alpha}$ if $\alpha \neq 1$ and $U_\alpha(x) = \log(x)$. As such, the team-oriented advantage is $\hat{\mathbf{A}}^{\text{SWF}} = \nabla_{\mathbf{u}}\phi(\hat{\mathbf{J}}(\theta))^{\top} \cdot \hat{\mathbf{A}}(\mathbf{o}, \mathbf{a})$, as opposed to the traditional self-oriented one $\hat{\mathbf{A}}_i^{\text{IND}} = \hat{A}_i(o_i, a_i)$. After derivation, note that for the GGF and α SWF $\nabla_{\mathbf{u}} G_{\mathbf{w}}(\mathbf{J}(\theta))$ is $\mathbf{w}_\sigma$ and $\mathbf{J}(\theta)^{-\alpha}$, for $\alpha \in [0, 1[$, respectively.

3.2 Heterogeneous Testing

We coin heterogeneous testing as the method utilised to explore fair vs.efficient relative frequencies within the policies chosen by the agents. Remember that, from SOTO, choosing either pi^{IND} and pi^{SWF} depended on the value of $\beta(e_r)$. Inspired by this, we employ the algorithm depicted in Algorithm 1 to test heterogeneous behaviours from two different previously learned policies. In this case, β a fixed parameter is provided as an argument that determines the probability of the agent being attributed the self-oriented policy.

It is important that under this testing method, no agent acts only selfishly or fairly but more often as to one of these according to β by the law of large numbers. The intent is to expose the behaviour of interactions of each policy kind for each agent, as all multi-agent architectures are independent per agent. Moreover, the policies are never updated under this method. As such, this is only a testing method, putting in evidence the policy resultant from training without changes.

Algorithm 1 Heterogeneous Testing

1: Initialize $\pi_i^1, \pi_i^2, v_i^1, v_i^2$, respectively the pre-trained team-oriented/self-oriented policies, team-oriented/self-oriented critics.
2: **for** each episode e **do**
3: **while** episode e is not completed **do**
4: $(\pi_i, v_i) \leftarrow \begin{cases} (\pi_i^1, v_i^1) & \text{with probability } \beta(e_r), e_r = \frac{e}{E} \\ (\pi_i^2, v_i^2) & \text{otherwise} \end{cases}$
5: Collect M a minibatch of transitions with π_i
6: **end while**
7: **end for**

3.3 Intertwined Self-Oriented Team-Oriented networks

In the SOTO architecture, the inclusion of a self-oriented policy is intended to provide a recommendation on how to act efficiently to the team-oriented one. It is unknown, however, if the recommendations of a team-oriented policy could improve the performance of the first as well.

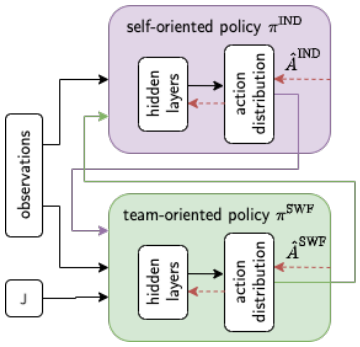

Figure 1: I-SOTO Architecture

We propose an extension to this architecture in which the self-oriented policy also receives insights from the team-oriented policy, generating an intertwined sharing of recommendations. We coin this model Intertwined Self-Oriented Team-Oriented networks - I-SOTO. The team-oriented policy then receives as input the action distribution resultant from forwarding π^{IND}, as shown in Figure 1.

A problem that arises is the circular dependency between policy recommendations or forwarded outputs. For instance, if we want to forward policy π^{SWF}, first we have to forward policy π^{IND}. However, to forward this policy, we also need the forwarded output of the first, which is dependent on the latter. In order to address this issue, whenever some policy π is being used, it forwards the other π' substituting the expected inputs from π with a null vector $\mathbf{0}^{|\mathcal{A}|}$. This null action recommendation could be seen as "no action", since the output of the policies is the probability distribution of each available action.

Note that the distribution of wealth $\mathbf{J}$ is yet not passed to the self-oriented policy to put in evidence the effect of intertwined selfish and fair policies.

3.4 Training Strategy functions

In the context of this work, training strategy refers to the function of $\beta(e_r)$, which determines the probability of each agent choosing to act under the self-oriented policy π^{IND} on a given episode for and on before mini-batch of M transitions. Higher β makes the agents train more the self-oriented policy and then vice-versa for the team-oriented policy. Notice that, specially in I-SOTO, because learning one of these policies has a potential impact on the other, using different training strategies may improve the performance of the model.

We test a variety of functions $\beta(e_r)$, where $e_r = \frac{e}{E}$ is the episode rate, i.e. the number of the current episode e divided by the total number of episodes E. In the original training setting of SOTO, $\beta(e_r)$ is a linearly decreasing function until half of the episodes $\frac{1}{e}e_r$ and constant from such point onwards on 0. We use 4 different beta families: constant, linear, baseline and v-shaped. A summary of their characteristics is present in Table 1. Sample ratio refers to the ratio between

Family	Variant	$\pi^{\text{IND}}/\pi^{\text{SWF}}$ sample ratio	Equation
Constant	0.25	25/75	0.25
	0.5	50/50	0.5
Linear	lin	50/50	e_r
	rlin	50/50	$1 - e_r$
Baseline	b	25/75	$\max(1 - 2e_r, 0)$
	rb	75/25	$1 - \max(1 - 2e_r, 0)$
V-shaped	v	50/50	$\max(1 - 2e_r, 2e_r - 1)$
	rv	50/50	$1 - \max(1 - 2e_r, 2e_r - 1)$

Table 1: Strategy Functions by family: Constant, Linear (and its reverse version rl) , Baseline (and its reverse version rb) and V-shaped (and its reverse version rv)

the areas below and under the curve, respectively. When β is higher, π^{IND} is trained more often and thus has access to a higher number of samples. The number of samples tends to the proportions of the areas below and under the curve, by the law of large numbers.

The constant family is the most simple of all. Under this family, β is not dependant on e_r, and the agents train selfishly/fairly according to the same probability throughout episodes. In particular, we want to study two values of β: 0.25 and 0.5. Studying $\beta = 0.5$ is important as it gives the same opportunity for each policy π^{IND} and π^{SWF} to converge - equal sample ratio. On the other hand, studying $\beta = 0.25$ provides the same sample ratio as the baseline. Alternatively, the linear family functions prolong the switch between choosing one or the other policy to double the period compared to the baseline setting. We aim to test whether the stabilising period of baseline after $\frac{T}{2}$ is necessary or if most of its performance gains come from slowly switching from self-oriented to team-oriented. As for the baseline family, it comprises the baseline setting of SOTO and its reverse, *rb* (see Table 1). Notice that the reverse variant ends up being very similar to the independent baseline except that 25% of the samples are, in the first half of episodes, directed to the team-oriented policy during training. Finally, derived from the baseline alternative, v-shaped functions are intended to provide the same function as baseline until $\frac{E}{2}$ episodes and then return to the initial value linearly in a V-shaped manner. The interest relies on testing whether should the system progressively return to the initially dominant policy during training, there is an improvement of its behaviour afterwards.

As such, we extend the literature by exploring fair and efficient goals in a holistic manner by evaluating execution and training methods which combine them, hoping to find competitive solutions in both goals.

4 Experimental Results & Discussion

4.1 Evaluation

All of these methods are evaluated through simulation. We choose two environments in which is the resource opportunity is unequal - Matthew Effect and Traffic Light Control. As baselines, we use the original SOTO model and the same Independent baseline utilised in its paper, which has the same architecture of SOTO self-oriented policy. In each environment, a domain-specific measure per time step and agent m_t^i is considered. Given $\mathbf{u} = \{u_i, i \in \mathcal{D}\}, u_i = \sum_t^T m_t^i$, metrics recorded were the total as $\sum_t^T \sum_i^{|\mathcal{D}|} m_t^i$, CV [2] as std($\mathbf{u}$)/mean($\mathbf{u}$), the min $\min(\mathbf{u})$ and the max $\max(\mathbf{u})$. Lower values of CV indicate fairer solutions. Higher/Lower values of total provide information on the efficiency of the model, depending whether m_t is to be maximised/minimised on the environment. All policies use PPO optimisation. The importance sampling has a 0.03 exploration bonus and 0.1 clipping ratio. The learning rate is 10^{-3} for the critic and 2.5^{-3} for the actor. Generalised Advantage Estimation was utilised with $\lambda = 0.97$. The neural networks have two hidden layers with 256 ReLU units each. We used 50 time step batches of transitions. Two Social Welfare Functions (SWF) were utilised: an instance of the Generalised Gini Function, with $\boldsymbol{w}_i = \frac{1}{2^i}$ and an instance of α-fairness with $\alpha = 0.9$

Each model is trained three times with different seeds for stochastic processes. The results presented are the average of every seed instance tested in 50 episodes. The values of β used for heterogeneous behaviour were $\{0.02i, \forall i \in \{0, 1, ...50\}\}$, for models where this is applicable.

4.2 Matthew Effect Problem

In the Matthew Effect environment, a set of 10 agents is placed in a map. Whenever an agent consumes a ghost, it gets bigger and faster, and a new ghost is spawned. As such, those who consume are more likely to consume again. In other words, the rich get richer and the poor get poorer. This is called the Matthew effect. The goal is to maximise the *income*, in this case the number of consumed ghosts n. The recorded measure for this environment coincides with n and the reward signal r, such that $m_t = r_t = n_t$.

Behavior generation through Heterogeneous Testing
We present the behaviour outcomes of heterogeneous testing between different pairs of policies, trained with the SOTO model, in the following paragraphs.

π^{IND} versus π^{SWF}: SOTO trains two policies with different aims: a self- and a team-oriented goal. The results of the heterogeneous behaviour produced by these policies in the Matthew Effect environment is depicted in Figure 2. It seems that the two SWFs utilised can generate ranges of behaviour, according to β, in two different directions. Regarding fairness, as expected, the lower values of β seem to be associated with lower CV values. This means the higher the probability of each agent to act under π^{SWF}, the fairer the system is, globally. On the other hand, with regards to efficiency, a more complex scenario occurs. Unexpectedly, for SOTO(α), there seems to be an inverse relationship between β and the value of total income. Indeed, the most efficient policy is also the fairest. As for SOTO(G_w), such a relationship is no longer linear. The most efficient policy is an intermediate behaviour between the two extremes π^{IND} and π^{SWF}. It is possible to observe that these two SWFs have quite distinct behaviours under this environment. However, a similarity between them is the proximity in performance between self-oriented extremes in each SWF. We believe this may be

[2] Coefficient of Variation

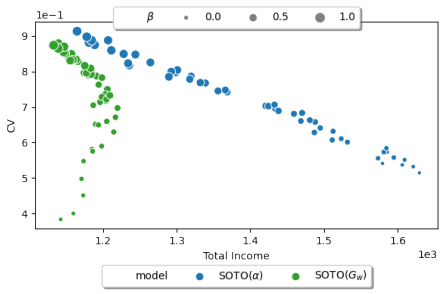

Figure 2: Heterogeneous behaviour between SOTO's π^{IND} and π^{SWF} in Matthew Effect

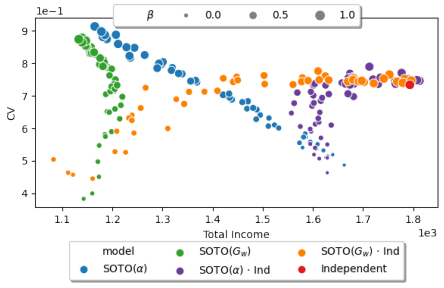

Figure 3: Heterogeneous behaviour between SOTO and Independent in Matthew Effect

model	β	π	Total	CV	Min	Max
I-SOTO(α)	0.25	IND	1440	0.75	11.80	356
		SWF	1529	**0.48**	35.22	271
	0.5	IND	1626	0.66	24.18	385
		SWF	1378	0.53	15.67	240
	lin	IND	**1771**	0.64	31.42	417
		SWF	888	0.69	4.77	167
	rlin	IND	1617	0.64	23.68	372
		SWF	**1680**	**0.48**	38.34	306
	b	IND	1138	0.94	3.69	338
		SWF	**1756**	**0.44**	58.02	316
	rb	IND	**1859**	0.65	23.31	423
		SWF	99	1.28	0.16	37
	v	IND	1641	0.64	22.59	372
		SWF	1473	0.56	29.96	293
	rv	IND	1573	0.68	17.12	374
		SWF	1550	0.52	21.75	282
SOTO(α)	b	IND	1178	0.90	5.88	342
		SWF	**1663**	**0.49**	43.29	297
I-SOTO(G_w)	0.25	IND	**1178**	0.86	6.91	327
		SWF	733	0.21	43.32	87
	0.5	IND	**1552**	0.64	25.88	358
		SWF	130	0.89	0.90	34
	lin	IND	**1724**	0.67	22.82	407
		SWF	37	1.09	0.07	10
	rlin	IND	**1210**	0.84	7.47	332
		SWF	649	0.33	24.33	87
	b	IND	**1156**	0.88	5.63	326
		SWF	1035	**0.01**	101.06	106
	rb	IND	**1799**	0.71	15.05	433
		SWF	11	1.58	0.01	5
	v	IND	**1479**	0.77	13.61	372
		SWF	355	0.88	2.68	100
	rv	IND	**1241**	0.81	8.97	325
		SWF	581	0.37	18.50	81
SOTO(G_w)	b	IND	**1139**	0.86	9.00	324
		SWF	**1052**	**0.03**	99.68	109
Independent	N.A.	N.A.	**1793**	**0.73**	8.11	421

Table 2: I-SOTO performance under the self- (IND) and team-oriented (SWF) policies with regards to Income in Matthew Effect

because such policy only trains for 25% of the training samples. This would also justify why the efficiency of such policy is comparatively mediocre, despite its training goal being directly oriented towards efficiency.

SOTO versus Independent: We test mixing SOTO with the Independent baseline. The results of this attempt are depicted in Figure 3. As we can see, no Pareto solutions are found.Both ranges of behaviours produced non-linear shapes.

I-SOTO and Training Strategies

We present the results for I-SOTO under different training strategies in Table 2. As can be seen, some solutions are able to outperform SOTO. For the α SWF, the rlin and b functions are both fairer and more efficient than this baseline. These functions are the only ones where β decreases (weakly) thoughout episodes, so perhaps this could be an intuition on the ideal function to be used in this SWF and environment. Morever, for G_w, it seems that no solution is better in both goals. We found that the baseline training strategy is able to improve in fairness, which is remarkable since it was already very close to 0 in SOTO.

Comparing to the Independent baseline, there are solutions in both SWF that outperform it both in fairness and efficiency. For α, the b function is the one which does this with greater difference. Another interesting result for this SWF is the rb function, achieving an even higher value of efficiency. For G_w, the rb alternative is able to have slightly higher efficiency and fairness (lower CV).

4.3 Traffic Light Control Problem

This environment consists of a 3x3 grid of lanes where the traffic light state of each intersection is controlled by an agent. The goal is to minimise the total weighting time in all intersections w. As such, the reward provided to agents at each time step is $r_t = w_{t-1} - w_t$. However, as measure, we simply record the weighting time such that $m_t = w_t$. Notice that, contrary to the previous environment, the reward attributed to each agent dependents on external factors: the vehicles waiting in such intersection. Again, this will naturally provide agents unequal opportunities to receiving rewards, as waiting times of intersections dependent highly on the trajectories of the vehicles, waiting times of other intersections, etc.

Behavior generation through Heterogeneous Testing

The results of the heterogeneous behaviour produced by the self- and team-oriented policies of SOTO are depicted in Figure 4. It is possible to observe that the range of SOTO behaviours generated is approximately linear in the efficiency-fairness space. The team-oriented end is both the more efficient and fair than the self-oriented one. When compared to the previous environment, this phenomenon also occurred for the α-fairness metric. In the G_w, the performance range was not linear in the efficiency dimension, so we can conclude that the behaviour of the same SWF can be different under different environments.

Regarding the range of behaviours generated by π^{IND} and π^{SWF}, it seems to be sparser than in the Matthew Effect. Nonetheless, the Independent baseline over-performs SOTO in any option of the range. For this reason, we did not proceed

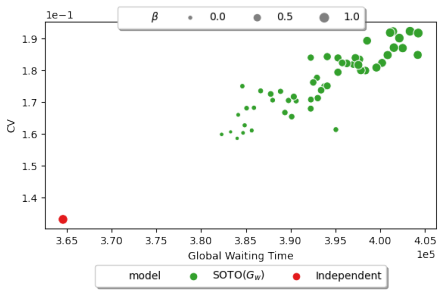

Figure 4: Heterogeneous behaviour between SOTO's π^{IND} and π^{SWF} in Traffic Light Control

with experiments testing heterogeneous behaviour between these two options, as one already Pareto-fronts the other. An intuition for this phenomenon would be the dependence between agents rewards causing a correlation between efficiency and fairness. Under that assumption, a model which focuses 100% entirely on one of them, compare to a model which divides samples between two policies, is more likely to succeed in both goals should they be correlated.

I-SOTO and Training Strategies

As in the previous environment, we present the results of I-SOTO in this environment in Table 3. As can be seen, many I-SOTO solutions are both more efficient and fair than the SOTO baseline. In particular, at least one extreme in each training strategy utilised outperforms SOTO. The most prominent of them would be the rb_{IND} training strategy function. This is the option with most samples dedicated to the selfish goal, being in agreement with the intuition that fairness and efficiency - in this environment - are somewhat correlated. This is also corroborated with the fact that the most competitive options are on the self-oriented side of I-SOTO, and not the team-oriented one as occurred in Matthew. As for the Independent baseline, a similar phenomenon occurs. Only the constant functions are not able to compete with these baseline, interestingly. While it is hard to provide an explanation for why this happened, one hypothesis could be that these are the only ones where samples are never entirely dedicated to one of the policies - or, more importantly in this case - the self-oriented one.

To conclude, there is a broader analysis than can be made relative to the whole experimental set. With regards to the self- and team-oriented policies in any of the models, there seems to be a pattern where either (1) π^{SWF} is the most fair and π^{SWF} is the most efficient policy or that (2) π^{SWF} is both the most efficient and fair. The latter case occurs for SOTO(α) in Matthew Effect and SOTO(G_w) in Traffic Light Signal. While we are not able to provide an exact explanation of why this happens, there are two potential intuitions for this reason.

On the one hand, it may have to do with the nature of the environment. As previously seen, in Traffic Light Control, the success of an agent (intersection) is highly dependent on the success of other agents. This leads to the intuition that finding a fair solution, in this environment, is also finding a fair one. A result that is in agreement with this is the fact that the

model	β	π	Total	CV	Min	Max
I-SOTO(G_w)	0.25	IND	**3.70e+05**	**0.08**	8.5e+02	3.6e+04
		SWF	3.70e+05	0.09	1.2e+03	2.9e+04
	0.5	IND	**3.33e+05**	**0.14**	8.4e+02	3.2e+04
		SWF	4.54e+05	0.12	2.2e+03	3.7e+04
	b	IND	3.60e+05	0.09	9.8e+02	3.1e+04
		SWF	**3.50e+05**	**0.08**	9.8e+02	2.9e+04
	lin	IND	**3.36e+05**	**0.09**	8.8e+02	3.3e+04
		SWF	4.72e+05	0.10	1.5e+03	4.2e+04
	rb	IND	**3.34e+05**	**0.06**	7.7e+02	4.0e+04
		SWF	7.39e+05	0.16	2.8e+03	6.4e+04
	rlin	IND	**3.57e+05**	**0.13**	9.7e+02	3.6e+04
		SWF	4.20e+05	0.11	1.4e+03	3.6e+04
	rv	IND	**3.44e+05**	**0.08**	9.7e+02	3.3e+04
		SWF	4.38e+05	0.10	1.5e+03	3.5e+04
	v	IND	**3.37e+05**	**0.08**	7.9e+02	3.2e+04
		SWF	4.24e+05	0.09	1.2e+03	3.4e+04
SOTO(G_w)	b	IND	4.03e+05	0.19	9.4e+02	4.1e+04
		SWF	**3.83e+05**	**0.16**	1.2e+03	3.2e+04
Independent	N.A.	N.A.	**3.65e+05**	**0.13**	9.8e+02	4.2e+04

Table 3: I-SOTO performance under the self- (IND) and team-oriented (SWF) policies with regards to waiting time in Traffic Light Control

best performing model in this environment is I-SOTO(G_w) with the constant 0.5 strategy function, in which self- and team-oriented insights are shared between policies and in a balanced (50/50) way between goals.

On the other hand, this may also have to do with the nature of the social welfare function utilised. As seen in section 3.1, the self- and team-oriented advantages utilised in the training process are a product of the derivative of the SWF with respect to the agents utilities, $\nabla_{\mathbf{u}}\phi(\hat{\mathbf{J}}(\theta))^{\top}$, with the original advantage. In the α-fairness scenario, this derivative is $\mathbf{u}^{0.9}$, while on the G_w setting it is $\mathbf{w} = \{2^{-i}, \forall_{i \in \mathcal{D}}\}$. This means that the team-oriented advantage for the first case is a sum of an exponential function to the agents utilities as opposed to a weighted sum based on their ranking. The fact that this function interprets social welfare as an independent concept from the ranking of individual utilities within the system perhaps deposits more confidence in individual success - efficiency - as a means towards fairness. Considering the utility order overall produces much fairer results as it ensures no agent is being left behind. This, however, comes at the cost of a great deal in efficiency.

5 Conclusion

We approach fairness and efficiency in a holistic manner: either by mixing pre-trained efficient and fair policies or by changing the learning method of SOTO such that fair-efficient recommendations are intertwined - I-SOTO. In the latter, we were able to find some solutions which outperformed not only the fair baseline but also the efficient baseline utilised. Despite being initial attempts in the problem, these are important results towards better understanding the fairness-efficiency relationship. With regards to our hypothesis we find that the heterogeneous behaviours found between efficient and fair policies are not always linear, unexpectedly. For I-SOTO, we confirmed that some results were indeed better performing than SOTO but for the G_w SWF no solution was found to be better in both of the goals at study.

This new approach to address fairness and efficiency could be particularly important in systems with endogenous resources: i.e. computation of the reward distribution has to

be paid for from the rewards themselves so that learning a fair and efficient combination with respect to available resources is particularly important. For that we intend to expand the testing space along different dimensions: environments, SWFs and training strategies.

References

[Brams *et al.*, 1996] Steven J Brams, Steven John Brams, and Alan D Taylor. *Fair Division: From cake-cutting to dispute resolution*. Cambridge University Press, 1996.

[Busa-Fekete *et al.*, 2017] Róbert Busa-Fekete, Balázs Szörényi, Paul Weng, and Shie Mannor. Multi-objective bandits: Optimizing the generalized gini index. In *International Conference on Machine Learning*, pages 625–634. PMLR, 2017.

[Chen *et al.*, 2021] Dezhi Chen, Qi Qi, Zirui Zhuang, Jingyu Wang, Jianxin Liao, and Zhu Han. Mean Field Deep Reinforcement Learning for Fair and Efficient UAV Control. *IEEE Internet of Things Journal*, 8(2):813–828, 1 2021.

[Claure *et al.*, 2019] Houston Claure, Yifang Chen, Jignesh Modi, Malte Jung, and Stefanos Nikolaidis. Multi-Armed Bandits with Fairness Constraints for Distributing Resources to Human Teammates. *ACM/IEEE International Conference on Human-Robot Interaction*, pages 299–308, 6 2019.

[Elmalaki, 2021] Salma Elmalaki. Fair-iot: Fairness-aware human-in-the-loop reinforcement learning for harnessing human variability in personalized iot. In *Proceedings of the International Conference on Internet-of-Things Design and Implementation*, pages 119–132, 2021.

[Hughes *et al.*, 2018] Edward Hughes, Joel Z Leibo, Matthew Phillips, Karl Tuyls, Edgar Dueñez-Guzman, Antonio García Castañeda, Iain Dunning, Tina Zhu, Kevin McKee, Raphael Koster, et al. Inequity aversion improves cooperation in intertemporal social dilemmas. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 3330–3340, 2018.

[Jiang and Lu, 2019] Jiechuan Jiang and Zongqing Lu. Learning fairness in multi-agent systems. *Advances in Neural Information Processing Systems*, 32:13854–13865, 2019.

[Lamertz, 2002] Kai Lamertz. The social construction of fairness: Social influence and sense making in organizations. *Journal of Organizational Behavior*, 23(1):19–37, 2002.

[Luss, 2012] Hanan Luss. *Equitable Resource Allocation: Models, Algorithms and Applications*, volume 101. John Wiley & Sons, 2012.

[Mehrabi *et al.*, 2021] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

[Mo and Walrand, 2000] Jeonghoon Mo and Jean Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on networking*, 8(5):556–567, 2000.

[Moulin, 2003] Hervé Moulin. *Fair division and collective welfare*. MIT press, 2003.

[Oliehoek and Amato, 2016] Frans A. Oliehoek and Christopher Amato. *A Concise Introduction to Decentralized POMDPs*. SpringerBriefs in Intelligent Systems. Springer International Publishing, Cham, 2016.

[Pióro *et al.*, 2002] Michał Pióro, Gábor Malicskó, and Gábor Fodor. Optimal link capacity dimensioning in proportionally fair networks. In *International Conference on Research in Networking*, pages 277–288. Springer, 2002.

[Siddique *et al.*, 2020] Umer Siddique, Paul Weng, and Matthieu Zimmer. Learning fair policies in multi-objective (deep) reinforcement learning with average and discounted rewards. In *International Conference on Machine Learning*, pages 8905–8915. PMLR, 2020.

[Wang *et al.*, 2020] Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. Shapley q-value: a local reward approach to solve global reward games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7285–7292, 2020.

[Weng, 2019] Paul Weng. Fairness in Reinforcement Learning. *34th International Conference on Machine Learning, ICML 2017*, 4:2542–2557, 7 2019.

[Weymark, 1981] John A Weymark. Generalized gini inequality indices. *Mathematical Social Sciences*, 1(4):409–430, 1981.

[Zhang and Shah, 2014] Chongjie Zhang and Julie A Shah. Fairness in multi-agent sequential decision-making. In *Advances in Neural Information Processing Systems*, pages 2636–2644, 2014.

[Zimmer *et al.*, 2021] Matthieu Zimmer, Claire Glanois, Umer Siddique, and Paul Weng. Learning fair policies in decentralized cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*, 2021.