Fairness in Cooperative Multiagent Multiobjective Reinforcement Learning using the Expected Scalarized Return

Farès Chouaki
LIP6, Sorbonne Université, CNRS
F-75005 Paris, France
fares.chouaki@lip6.fr

Aurélie Beynier
LIP6, Sorbonne Université, CNRS
F-75005 Paris, France
aurelie.beynier@lip6.fr

Nicolas Maudet
LIP6, Sorbonne Université, CNRS
F-75005 Paris, France
nicolas.maudet@lip6.fr

Paolo Viappiani
LAMSADE, CNRS, Université Paris Dauphine - PSL
Paris-75016, France
paolo.viappiani@lamsade.dauphine.fr

ABSTRACT

Fairness as equity and compromise across multiple viewpoints is a necessary consideration in any kind of decision that is evaluated from several possibly conflicting perspectives. It is also a property that artificial decision-making agents should uphold to be deployable to real-world problems. In the sequential decision-making community, focus has been put on designing algorithms that ensure fairness either only among agents or just among objectives. However, most real-world problems can't be reduced to the optimization of a single objective and are concerned with the control of a fleet of cooperative agents. The multi-objective and multi-agent nature of such problems makes existing algorithms inadequate. Indeed, single-objective multi-agent approaches are not adapted for multi-objective optimization and single-agent multi-objective approaches cannot handle multiple agents. Furthermore, research integrating fairness into Multi-objective Reinforcement Learning (MORL) is focused on the scalarized expected return (SER) optimization criterion while mostly ignoring the expected scalarized reward criterion (ESR). We argue that fairness in MORL should also be investigated under ESR since sometimes it is more suitable when solving problems where fairness matters. In this paper, we consider the problem of learning objective-wise fair policies in cooperative multi-agent multi-objective sequential decision-making problems. We propose the first mono-policy algorithm able to learn efficient decentralized policies while ensuring fairness across objectives under ESR. Our algorithm is evaluated on a novel environment that models a cooperative multi-objective multi-agent task and achieves better performances than the considered baselines.

KEYWORDS

Multi-agent learning, Reinforcement learning, Fairness,
Multi-objective learning

1 INTRODUCTION

Solving real-world sequential decision-making problems usually requires compromising between several conflicting objectives while coordinating multiple agents. For example, in a ride-sharing application, vehicles are rewarded based on the number of passengers of each type that achieves their destination. It has been shown that a naive scalar reward approach for such a problem can discriminate against passengers who require more time to get in and out of a vehicle, such as wheelchair users [2]. Also, such policies need to perform relatively well on each execution and not just on average over several executions. For instance, a group of electric plants supplying different parts of a city with electricity needs to coordinate their daily production and distribution to ensure that each neighborhood is well accommodated. This distinction in the value of the policy is reflected in the Multi-Objective Reinforcement Learning (MORL) literature by the existence of two optimization criteria, namely the Scalarized Expected Return (SER), which searches for the policy with the best average value, and the Expected Scalarized Return (ESR), that searches for the policy that yields the best value at each execution. Such algorithmic solutions must satisfy several ethical values, such as fairness,to be deployed in the real world and accepted by the users. In the previous example, a fair and efficient policy should ensure that the group of power plants satisfies the needs of each neighborhood to the same proportion every day while maximizing that proportion.

In this paper, we will focus on computing fair and efficient policies among objectives.Existing RL-based solutions usually consider single-agent or single-objective simplifications of the problem, however, such solutions cannot solve the multi-objective multi-agent version of the problem. On the one hand, single-objective multi-agent algorithms are often concerned with fairness among acting agents and are not suited for multi-objective optimization. On the other hand, single-agent multi-objective solutions rarely consider fairness under ESR, thus only guaranteeing fairness across objectives on the average of several policy executions. Solutions tackling the problem in its full multi-objective multi-agent complexity do not yet integrate fairness.

This paper provides an argument as to why ESR can be more suitable for solving multi-objective sequential decision-making problems while ensuring fairness. A novel reinforcement learning algorithm to solve the multi-agent version of such problems is presented. The evaluation results of the algorithm show that it ensures a tradeoff between efficiency and fairness for each policy execution and outperforms the baselines considered.

Proc. of the Adaptive and Learning Agents Workshop (ALA 2025), Avalos, Aydeniz, Müller, Mohammedalamen (eds.), May 19 – 20, 2025, Detroit, Michigan, USA, ala-workshop.github.io. 2025.

2 BACKGROUNG AND NOTATIONS

This section introduces the background our approach is built upon and notations used in the remainder of the paper.

2.1 Multi-Objective Decentralized Partially Observable Markov Decision Processes

The Multi-Objective Decentralized Partially Observable Markov Decision Processes (MO-DEC-POMDPs) framework is used to solve multi-objective cooperative multi-agent reinforcement learning problems. A MO-DEC-POMDP is a special case of the more general Multi-Objective Partially Observable Stochastic Game (MO-POSG) framework [22] and a multi-objective extension of the DEC-POMDP model [16]. A MO-DEC-POMDP is a tuple $\langle d, S, \{A^i\}, T, \mu, R, \{O^i\}, \Omega, \gamma \rangle$ where:

- d: the number of objectives to optimize.
- S: the state space.
- A^i: the action space of agent i. The joint action space is $A = \times_i A^i$. In this work, we assume that all agents have identical action spaces.
- $T : S \times A \times S \rightarrow [0, 1]$: the transition function mapping $(s, \mathbf{a}, s')$ to the probability of transitioning from $s \in S$ to $s' \in S$ under joint action $\mathbf{a} \in A$. The function satisfies the constraint: $\forall s \in S, \mathbf{a} \in A, \sum_{s' \in S} T(s, \mathbf{a}, s') = 1$.
- $\mu : S \rightarrow [0, 1]$: the initial state distribution over S, satisfying $\sum_{s \in S} \mu(s) = 1$.
- $R : S \times A \rightarrow \mathbb{R}^d$: a vector-valued reward function that maps $(s, \mathbf{a})$ to a reward vector $\mathbf{r} \in \mathbb{R}^d$. The vectorial structure reflects the presence of multiple objectives, not multiple agents. Unless stated otherwise, we assume a shared reward structure, meaning that all agents are cooperative and receive the same reward vector at each timestep.
- O^i: the observation space of agent i, i.e., the set of observations that agent i may receive when interacting with the environment. We assume all agents have the same observation space.
- $\Omega^i : A \times S \times O^i \rightarrow [0, 1]$: the observation function for agent i, mapping $(\mathbf{a}, s', o^i)$ to the probability of agent i observing $o^i \in O^i$ upon transitioning to state $s' \in S$ after joint action $\mathbf{a} \in A$. It satisfies the constraint: $\forall \mathbf{a} \in A, s' \in S, \sum_{o^i \in O^i} \Omega^i(\mathbf{a}, s', o^i) = 1$.
- $\gamma \in [0, 1)^d$: a vector of discount factors, one for each objective.

If the reward function is agent-specific then the model becomes a MO-POSG[22]. When only one agent is present in the model it becomes an MO-POMDP [24] and when that agent observes its real state the model is referred to as MO-MDP [9].

2.2 Fair aggregation functions

In multi-objective reinforcement learning, the reward function $R(s, a)$ consists of a d−dimensional function that returns the value of performing an action in a certain state over each of the d objectives considered in a problem. Additional information is needed to establish a total ordering over an agent's possible policies. Most works in MORL achieve this total ordering through a scalarisation function i.e.a multivariate function that transforms the vector reward into a scalar. Such a scalarisation function can be learnt from the user of the MORL algorithm, or directly from the problem specifications. Several functions that provide a tradeoff between fairness and efficiency can be found in the fair multi-objective optimization literature. Based on the findings of [14] we opt to use the Nash Social Welfare (NSW) scalarisation function. When all the objectives are stictly positive, this function is defined by Equation 1:

$$NSW(\mathbf{r}) = \prod_{i=1}^{d} \mathbf{r}_i. \tag{1}$$

2.3 Optimization criteria in multi-objective reinforcement learning

A single-objective reinforcement learning algorithm learns a policy that maximizes the expected discounted or average reward. However, in multi-objective reinforcement learning, when a scalarisation function is given, the designer of the system has to choose between two criteria [21]:

- Scalarised Expected Return (SER): given a scalarisation function $u : \mathbb{R}^d \rightarrow \mathbb{R}$ the value V^π of a policy π under this criterion is given by Equation 2:
$$V_u^\pi = u\left(\mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^t r_i \mid \pi, s_0\right]\right). \tag{2}$$
- Expected scalarized Return (ESR): given a scalarisation function $u : \mathbb{R}^d \rightarrow \mathbb{R}$ the value V^π of a policy π under this criterion is given by Equation 3:
$$V_u^\pi = \mathbb{E}\left[u\left(\sum_{i=0}^{\infty} \gamma^t r_i\right) \mid \pi, s_0\right]. \tag{3}$$

Interestingly, when the scalarisation function is linear both criteria are equivalent and lead to the same optimal policies. However, when the scalarisation function is non-linear, the learned optimal policies are sensibly different [9, 21, 22]. Section 4.1 explains how the choice of criteria can affect the learned policies and illustrates this difference through toy examples.

3 RELATED WORK

Our approach aims to solve Multi-Objective and Multi-Agent Reinforcement Learning (MOMARL) problems assuming a utility-based approach. This section presents a non-exhaustive review of the existing solutions on single-agent multi-objective RL, and single-objective multi-agent RL while focusing on algorithms that explicitly aim to learn fair and efficient policies.

3.1 Multi-objective reinforcement learning

Roijers, et al. [20] proposed the Expected Utility Policy Gradient (EUPG) algorithm that finds the optimal policy given a known non-linear utility function under the ESR criterion. This algorithm extends the policy gradient algorithm by integrating the accrued returns in the computation of state action values and conditioning the policy by the total past return and the state instead of just considering the state. Reymond, et al. [19] extend the previous approach and propose an actor-critic variation of EUPG by introducing a critic that learns multivariate distribution over the returns allowing the bootstrapping of returns and learning during episodes (unlike EUPG that only updates the agent's policy at the end of an episode). Based on the Monte Carlo Tree Search (MCTS) algorithm, [8] introduces Non-Linear Utility MCTS (NLU-MCTS) and Distributional MCTS (D-MCTS). These extensions compute optimal policies under the ESR criterion with known non-linear utility functions.

Cimpean et al. [3] propose a framework to estimate group and individual fairness as extra objectives in the MDP to ensure that the policy learned by the agents is not biased towards a certain group of people or toward some individuals. Mandal et al. [14] propose a set of axioms that a fair scalarisation function should satisfy for fair SER optimization. They show that the NSW scalarisation function (presented in Section 2.2) is the only one that can satisfy the proposed axioms. Nevertheless, their approach is only valid for SER optimization. Indeed, we show in Section 4.1 that NSW doesn't necessarily respect the Pareto optimality axiom under ESR. Siddique et al. [23] proposed the GGI-DQN, GGI-A2C, and GGI-PPO algorithms that are used to learn policies in multi-objective problems while ensuring fairness among the objectives. Fairness is achieved through the Generalized Gini Index (GGI) aggregation function. However, this approach considers the SER criterion instead of the ESR criterion and can only be used in the case of single-agent problems. On the other hand [4] propose an adaptation of Q-learning they call *welfare Q-learning* that learns a single agent policy optimizing the NSW function under the ESR criterion.

More exhaustive reviews on multi-objective reinforcement learning research can be found in [21] and [9].

3.2 Multi-agent reinforcement learning

We are interested in settings where the agents learn how to act in a partially observable environment with a common vectorial reward structure. Since there is no centralized controller to coordinate the agents, only decentralized policies are admissible solutions. Peshkin et al. [17] proposed the first fully independent extension of the REINFORCE algorithm to the cooperative single-objective multi-agent case. Aiming to leverage the use of value functions and bootstrapping [6] proposed the independent-A2C algorithm as one baseline to evaluate their counter-factual actor-critic algorithm. One of the drawbacks of fully independent algorithms is the assumption that centralized information is never available to the agents. However current trends in MARL show that during training policy-based algorithms [6, 13] and value-based algorithms [18, 25–27] can leverage centralized information to learn better policies.

Jiang et al.[11] consider the problem of fair competitive MARL. Their approach, called Fair Efficient Networks (FEN), consists of a fully decentralized hierarchical learning approach that decomposes fairness among agents. This decomposition is made possible thanks to their fair-efficient reward function. The same problem is tackled by [12] where they propose a first algorithm that achives sub-linear regret bound for the α-fairness function.

3.3 Multi-agent multi-objective reinforcement learning

In a multi-agent and multi-objective context, Mannion et al. [15] presents a theoretical analysis of the difference reward when applied in a multi-objective multi-agent setting and proves that using a difference reward instead of a global reward doesn't alter the relative ordering of rewards. This property allows the usage of the difference reward as a reward-shaping mechanism. [10] introduces the first multipolicy algorithm that can solve cooperative multi-objective multi-agent reinforcement learning problems. They propose a Centralized Training Decentralized Execution approach (CTDE) where each agent, conditioned by the preferences over the objectives, learns to estimate its vectorial state-action value function while a mixing network is used to estimate the global state-action value function. To provide a unified benchmark for both cooperative and competitive MOMARL algorithms, [5] presents MOMALand, the first suit of environments which includes 10 environments that can be used to train and evaluate agents across several MOMARL tasks. Radulescu et al. [22] propose a utility-based taxonomy of multi-gent multi-objective problems. A review of the algorithms and their applications is also presented along with several open research questions.

3.4 Open questions

This section presented a non-exhaustive review of approaches integrating fairness in multi-objective and multi-agent reinforcement learning. Although fairness is an important part of both single-objective MARL and single-agent MORL, existing solutions still suffer from the following limitations:

1. Single objective MARL solutions like [11] are concerned with fairness between agents which makes them unsuitable for the common reward setting.
2. MOMARL being a new research field, solutions from that sphere like [10] are general and have not yet tackled fairness issues.
3. Existing MOMARL solutions have not proposed algorithms for cooperative tasks with known utility functions, and the existing solutions are only applicable to the SER criterion.
4. MORL solutions that try to learn fair policies regarding the objectives are only applicable for SER and overlook ESR.
5. Existing MORL algorithm for ESR optimization cannot currently solve multi-agent problems.

In this paper we address limitations 3, 4 and 5. We argue that fairness should be considered under both SER and ESR criteria. A new fully decentralized algorithm for the known utility and common reward setting is proposed. This algorithm is evaluated on a novel MOMARL environment consisting of a task of multi-agent resource distribution. Our novel algorithm achieves better performances on the evaluation metrics when compared to the considered baselines.

4 FAIRNESS WITH EXPECTED SCALARIZED RETURN

In single-objective RL the value of a policy is the expected return obtained from its execution. The natural extension of this paradigm

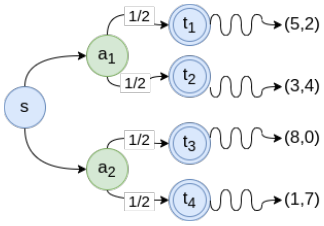

Figure 1: MO-MDP considered in Example 1

to multi-objective reinforcement learning gives rise to the Scalarized Expected Return (SER) criterion. However, in many real-world scenarios, a policy's value is given by one single execution. In such cases, algorithms tailored for SER fail to learn optimal policies, and we have to resort to optimizing ESR. This section demonstrates how using algorithms designed for SER optimization to achieve objective-wise fairness can lead to seemingly unfair policies and explains how ESR optimization can solve this problem. Through this section, we aim to convince readers of the necessity of differentiating between policies that ensure fairness while optimizing SER and those optimizing ESR.

4.1 ESR vs SER for objective-wise fairness in multi-objective single-agent reinforcement learning

Example 1: Consider the MO-MDP[9] shown in Figure 1, with an initial state s and two actions a_1 and a_2 leading to the terminal states t_1, t_2, t_3, t_4 . The transition function is stochastic with $T(s, a_1, t_1) = T(s, a_1, t_2) = T(s, a_2, t_3) = T(s, a_2, t_4) = \frac{1}{2}$,the reward function is a 2-dimensional function of the state and is given by the curvy arrows going out of a state.

Consider the two deterministic policies $\pi_1(s, a_1) = 1$ and $\pi_2(s, a_2) = 1$. Under SER, and using the NSW scalarisation function, the values of these policies are computed as follows:

$$V^{\pi_1}_{NSW} = NSW(\tfrac{1}{2}[(5, 2) + (3, 4)]) = 12$$
$$V^{\pi_2}_{NSW} = NSW(\tfrac{1}{2}[(8, 0) + (1, 7)]) = 15.75$$

Thus under SER, $\pi_2 \succ \pi_1$.

Since the NSW function is non-linear, we can expect that the values of π_1 and π_2 will be different under ESR, these values are given by the following :

$$V^{\pi_1}_{NSW} = \tfrac{1}{2}[NSW(5, 2) + NSW(3, 4)] = 11$$
$$V^{\pi_2}_{NSW} = \tfrac{1}{2}[NSW(8, 0) + NSW(1, 7)] = \tfrac{7}{2}$$

Notice that under ESR not only did the values of the policies change, but their order changed too as under ESR, $\pi_1 \succ \pi_2$.

Example 2: Consider now the MO-MDP shown in Figure 2, with an initial state s_0 , three actions a_0, a_1 and a_2, one non-terminal state s_1 terminal states t_1, t_2 and t_3. The transition function is stochastic with $T(s_0, a_0, s_1) = T(s_1, a_1, t_1) = 1$ and $T(s_1, a_2, t_2) = T(s_1, a_2, t_3) = \frac{1}{2}$. The reward function is a 2-dimensional function of the state given by the curvy arrows going out of a state with $\alpha \in \mathbb{R}^*_+, \epsilon \in \mathbb{R}^*_+$ and $\alpha > \epsilon$.

We compare the deterministic policies π_1 and π_2 defined as $\pi_1(s_0) = \pi_2(s_0) = a_0$, $\pi_1(s_1) = a_1$ and $\pi_2(s_2) = a_2$. Under SER the values of these policies are given by the following:

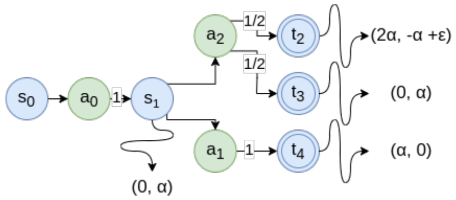

Figure 2: MO-MDP considered in Example 2

$$V^{\pi_1}_{NSW} = NSW(\alpha, \alpha) = \alpha^2$$
$$V^{\pi_2}_{NSW} = NSW(\tfrac{1}{2}[(0, 2\alpha) + (2\alpha, \epsilon)]) = \alpha(\alpha + \tfrac{\epsilon}{2})$$

Under SER, $\forall \epsilon > 0$: $\pi_2 \succ \pi_1$. Thus, under SER the policy π_2 would be selected rather than π_1.

Under ESR, π_1's value remains the same as the one found under SER. However, the value of π_2 changes, and it is given by the following:

$$V^{\pi_2}_{NSW} = \tfrac{1}{2}[NSW(0, 2\alpha) + NSW(2\alpha, \epsilon)) = \alpha\epsilon$$

Under ESR, since $\alpha > \epsilon$, we can see that the preference over policies is switched ($\pi_1 \succ \pi_2$).

We argue that the policy π_1 is the fairest of the two considered policies in both examples.

Example 1 shows that π_1 achieves fairer returns during a single execution (since it has a greater value under ESR) and allows it to remain fair on average of multiple executions, meanwhile policy π_2 only achieves fairness when the average of its returns over multiple executions is considered. Also, in the MDP shown in Figure 1, the returns obtained by the action a_1 are Pareto-dominated by the returns obtained by action a_2 since $(\frac{9}{2}, \frac{7}{2}) \succ_P (\frac{8}{2}, \frac{6}{2})$. However, using NSW under ESR we prefer π_1 to π_2. This shows that NSW does not satisfy the Pareto optimality under ESR, even though [14] showed that NSW satisfies this axiom under SER. We argue that a more suitable dominance property for ESR optimization is the first-order stochastic dominance and its extension proposed by [7].

Example 2 highlights two aspects that are undesirable in a fair policy:

- A fair policy learned by optimizing SER can lead to a more risky policy.
- Even if a policy is conditioned by the reward accumulated by the agent until decision time, SER cannot guarantee fairness. Indeed, when the agent finds itself in s_1 with an accumulated reward of $(0, \alpha)$ SER still prefers to perform action a_2 over a_1 (since $\pi_2 \succ \pi_1$ under SER) even though the fairest action in that situation appears to be a_1.

4.2 ESR vs SER for objective-wise fairness in multi-objective multi-agent reinforcement learning

Consider a multi-objective variant of the prisoner's dilemma presented in the common reward multi-objective matrix game shown in Table 1. The agents receive the same 2-dimensional reward. We compare the values of the deterministic joint policy π_1 that always selects the joint action (b, b) and the stochastic policy π_2 that selects with uniform probability joint actions (a, b) and (b, a).

The values of these policies under SER are the following:

	a	b
a	$(1,1)$	$(0,11)$
b	$(11,0)$	$(5,5)$

Table 1: MO matrix game

$$V^{\pi_1}_{NSW} = NSW(5,5) = 25$$
$$V^{\pi_2}_{NSW} = NSW(\tfrac{1}{2}[(11,0) + (0,11)]) = 30.25$$

Therefore, under SER $\pi_2 \succ \pi_1$. If we instead consider the values of these policies under ESR now, only the values of π_2 are changed, and we obtain:

$$V^{\pi_2}_{NSW} = \tfrac{1}{2}[NSW(11,0) + NSW(0,11))] = 0$$

Consequently, $\pi_1 \succ \pi_2$, where π_1 ensures that the fairest option is selected at each policy execution.

Notice that even though the matrix game considered is deterministic, the stochastic nature of policy π_2 induces a difference in the value of the policy depending on the optimization criterion considered. We thus argue that, in a multi-objective environment whether stochastic or deterministic, the distinction between ESR and SER is necessary if the agents' policies are stochastic and the common scalarisation function is non-linear (ESR and SER are equivalent when the scalarisation function is linear [9]).

5 FAIR MULTI-OBJECTIVE MULTI-AGENT REINFORCEMENT LEARNING UNDER ESR

The previous section explained thoroughly the distinction between ESR and SER. It presents arguments why fairness should be studied under both criteria. It illustrates why it can be misleading to learn an objective-wise fair policy under SER for both the single and multi-agent cases. Motivated by those arguments this section presents a novel decentralized policy-gradient algorithm that is able to learn fair distributed policies under ESR.

5.1 Fully decentralized multi-objective reinforcement learning algorithms

Based on [20], the proposed algorithm is called *Decentralized Expected Utility Policy Gradient (Dec-EUPG)* because it applies the EUPG algorithm on each agent independently. The pseudocode of the algorithm is given in Algorithm 1 with

$$G_t^- = \sum_{k=0}^{t} \gamma^k \mathbf{R}_k \quad \text{and} \quad G_t^+ = \sum_{k=t}^{T-1} \gamma^k \mathbf{R}_k$$

where G_t^- represents the reward accumulated by the agents from the beginning of the episode until timestep t and G_t^+ represents the future reward that the agent will get from timestep t until the end of the episode. To generate an episode from a joint policy $\Pi =< \pi^1, \pi^2, ..., \pi^n >$, at each timestep and given a joint observation $o_t =< o^1, o^2, ..., o^n >$, an agent i selects an action a_t^i based on its policy π^i that is conditioned by o_i and G_t^- the accumulated reward up until t thus $a_t^i \sim \pi^i(s_i, G_t^-)$.

The scalarization function used is the one presented in Section 2.2.

Algorithm 1: Decentralized EUPG

```
Data:
    $tr\_timesteps > 0$ ;        // Number of training timesteps
    $\alpha > 0$ ;               // Learning rate of the algorithm
    $n\_agents > 0$ ;            // number of agents in the problem
    $u : \mathbb{R}^d \to \mathbb{R}$ ;   // scalarisation function
Initialize the policy parameter $\theta_i$ for each agent $i$ at random.;
$steps \leftarrow 0$;
while $steps < tr\_timesteps$ do
    Generate one episode by following the joint policy $\Pi$:
     $\mathbf{S}_0, \mathbf{A}_0, \mathbf{R}_0, \mathbf{S}_1, ..., \mathbf{S}_{T-1}, \mathbf{A}_{T-1}, \mathbf{R}_{T-1}, \mathbf{S}_T$;
    for each agent do
        for $t \in [1, T]$ do
            Estimate $\mathbf{G}_t^+$ and $\mathbf{G}_t^-$ ;
            Update the agent's policy parameters $\theta_i$ as
            $\theta_i \leftarrow \theta_i + \alpha\gamma^t u(\mathbf{G}_t^- + \mathbf{G}_t^+)\nabla_{\theta_i} \ln \pi_{\theta_i}(a_t^i | h_t^i, G_t^-)$;
        end
    end
    $steps \leftarrow steps + T$;
end
```

Note that the resulting policies are only conditioned by the history of local observations of the agent h_t^i and the global reward accumulated G_t^{i-} as suggested in the original EUPG algorithm.

5.2 Local reward decentralized expected utility policy gradient

One of the main limitations of Algorithm 1 is the conditioning on the global accumulated reward as agents often do not have access to this information during execution. To overcome this problem, we propose a variant of this algorithm where the policy of an agent i is instead conditioned on the history h_t^i of her local observations and her local reward accumulated G_t^{i-} where $G_t^{i-} = \sum_{k=0}^{t} \gamma^k \mathbf{R}_k^i$. However, this variant is currently only applicable on scenarios where a local reward can be defined for each agent.

6 EXPERIMENTS AND RESULT

This section presents the evaluation scenarios that the algorithm is evaluated on, the baselines considered to compare our algorithm to, the evaluation metrics used to assess the performances of the agents, and the results obtained. This section concludes with a thorough discussion of the results.

6.1 Evaluation scenarios

We evaluate our approach on a partially observable delivery task where multi-agent coordination is needed to achieve optimal scores on the evaluation criteria. See Appendix A for a complete description of the environment.

The first evaluation scenario is a 2-agents bi-objective task. The agents can deliver resources to two types of households, the number of households of each type present in the environment varies between 3 and 7 whereas each agent carries enough resources to

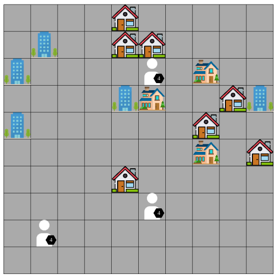

Figure 3: Environment with 3 agents and 3 household types: houses, tower buildings, and villas. The number on the badges of the agents shows the number of households the agents can still accommodate.

accommodate up to 4 houses. We use this setting to validate our approach and prove that decentralized policies can indeed be learned by the agents using our algorithm.

To evaluate the scaling abilities of our agents, we propose a harder setting where 3 agents have to deliver resources to 3 types of households, making it a 3-agents tri-objective task. This task is harder because the agents have the same limitation on the amount of resources they can distribute but more objectives exist in the environment. Figure 3 illustrates this setting with an environment used to train 3 agents to distribute resources across 3 types of households.[1]

6.2 Baselines

After a careful literature review on the problem of fairness in MOMARL, the issue of integrating fairness in MOMARL algorithms under the ESR criterion has not yet received any attention from the community. Consequently, an approach like the one proposed in [10] cannot be used to solve this problem since the fairest policy learned by that algorithm would not guarantee fairness for a singular execution, the same also goes for multi-agent extensions of the algorithms suggested by [23].

Therefore, no existing algorithm can effectively solve the evaluation scenarios. For this reason, we propose to compare our solution to a centralized single-agent solution and a decentralized single-agent single-objective solution.

Centralized single-agent baseline: This baseline assumes that there exists a centralized agent controlling all the agents at once. At each step of an episode, the controller receives the joint observations of the agents, the global vectorial reward received by the agents, and performs a joint action in the environment. This baseline can alleviate problems related to the partial observability of the environment and multi-agent credit assignment, however, it may suffer from exploration issues due to a larger action space.

Single-agent single-objective decomposition baseline: This baseline takes a n-objective n-agent problem where n is both the number of agents and the number of objectives in the problem and decomposes into n single-objective single-agent problems. In this setting, an agent i is assigned to optimize objective i. Each agent learns an independent policy conditioned only by its local observations. Agent i receives a local reward signal consisting of the number of households of type i accommodated during an episode. Even though this baseline solves the multi-agent credit assignment problem and allows the usage of more sophisticated single-objective single-agent RL algorithms, it is only usable in a small subset of MOMARL applications where the number of agents in the system is equal to the number of objectives and each agent can focus on solely optimizing one objective. Real-world problems where all these constraints are fulfilled are limited.

To ensure a fair comparison between our approach and these baselines, we use the EUPG algorithm for the centralized baseline and the policy gradient algorithm for the decomposition baseline.

6.3 Evaluation metrics

In this paper, the agents are expected to provide **efficient** and **fair** solutions toward the objective. Consequently, both aspects are evaluated using a dedicated metric.

Fairness metric: The fairness measure considered in this article is **proportional fairness**. In the fair division literature, an allocation of goods is proportional if each agent values the bundle it received as at least $1/n$ of the utility it associates with receiving all the objects [1]. Note the fair-division literature usually considers fairness among agents but this can still be extended in our setting to fairness among objectives.

This measure is adapted to the task of resource distribution the agents are evaluated on, following three steps:

(1) At the start of an episode e, we define the vector $\mathbf{r}^e_{max} \in \mathbb{N}^*$ as the vector where $\mathbf{r}^e_{max}[i]$ represents the number of households of type i available in the environment during the episode.
(2) We define the best proportion $\mathbf{bp}$ each objective i can achieve under a proportional solution (where $\mathbf{bp}_i = \frac{\mathbf{br}_i}{\mathbf{r}^e_{max}}$ and $\mathbf{br}_i$ is the maximum number of households of type i accommodated under a proportional solution). We compute this best proportion using the algorithm given in Appendix B.
(3) After an episode e, the vectorial reward received by the agents $\mathbf{r}^e$ can be interpreted as an allocation where $\mathbf{r}^e[i]$ is the number of households of type i accommodated by the agents during e. We also define the proportion vector achieved by the agents after an episode as $\mathbf{p}^e = \frac{r^e}{r^e_{max}}$. The score associated with the joint policy of the agents at episode e is given by

$$\min_{i \in \{1,2,\ldots,d\}} \frac{\mathbf{p}^e_i}{\mathbf{bp}_i}.$$

To better understand this metric, let us consider an environment with 3 types of households: houses, tower buildings, and villas. The environment contains 7 houses, 4 tower buildings, and 3 villas; $r^e_{max} = [7, 4, 3]$. Let us consider a 3-agent joint-policy execution that accommodates the needs of 5 houses, 4 buildings, and 3 villas. A system with prior knowledge of the environment can achieve a best minimal proportion value of $\mathbf{bp} = [\frac{6}{7}, \frac{3}{4}, \frac{3}{3}]$. If the policy execution achieves a minimal proportion value of $\mathbf{p}^e = [\frac{5}{7}, \frac{4}{4}, \frac{3}{3}]$,

[1] Code is available as supplementary material along the paper submission

the fairness score of this policy execution is given by:

$$\min_{i\in\{1,2,3\}} \frac{\mathbf{p}_i^e}{\mathbf{bp}_i} = \min\{\frac{5}{6}, \frac{4}{3}, \frac{3}{3}\} = \frac{5}{6}.$$

Efficiency metric: Episode length is used to quantify the efficiency of a learned joint policy. Indeed, the agents are given enough time to solve the resource-distribution task, thus an efficient policy would be expected to accommodate households fairly as quickly as possible.

Experimental pipeline: We train the agents for 10 million timesteps of the environment. The joint policy of the agents is periodically evaluated after 250 thousand timesteps. Since the joint policies learned are usually stochastic each evaluation is conducted on 100 random environments. We log the median, first, and third quartile obtained from the 100 evaluations. To reduce the impact of random initialization on the neural network modeling the policy, the training is repeated on 5 random seeds.

6.4 Results

This section compares the results obtained by each approach considered on both evaluation tasks presented in Section 6.1. Note that the results obtained here were generated with the NSW scalarisation but the results are still valid for the Ordered Weighted Average (OWA) [28] family of functions as long as the weights are decreasing.

6.4.1 Decentralized EUPG vs Centralized EUPG vs Decomposed PG. We first focus on comparing the proposed algorithm against the considered baselines. Figures 4 and 5 show the results obtained by each algorithm on the scenario with 2 agents and 2 objectives whereas Figures 6 and 7 show the results obtained on the task with 3 agents and 3 objectives. From Figure 4, we can see that our algorithm (in blue) reaches the same median minimal proportion reached by the considered baseline (around 90%). On top of that Figure 5 shows that our algorithm solves the task at hand more efficiently since our algorithm can solve the 2-agents bi-objective task in less than 300 timesteps at the end of the training whereas the centralized baseline needs 350 on average and the decomposition baseline needs an average of 400 timesteps.

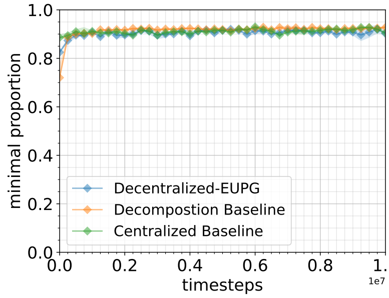

Figure 4: Median minimal proportion attained at 2-agents 2-objectives evaluation

Figures 6 and 7 show the same tendency and prove that our algorithm can achieve the same performance even when the number of agents is increased.

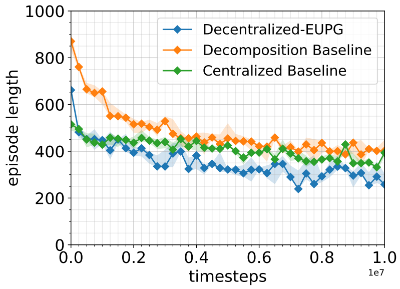

Figure 5: Median episode length required to solve 2-agents 2-objectives evaluation environments

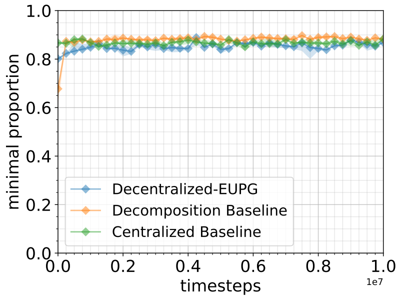

Figure 6: Median minimal proportion attained at 3-agents 3-objectives evaluation

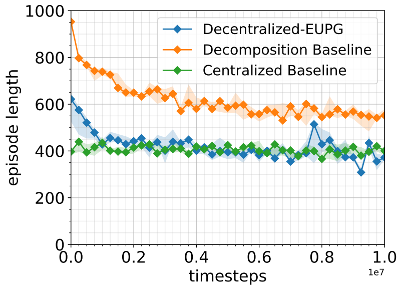

Figure 7: Median episode length required to solve 3-agents 3-objectives evaluation environments

6.4.2 Decentralized EUPG vs Local reward Decentralized EUPG. To make the algorithm deployable, we proposed in Section 5.2 a local reward variant of the algorithm. Figures 8 and 9 show the results of the comparison of the decentralized EUPG algorithm against the local reward variant of the algorithm on the 3-agent 3-objectives evaluation scenario. We can see from Figure 8 that both algorithms converge to the same minimal proportion score (around 90%) and from Figure 9 that the local reward variant of the decentralized EUPG algorithm can solve the environments more efficiently as by the end of the training the local reward variant needs 180 timesteps to solve the task whereas the global reward variant needs 380 timesteps on average.

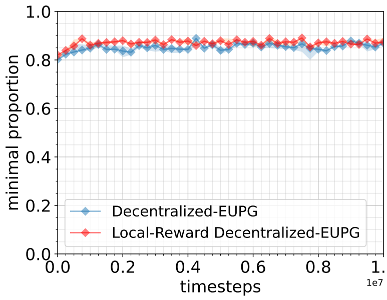

Figure 8: Median minimal proportion attained by Decentralized EUPG and the local reward variant at 3-agents 3-objectives evaluation

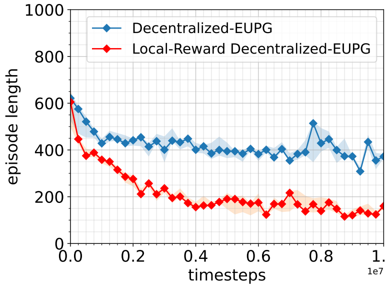

Figure 9: Median episode length required by Decentralized EUPG and the local reward variant to solve 3-agents 3-objectives evaluation environments

6.5 Discussion

This section discusses the results obtained and provides answers to the following questions:

- Why do policies learned using the decomposition baseline require many more steps in the environment to solve the task compared to policies learned under other studied algorithms?
- Why is the local-reward variant more efficient than the original decentralized EUPG algorithm?

Decentralized EUPG a first solution for achieving fairness in MOMARL under ESR: The results presented in Section 6.4 show that the algorithms we propose can solve cooperative multi-agent multi-objective sequential decision-making problems while ensuring fairness towards the objectives. The local-reward variant proved to be more efficient than the global-reward variant while achieving the same fairness. This makes our algorithms more practical and suitable for deployment in real-world applications. We argue that the proposed solutions are most suited for applications where fairness needs to be guaranteed at each execution such as ethical multi-objective reinforcement learning.

Issues with the decomposition baseline: The decomposition baseline achieves the same fairness scores but struggles to solve the task efficiently. Indeed, policies learned by the decomposition baseline always require 200 to 300 hundred more timesteps to solve the resource distribution task fairly. This is because households are sampled uniformly across the environment, once an agent is assigned an objective at the beginning of the training, it can only accommodate that objective. Thus, the agent has to explore the environment more extensively before identifying all the households it needs to accommodate.We argue that this solution can still be viable if the objectives' disposition is known to be clustered geographically . However, this kind of information on the structure of the environment is usually unavailable for RL algorithms. Lastly, we conjecture that this issue can be solved by allowing for multi-agent communication but this is out of the scope of this work.

Local-reward: a recipe for success? Figures 8 and 9, show that the local reward variant of the algorithm outperforms the global reward one on the efficiency metric while achieving comparable results on the fairness metric. Local reward helps solve multi-agent credit-assignment issues allowing the agent to learn more efficiently the consequences of their actions on the environment. This solution however is only feasible when it is possible to transform the reward from global to local. When no such transformation exists, and since an agent cannot have access to the reward accumulated by the other agents during the execution of the policy, one could try to use agent modeling approaches to learn in a centralized manner models of the reward accumulated by the other agents. Providing that the learned model is sufficiently accurate one could expect the same performances as the ones obtained by our local-reward decentralized EUPG algorithm.

Limitations of the Decentralized EUPG algorithm: The proposed algorithm can solve tasks involving a limited number of agents and objectives. However, as an on-policy algorithm, it struggles with sample efficiency. The current version of the algorithm does not leverage a centralized training environment and cannot use bootstrapping since it does not use a critic during the training. These limitations could be addressed in future work.

7 CONCLUSION AND FUTURE WORK

This paper highlights a fundamental flaw in existing fair multi-objective RL algorithms, where current approaches ensure fairness only over multiple policy executions but fail to guarantee fairness across objectives within a single execution of the policy.

To address the identified issue, we introduced a novel RL algorithm inspired by the work of [20], designed to solve multi-agent multi-objective cooperative sequential decision-making problems while ensuring proportional fairness across objectives. Experiments on delivery task scenarios demonstrated that our algorithm achieves similar levels of fairness compared to a centralized approach, where a central controller benefits from greater observability. Additionally, our algorithm requires fewer timesteps to compute solutions, a comparative study with a decomposition-based approach also favored our algorithm, further highlighting its efficiency. Although the proposed approach effectively solves the small-scale problems it was tested on, several limitations were noted, including sample efficiency, challenges in multi-agent credit assignment, and difficulties in deployment when constructing a local reward is not feasible.

Potential future work directions include but are not limited to extending state-of-the-art single-agent MARL algorithms to the multi-objective setting, adapting them for the case of ESR optimization, and proposing inner-loop multi-policy algorithms for ESR optimization for both single and multi-agent applications.

REFERENCES

[1] Sylvain Bouveret, Yann Chevaleyre, and Nicolas Maudet. 2016. *Fair Allocation of Indivisible Goods*. Cambridge University Press, 284–310.

[2] Tarek Chouaki and Sebastian Hörl. 2024. Comparative assessment of fairness in on-demand fleet management algorithms. In *The 12th Symposium of the European Association for Research in Transportation (hEART)*. Espoo, Finland. https://hal.science/hal-04551002

[3] Ioana Alexandra Cimpean, Catholijn M Jonker, Pieter Libin, and Ann Nowe. 2023. A Multi-objective Framework For Fair Reinforcement Learning. 1–11. https://modem2023.vub.ac.be Multi-Objective Decision Making Workshop 2023, MODeM 2023 ; Conference date: 01-10-2023 Through 01-10-2023.

[4] Ziming Fan, Nianli Peng, Muhang Tian, and Brandon Fain. 2023. Welfare and Fairness in Multi-objective Reinforcement Learning. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems* (London, United Kingdom) *(AAMAS '23)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1991–1999.

[5] Florian Felten, Umut Ucak, Hicham Azmani, Gao Peng, Willem Röpke, Hendrik Baier, Patrick Mannion, Diederik M. Roijers, Jordan K. Terry, El-Ghazali Talbi, Grégoire Danoy, Ann Nowé, and Roxana Rădulescu. 2024. MOMAland: A Set of Benchmarks for Multi-Objective Multi-Agent Reinforcement Learning. arXiv:2407.16312 [cs.MA] https://arxiv.org/abs/2407.16312

[6] Jakob N. Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual multi-agent policy gradients. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence* (New Orleans, Louisiana, USA) *(AAAI'18/IAAI'18/EAAI'18)*. AAAI Press, Article 363, 9 pages.

[7] Conor Hayes, Timothy Verstraeten, Diederik Roijers, Enda Howley, and Patrick Mannion. 2022. Expected scalarised returns dominance: a new solution concept for multi-objective decision making. *Neural Computing and Applications* (07 2022). https://doi.org/10.1007/s00521-022-07334-x

[8] Conor F. Hayes, Mathieu Reymond, Diederik M. Roijers, Enda Howley, and Patrick Mannion. 2023. Monte Carlo tree search algorithms for risk-aware and multi-objective reinforcement learning. *Autonomous Agents and Multi-Agent Systems* 37, 2 (April 2023), 37. https://doi.org/10.1007/s10458-022-09596-0

[9] Conor F. Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M. Zintgraf, Richard Dazeley, Fredrik Heintz, Enda Howley, Athirai A. Irissappane, Patrick Mannion, Ann Nowé, Gabriel Ramos, Marcello Restelli, Peter Vamplew, and Diederik M. Roijers. 2022. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems* 36, 1 (April 2022). https://doi.org/10.1007/s10458-022-09552-y

[10] Tianmeng Hu, Biao Luo, Chunhua Yang, and Tingwen Huang. 2023. MO-MIX: Multi-Objective Multi-Agent Cooperative Decision-Making With Deep Reinforcement Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 10 (2023), 12098–12112. https://doi.org/10.1109/TPAMI.2023.3283537

[11] Jiechuan Jiang and Zongqing Lu. 2019. *Learning fairness in multi-agent systems*. Curran Associates Inc., Red Hook, NY, USA.

[12] Peizhong Ju, Arnob Ghosh, and Ness Shroff. 2024. Achieving Fairness in Multi-Agent MDP Using Reinforcement Learning. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=yoVq2BGQdP

[13] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 6382–6393.

[14] Debmalya Mandal and Jiarui Gan. 2022. Socially fair reinforcement learning. *arXiv preprint arXiv:2208.12584* (2022).

[15] Patrick Mannion, Sam Devlin, Jim Duggan, and Enda Howley. 2018. Reward shaping for knowledge-based multi-objective multi-agent reinforcement learning. *The Knowledge Engineering Review* 33 (2018), e23. https://doi.org/10.1017/S0269888918000292

[16] Frans A Oliehoek, Christopher Amato, et al. 2016. *A concise introduction to decentralized POMDPs*. Vol. 1. Springer.

[17] Leonid Peshkin, Kee-Eung Kim, Nicolas Meuleau, and Leslie Pack Kaelbling. 2000. Learning to cooperate via policy search. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence* (Stanford, California) *(UAI'00)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 489–496.

[18] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2020. Monotonic value function factorisation for deep multi-agent reinforcement learning. *J. Mach. Learn. Res.* 21, 1, Article 178 (Jan. 2020), 51 pages.

[19] Mathieu Reymond, Conor Hayes, Denis Steckelmacher, Diederik Roijers, and Ann Nowe. 2023. Actor-critic multi-objective reinforcement learning for non-linear utility functions. *Autonomous Agents and Multi-Agent Systems* 37 (04 2023). https://doi.org/10.1007/s10458-023-09604-x

[20] Diederik M. Roijers, Denis Steckelmacher, and Ann Nowé. 2020. Multi-objective reinforcement learning for the expected utility of the return. 2018 Adaptive Learning Agents, ALA 2018 - Co-located Workshop at the Federated AI Meeting, FAIM 2018 ; Conference date: 14-07-2018 Through 15-07-2018.

[21] D. M. Roijers, P. Vamplew, S. Whiteson, and R. Dazeley. 2013. A Survey of Multi-Objective Sequential Decision-Making. *Journal of Artificial Intelligence Research* 48 (Oct. 2013), 67–113. https://doi.org/10.1613/jair.3987

[22] Roxana Rădulescu, Patrick Mannion, Diederik M. Roijers, and Ann Nowé. 2019. Multi-objective multi-agent decision making: a utility-based analysis and survey. *Autonomous Agents and Multi-Agent Systems* 34, 1 (Dec. 2019). https://doi.org/10.1007/s10458-019-09433-x

[23] Umer Siddique, Paul Weng, and Matthieu Zimmer. 2020. Learning fair policies in multiobjective (deep) reinforcement learning with average and discounted rewards. In *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*. JMLR.org, Article 826, 11 pages.

[24] Harold Soh and Yiannis Demiris. 2011. Evolving policies for multi-reward partially observable markov decision processes (MR-POMDPs). In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation* (Dublin, Ireland) *(GECCO '11)*. Association for Computing Machinery, New York, NY, USA, 713–720. https://doi.org/10.1145/2001576.2001674

[25] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. 2019. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International conference on machine learning*. PMLR, 5887–5896.

[26] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. 2018. Value-Decomposition Networks For Cooperative Multi-Agent Learning Based On Team Reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems* (Stockholm, Sweden) *(AAMAS '18)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2085–2087.

[27] Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. 2021. QPLEX: Duplex Dueling Multi-Agent Q-Learning. arXiv:2008.01062 [cs.LG] https://arxiv.org/abs/2008.01062

[28] R.R. Yager. 1988. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man, and Cybernetics* 18, 1 (1988), 183–190. https://doi.org/10.1109/21.87068