On Minimizing Ordered Weighted Regrets in Multiobjective Markov Decision Processes

Wlodzimierz Ogryczak[1], Patrice Perny[2], and Paul Weng[2]

[1] ICCE, Warsaw University of Technology, Warsaw, Poland
wogrycza@elka.pw.edu.pl
[2] LIP6 - UPMC, Paris, France
{patrice.perny,paul.weng}@lip6.fr

Abstract. In this paper, we propose an exact solution method to generate fair policies in Multiobjective Markov Decision Processes (MMDPs). MMDPs consider n immediate reward functions, representing either individual payoffs in a multiagent problem or rewards with respect to different objectives. In this context, we focus on the determination of a policy that fairly shares regrets among agents or objectives, the regret being defined on each dimension as the opportunity loss with respect to optimal expected rewards. To this end, we propose to minimize the ordered weighted average of regrets (OWR). The OWR criterion indeed extends the minimax regret, relaxing egalitarianism for a milder notion of fairness. After showing that OWR-optimality is state-dependent and that the Bellman principle does not hold for OWR-optimal policies, we propose a linear programming reformulation of the problem. We also provide experimental results showing the efficiency of our approach.

Keywords: Ordered Weighted Regret, Fair Optimization, Multiobjective MDP.

1 Introduction

Markov Decision Process (MDP) is a standard model for planning problems under uncertainty [15,10]. This model admits various extensions developed to address different questions that emerge in applications of Operations Research and Artificial Intelligence, depending on the structure of state space, the definition of actions, the representation of uncertainty, and the definition of preferences over policies. We consider here the latter point. In the standard model, preferences over actions are represented by immediate rewards represented by scalar numbers. The value of a sequence of actions is defined as the sum of these rewards and the value of a policy as the expected discounted reward. However, there are various contexts in which the value of a sequence of actions is defined using several reward functions. It is the case in multiagent planning problems [2,7] where every agent has its own value system and its own reward function. It is also the case of multiobjective problems [1,13,3], for example path-planning problems under uncertainty when one wishes to minimize length, time, energy consumption

R.I. Brafman, F. Roberts, and A. Tsoukiàs (Eds.): ADT 2011, LNAI 6992, pp. 190–204, 2011.

and risk simultaneously. In all these problems, n distinct reward functions need to be considered. In general, they cannot be reduced to a single reward function even if each of them is additive over sequences of actions, and even if the value of a policy can be synthesized into a scalar overall utility through an aggregation function (except for linear aggregation). This is why we need to develop specific approaches to determine compromise solutions in Multiobjective or Multiagent MDPs.

Many studies on Multiobjective MDPs (MMDP) concentrate on the determination of the entire set of Pareto-optimal solutions, i.e., policies having a reward vector that cannot be improved on a component without being downgraded on another one. However, the size of the Pareto set is often very large due to the combinatorial nature of the set of deterministic policies, its determination induces prohibitive response times and requires very important memory space as the number of states and/or criteria increases. Fortunately, there is generally no need to determine the entire set of Pareto-optimal policies, but only specific compromise policies achieving a well-balanced tradeoff between criteria or equivalently, in a multiagent context, policies that fairly shares expected rewards among agents. Motivated by such examples, we study in this paper the determination of fair policies in MMDPs. To this end, we propose to minimize the ordered weighted average of regrets (OWR). The OWR criterion indeed extends the minimax regret, relaxing egalitarianism on regrets for a milder notion of fairness.

The paper is organized as follows: In Section 2, we recall the basic notions related to Markov decision processes and their multiobjective extension. In Section 3, we discuss the choice of a scalarizing function to generate fair solutions. This leads us to adopt the ordered weighted regret criterion (OWR) as a proper scalarizing function to be minimized. Section 4 is devoted to the search of OWR-optimal policies. Finally, Section 5 presents some experimental results showing the effectiveness of our approach for finding fair policies.

2 Background

A *Markov Decision Process* (MDP) [15] is described as a tuple (S, A, T, R) where S is a finite set of states, A is a finite set of actions, transition function $T(s, a, s')$ gives the probability of reaching state s' by executing action a in state s, reward function $R(s, a) \in \mathbb{R}$ gives the immediate reward obtained for executing action a in state s.

In this context, a *decision rule* δ is a procedure that determines which action to choose in each state. A decision rule can be *deterministic*, i.e., defined as $\delta : S \rightarrow A$, or more generally, *randomized*, i.e., defined as $\delta : S \rightarrow \mathbf{Pr}(A)$ where $\mathbf{Pr}(A)$ is the set of probability distributions over A.

A *policy* π is a sequence of decision rules $(\delta_0, \delta_1, \ldots, \delta_t, \ldots)$ that indicates which decision rule to apply at each step. It is said to be *deterministic* if each decision rule is deterministic and *randomized* otherwise. If the same decision rule δ is applied at each step, the policy is said *stationary* and is denoted δ^∞.

The value of a policy π is defined by a function $v^\pi : S \rightarrow \mathrm{I\!R}$, called *value function*, which gives the expected discounted total reward yielded by applying π from each initial state. For $\pi = (\delta_0, \delta_1, \ldots, \delta_t, \ldots)$, they are given $\forall h > 0$ by:

$$\begin{array}{ll} v_0^\pi(s) = 0 & \forall s \in S \\ v_t^\pi(s) = R(s, \delta_{h-t}(s)) + \gamma \sum\limits_{s' \in S} T(s, \delta_{h-t}(s), s') v_{t-1}^\pi(s') & \forall s \in S, \forall t = 1, \ldots, h \end{array}$$

where $\gamma \in [0, 1[$ is the discount factor. This sequence converges to the value function of π.

In this framework, there exists an optimal stationary policy that yields the best expected discounted total reward in each state. Solving an MDP amounts to finding one of those policies and its associated value function. The optimal value function $v^* : S \rightarrow \mathrm{I\!R}$ can be determined by solving the *Bellman equations*:

$$\forall s \in S, \quad v^*(s) = \max_{a \in A} R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') v^*(s')$$

There are three main approaches for solving MDPs. Two are based on dynamic programming: value iteration and policy iteration. The third is based on linear programming. We recall the last approach as it is needed for the exposition of our results. The linear program $(\mathcal{P})$ for solving MDPs can be written as follows:

$$(\mathcal{P}) \begin{cases} \min \sum\limits_{s \in S} \mu(s) v(s) \\ \text{s.t. } v(s) - \gamma \sum\limits_{s' \in S} T(s, a, s') v(s') \geq R(s, a) \quad \forall s \in S, \forall a \in A \end{cases}$$

where weights μ could be interpreted as the probability of starting in a given state. Any positive μ can in fact be chosen to determine the optimal value function. Program $\mathcal{P}$ is based on the idea that the Bellman equations imply that functions satisfying the constraints of $\mathcal{P}$ are upper bounds of the optimal value function. Writing the dual $(\mathcal{D})$ of this program is interesting as it uncovers the dynamic of the system:

$$(\mathcal{D}) \begin{cases} \max \quad \sum\limits_{s \in S} \sum\limits_{a \in A} R(s, a)\, x_{sa} \\ \left.\begin{array}{l} \text{s.t. } \sum\limits_{a \in A} x_{sa} - \gamma \sum\limits_{s' \in S} \sum\limits_{a \in A} T(s', a, s)\, x_{s'a} = \mu(s) \quad \forall s \in S \\ \quad x_{sa} \geq 0 \quad \forall s \in S, \forall a \in A \end{array}\right\} (\mathcal{C}) \end{cases}$$

To interpret variables x_{sa}, we recall the following two propositions relating feasible solutions of $\mathcal{D}$ to stationary randomized policies in the MDP [15].

Proposition 1. *For a policy π, if x^π is defined as $x^\pi(s, a) = \sum_{t=0}^{\infty} \gamma^t p_t^\pi(s, a)$, $\forall s \in S, \forall a \in A$ where $p_t^\pi(s, a)$ is the probability of reaching state s and choosing a at step t, then x^π is a feasible solution of $\mathcal{D}$.*

Proposition 2. *If x_{sa} is a solution of $\mathcal{D}$, then the stationary randomized policy δ^∞, defined by $\delta(s,a) = x_{sa}/\sum_{a' \in A} x_{sa'}, \forall s \in S, \forall a \in A$ defines $x^{\delta^\infty}(s,a)$ as in Proposition 1, that are equal to x_{sa}.*

Thus, the set of randomized policies is completely characterized by constraints $(\mathcal{C})$. Besides, the basic solutions of $\mathcal{D}$ correspond to deterministic policies. Moreover, the basic solutions of $\mathcal{P}$ correspond to the value functions of deterministic policies. Those of randomized policies are in the convex hull of those basic solutions. Note that in an MDP, any feasible value function can be obtained with a randomized policy.

Multiobjective MDP. MDPs have been extended to take into account multiple dimensions or criteria. A *multiobjective MDP* (MMDP) is an MDP where the reward function is redefined as: $R\colon S \times A \to \mathrm{I\!R}^n$ where n is the number of objectives, $R(s,a) = (R_1(s,a), \ldots, R_n(s,a))$ and $R_i(s,a)$ is the immediate reward for objective $i \in O = \{1, \ldots, n\}$.

Now, a policy π is valued by a value function $V^\pi : S \to \mathrm{I\!R}^n$, which gives the expected discounted total reward vector in each state. To compare the value of policies in a given state s, the basic model adopted in most previous studies [5,17,18] is *Pareto dominance* defined as follows:

$$\forall x, y \in \mathrm{I\!R}^n, x \succ_P y \text{ iff } [x \neq y \text{ and } \forall i \in O, x_i \geq y_i] \tag{1}$$

Hence, for any two policies π, π', π is preferred to π' in a state s if and only if $V^\pi(s) \succ_P V^{\pi'}(s)$. For a set $X \subset \mathrm{I\!R}^n$, a vector $x \in X$ is said to be *Pareto-optimal* in X if there is no $y \in X$ such that $y \succ_P x$. Due to the incompleteness of Pareto dominance, there may exist several Pareto-optimal vectors in a given state.

Standard methods for MDPs can be extended to solve MMDPs [18,17]. As shown by Viswanathan et al. [17], the dual linear program $(\mathcal{D})$ can be extended to a multiobjective linear program for finding Pareto-optimal solutions in a MMDP since the dynamics of a MDP and that of a MMDP are identical. Thus, we obtain the following multiobjective linear program $v\mathcal{D}$:

$$(v\mathcal{D}) \begin{cases} \max f_i(x) = \sum_{s \in S} \sum_{a \in A} R_i(s,a)\, x_{sa} & \forall i = 1, \ldots, n \\ \text{s.t. } (\mathcal{C}) \end{cases}$$

Looking for all Pareto-optimal solutions can be difficult and time-consuming as there are instances of problems where the number of Pareto-optimal value functions of deterministic policies is exponential in the number of states [8].

Besides, in practice, one is generally only interested in specific compromise solutions among Pareto-optimal solutions achieving interesting tradeoffs between objectives. To this end, one could try to optimize one of the objectives subject to constraints over the other objectives (see for instance [1]). However, this approach reveals to be cumbersome to reach well-balanced tradeoffs, as the number of objectives grows. A more natural approach for that could be to use a *scalarizing function* $\psi : \mathrm{I\!R}^n \to \mathrm{I\!R}$, monotonic with respect to Pareto dominance, that defines the value v^π of a policy π in a state s by: $v^\pi(s) = \psi(V_1^\pi(s), \dots, V_n^\pi(s))$. The problem can then be reformulated as the search for a policy π optimizing $v^\pi(s)$ in an initial state s. We discuss now about a proper choice of ψ in order to achieve a fair satisfaction of objectives.

3 Fair Regret Optimization

Weighted Sum. The most straightforward choice for ψ seems to be *weighted sum* (WS), i.e., $\forall y \in \mathbb{R}^n$, $\psi(y) = \lambda \cdot y$ where $\lambda \in \mathbb{R}^n_+$. By linearity of WS and that of mathematical expectation, optimizing v is equivalent to solving the standard MDP obtained from the MMDP where the reward function is defined as: $r(s,a) = \lambda \cdot R(s,a), \forall s, a$. In that case, an optimal stationary deterministic policy exists and standard solution methods can then be applied. However, using WS is not a good procedure for reaching balanced solutions as weighted sum is a fully compensatory operator. For example, with WS, $(5,5)$ would never be stricly preferred to $(10,0)$ and $(0,10)$ simultaneously, whatever the weights.

MaxMin. In opposition to the previous utilitarist approach, we could adopt egalitarianism that consists in maximizing the value of the least satisfied objective ($\psi = \min$). This approach obviously includes an idea of fairness as for example, here, $(5,5)$ is strictly preferred to both $(10,0)$ and $(0,10)$. However, it has two significant drawbacks: (i) min does not take into account the potentialities of each objective with respect to the maximum values that each objective can achieve. For instance, if objective 1 can reach a maximum of 10 while objective 2 can reach a maximum of 6, a solution leading to $(6,6)$ might be seemed less fair than another valued by $(8,4)$ since the second better distributes the opportunity losses; (ii) reducing a vector to its worst component is too pessimistic and creates drowning effects, i.e., $(1,0)$ is seen as equivalent to $(10,0)$, whereas the latter Pareto-dominates the former.

Minmax Regret. A standard answer to (i) is to consider *Minmax regret* (MMR), which is defined as follows. Let Y be a set of valuation vectors in $\mathbb{R}^n$ and $I \in \mathbb{R}^n$ denote the ideal point defined by $I_i = \sup_{y \in Y} y_i$ for all $i \in O$. The regret of choosing $y \in Y$ according to objective i is defined by $\eta_i = I_i - y_i$. Then, MMR is defined for all $y \in Y$ by $\psi(y) = \max_{i \in O}(\eta_i)$. However, MMR does not address issue (ii). In order to guarantee the Pareto monotonicity, MMR may be further generalized to take into account all the regret values according to the Ordered Weighted Average (OWA) aggregation [19], thus using the following scalarizing function [20]:

$$\rho_w(y) = \sum_{i \in O} w_i \eta_{\langle i \rangle} \tag{2}$$

where $(\eta_{\langle 1 \rangle}, \eta_{\langle 2 \rangle}, \dots, \eta_{\langle n \rangle})$ denotes the vector obtained from the regret vector η by rearranging its components in the non-increasing order (i.e., $\eta_{\langle 1 \rangle} \geq \eta_{\langle 2 \rangle} \geq \dots \geq \eta_{\langle n \rangle}$ and there exists a permutation τ of set O such that $\eta_{\langle i \rangle} = \eta_{\tau(i)}$ for $i \in O$) and weights w_i are non-negative and normalized to meet $\sum_{i \in O} w_i = 1$.

Example 1. *We illustrate how ρ_w is computed (see Table 1) with ideal point $I = (9, 7, 6)$ and weights $w = (1/2, 1/3, 1/6)$. One first computes the regrets η, then reorders them. Finally, ρ_w can be computed, inducing the preference order $x \succ z \succ y$.*

Table 1. Example of computation of ρ_w

	1	2	3	η_1	η_2	η_3	$\eta_{\langle 1\rangle}$	$\eta_{\langle 2\rangle}$	$\eta_{\langle 3\rangle}$	ρ_w
x	8	4	5	1	3	1	3	1	1	12/6
y	9	2	6	0	5	0	5	0	0	15/6
z	6	7	4	3	0	2	3	2	0	13/6

Note that ρ_w is a symmetric function of regrets. Indeed, weights w_i's are assigned to the specific positions within the ordered regret vector rather than to the individual regrets themselves. These rank-dependent weights allow to control the importance attached to small or large regrets. For example, if $w_1 = 1$ and $w_2 = \ldots = w_n = 0$, one can recognize the standard MMR, which focuses on the worst regret.

Augmented Tchebycheff norm. This criterion, classically used in multiobjective optimization [16], is defined by $\psi(y) = \max_{i\in O} \eta_i + \epsilon \sum_{i\in O} \eta_i$ where ϵ is a small positive real. It addresses issues (i) and (ii). However, it has some drawbacks as soon as $n \geq 3$. Indeed, when several vectors have the same max regret, then they are discriminated with a weighted sum, which does not provide any control on fairness.

Ordered Weighted Regret. In order to convey an idea of fairness, we now consider the subclass of scalarizing functions defined by Equation (2) with the additional constraints: $w_1 > \ldots > w_n > 0$. Any function in this subclass is named *Ordered Weighted Regret* (OWR) in the sequel. This additional constraint on weights can easily be explained by the following two propositions:

Proposition 3. $[\forall y, z \in \mathbb{R}^n, y \succ_P z \Rightarrow \rho_w(y) < \rho_w(z)] \Leftrightarrow \forall i \in O, w_i > 0$

Proposition 4. $\big[\forall y \in \mathbb{R}^n, \forall i, k \in O, \forall \varepsilon,$ *s.t.* $0 < \varepsilon < \eta_k - \eta_i,$
$\rho_w(y_1, \ldots, y_i - \varepsilon, \ldots, y_k + \varepsilon, \ldots, y_n) < \rho_w(y_1, y_2, \ldots, y_n)\big] \Leftrightarrow w_1 > \ldots > w_n > 0.$

Proposition 3 states that OWR is Pareto-monotonic. It follows from monotonicity of the OWA aggregation [11]. Consequently, OWR-optimal solutions are Pareto-optimal. Proposition 4 is the Schur-convexity of ρ_w, a key property in inequality measurement [12], and it follows from the Schur-convexity of the OWA aggregation with monotonic weights [9]. In MMDPs, it says that a reward transfer reducing regret inequality, i.e., a transfer of any small reward from an objective to any other objective whose regret is greater, results in a preferred valuation vector (a smaller OWR value). For example, if $w = (3/5, 2/5)$ and $I = (10, 10)$, $\rho_w(5, 5) = 5$ whereas $\rho_w(10, 0) = \rho_w(0, 10) = 6$, which means that $(5, 5)$ is preferred to the two others. Due to Proposition 4, if x is an OWR-optimal solution, x cannot be improved by any reward transfer reducing regret inequality, thus ensuring the fairness of OWR-optimal solutions.

Due to Propositions 3 and 4, minimizing OWR leads to a Pareto-optimal solution that fairly distributes regrets over the objectives (see the left part of Figure 1). Moreover, whenever the objectives (criteria or agents) do not have the same importance, it is possible to break the symmetry of OWR by introducing scaling factors $\lambda_i > 0, \forall i \in O$ in Equation (2) so as to deliberately deliver biased (Pareto-optimal) compromise solutions (see the right part of Figure 1). To this end, we generalize OWR by considering:

$$\rho_w^\lambda(y) = \sum_{i \in O} w_i \eta^\lambda_{\langle i \rangle} \quad \text{with} \quad \eta_i^\lambda = \lambda_i (I_i - y_i) \quad \forall\ i \in O \tag{3}$$

where $\lambda = (\lambda_1, \ldots, \lambda_n)$ and $(\eta^\lambda_{\langle 1 \rangle}, \eta^\lambda_{\langle 2 \rangle}, \ldots, \eta^\lambda_{\langle n \rangle})$ denotes the vector obtained from the scaled regret vector η^λ by rearranging its components in the non-increasing order. For the sake of simplicity, ρ_w^λ is also called an OWR.

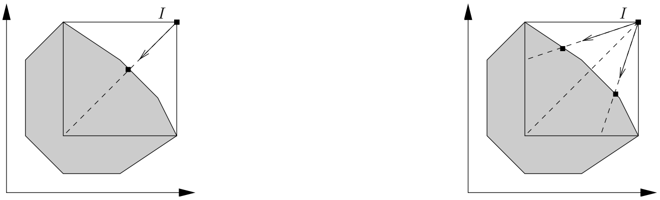

Fig. 1. *Fair* (left) and *biased* (right) compromises

Using OWR, a policy π is weakly preferred to a policy π' in a state s (denoted $\pi \succsim_s \pi'$) iff $\rho_w^\lambda(V^\pi(s)) \leq \rho_w^\lambda(V^{\pi'}(s))$. Hence, an optimal policy π^* in s can be found by solving:

$$v^{\pi^*}(s) = \min_\pi \rho_w^\lambda(V^\pi(s)). \tag{4}$$

As a side note, ρ_w^λ can be used to explore interactively the set of Pareto solutions by solving problem (4) for various scaling factors λ_i and a proper choice of OWR weights w_i. Indeed, we have:

Proposition 5. *For any polyhedral compact feasible set $F \subset \mathbb{R}^n$, for any feasible Pareto-optimal vector $\bar{y} \in F$ such that $\bar{y}_i < I_i, \forall i \in O$, there exist weights $w_1 > \ldots > w_n > 0$, and scaling factors $\lambda_i > 0, \forall i \in O$ such that $\bar{y}$ is a ρ_w^λ-optimal solution.*

Proof. Let $\bar{y} \in F$ be a feasible Pareto-optimal vector such that $\bar{y}_i < I_i, \forall i \in O$. Since, F is a polyhedral compact feasible set, there exists $\Delta > 0$ such that for any feasible vector $y \in F$ the implication

$$y_i > \bar{y}_i \ \text{ and } \ y_k < \bar{y}_k \Rightarrow (y_i - \bar{y}_i)/(\bar{y}_k - y_k) \leq \Delta \tag{5}$$

is valid for any $i, k \in O$ [6].

Let us set the scaling factors $\lambda_i = 1/(I_i - \bar{y}_i), \forall i \in O$ and define weights $w_1 > \ldots > w_n > 0$ such that $w_1 \geq L\Delta\sum_{i=1}^{n} w_i$, where $L \geq \lambda_i/\lambda_k$ for any $i, k \in O$. We will show that $\bar{y}$ is a ρ_w^λ-optimal solution.

Suppose there exists a feasible vector $y \in F$ with better OWR value $\rho_w^\lambda(\bar{y}) = \sum_{i \in O} w_i \bar{\eta}_{\langle i \rangle}^\lambda < \sum_{i \in O} w_i \eta_{\langle i \rangle}^\lambda = \rho_w^\lambda(y)$. Note that $\bar{\eta}_i^\lambda = \lambda_i(I_i - \bar{y}_i) = 1$ for all $i \in O$. Hence, $\eta_{\langle i \rangle}^\lambda - \bar{\eta}_{\langle i \rangle}^\lambda = \eta_{\tau(i)}^\lambda - \bar{\eta}_{\tau(i)}^\lambda$ for all $i \in O$ where τ is the ordering permutation for the regret vector η^λ with $\eta_i^\lambda = \lambda_i(I_i - y_i) = 1$ for $i \in O$. Moreover, $\bar{\eta}_{\tau(i)}^\lambda - \eta_{\tau(i)}^\lambda = \lambda_{\tau(i)}(y_{\tau(i)} - \bar{y}_{\tau(i)})$ and, due to Pareto-optimality of $\bar{y}$, $0 > \bar{\eta}_{\tau(1)}^\lambda - \eta_{\tau(1)}^\lambda = \lambda_{\tau(1)}(y_{\tau(1)} - \bar{y}_{\tau(1)})$. Thus, taking advantages of inequalities (5) for $k = \tau(1)$ one gets

$$\sum_{i=2}^{m} w_i \lambda_{\tau(i)}(y_{\tau(i)} - \bar{y}_{\tau(i)}) \leq -\sum_{i=2}^{m} w_i L\Delta\lambda_{\tau(1)}(y_{\tau(1)} - \bar{y}_{\tau(1)}) \leq -w_1\lambda_{\tau 1}(y_{\tau(1)} - \bar{y}_{\tau(1)})$$

which contradicts to the inequality $\sum_{i \in O} w_i \bar{\eta}_{\langle i \rangle}^\lambda < \sum_{i \in O} w_i \eta_{\langle i \rangle}^\lambda$ and thereby it confirms ρ_w^λ-optimality of $\bar{y}$. ■

Note that the condition $\bar{y}_i < I_i, \forall i \in O$ is not restrictive in practice: one can replace I_i by $I_i + \epsilon$ for any arbitrary small positive ϵ to extend the result to any $\bar{y}$ in F.

4 Solution Method

We now address the problem of solving problem (4). First, remark that, for all scalarizing functions considered in the previous section (apart from WS), finding an optimal policy in an MMDP cannot be achieved by aggregating first the immediate vectorial rewards and solving the resulting MDP. Optimizing OWR implies some subtleties that we present now.

Randomized Policies. When optimizing OWR, searching for a solution among the set of stationary deterministic policies may be suboptimal. Let us illustrate this point on an example where $n = 2$. Assume that points on Figure 2 represent the value of deterministic policies in a given state. The Pareto-optimal solutions are then a, b, c and d. If we were searching for a fair policy, we could consider c as a good candidate solution. However, by considering also randomized policies, we could obtain an even better solution. Indeed, the valuation vectors of randomized policies are in the convex hull of the valuation vectors of deterministic policies, represented by the light-greyed zone (Figure 3). The dotted lines linking points a, b and d represent all Pareto-optimal valuation vectors. The dark greyed zone represents all feasible valuation vectors that are preferred to point c. Those vectors that are Pareto-optimal seem to be good candidate solutions. Therefore, we will not restrict ourselves to deterministic policies and we will consider any feasible randomized policy.

OWR-Optimality is State-Dependent. Contrary to standard MDPs where optimal policies are optimal in every initial state, the optimality notion based on

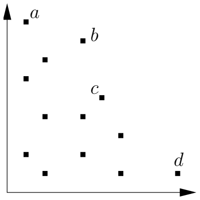

Fig. 2. Valuation vectors

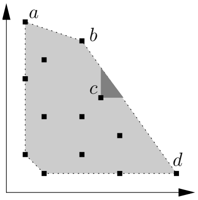

Fig. 3. Better solutions

OWR depends on the initial state, i.e., an OWR-optimal policy in a given initial state may not be an OWR-optimal solution in another state.

Example 2. *Consider the deterministic MMDP represented on Figure 4 with two states ($S = \{1, 2\}$) and two actions ($A = \{a, b\}$). The vectorial rewards can be read on Figure 4.*

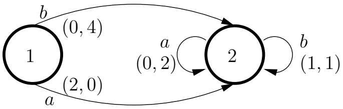

Fig. 4. Representation of the MMDP

Set $\gamma = 0.5$, $w = (0.9, 0.1)$ and $\lambda = (1, 1)$. The ideal point from state 1 is $I_1 = (3, 6)$. Reward 3 is obtained by first choosing a in state 1 and then repeatedly b in state 2 while reward 6 is obtained by first choosing b in state 1 and then repeatedly a in state 2. By similar computations, the ideal point from state 2 is $I_2 = (2, 4)$. There are four stationary deterministic policies, denoted δ_{xy}, which consists in choosing action x in state 1 and action y in state 2.

The OWR-optimal policies in state 2 are δ^∞_{aa} and δ^∞_{ba} with the same value in state 2: $V^{\delta^\infty_{aa}}(2) = V^{\delta^\infty_{ba}}(2) = (0, 4)$ (OWR of 1.8 with I_2). One can indeed check that no randomized policy can improve this score. However, none of these policies are optimal in state 1 as they are beaten by δ^∞_{bb}. Indeed, $V^{\delta^\infty_{bb}}(1) = (1, 5)$ (OWR of 1.9 with I_1) whereas $V^{\delta^\infty_{aa}}(1) = (2, 2)$ (OWR of 3.7 with I_1) and $V^{\delta^\infty_{ba}}(1) = (0, 6)$ (OWR of 2.7 with I_1). This shows that a policy that is optimal when viewed from one state is not necessarily optimal when viewed from another.

Therefore the OWR-optimality is state-dependent.

Violation of the Bellman Optimality Principle. The Bellman Optimality Principle, which says that any subpolicy of any optimal policy is optimal is not guaranteed to be valid anymore when optimizing OWR as it is not a linear scalarizing function. We illustrate this point on Example 2.

Example 2 (continued). *We have $V^{\delta^\infty_{aa}}(1) = (2, 2)$ (OWR of 3.7) and $V^{\delta^\infty_{ab}}(1) = (3, 1)$ (OWR of 4.5). Thus, $\delta^\infty_{aa} \succ_1 \delta^\infty_{ab}$ (seen from state 1). Now, if we consider policy $(\delta_{bb}, \delta^\infty_{aa})$ and policy $(\delta_{bb}, \delta^\infty_{ab})$ that consist in applying δ_{bb} first, then policy*

δ_{aa}^{∞} or policy δ_{ab}^{∞} respectively, we get $V^{(\delta_{bb},\delta_{aa}^{\infty})}(1) = (0,6)$ (OWR of 2.7) and $V^{(\delta_{bb},\delta_{ab}^{\infty})}(1) = (1,5)$ (OWR of 1.9). This means that now $(\delta_{bb},\delta_{aa}^{\infty}) \prec_1 (\delta_{bb},\delta_{ab}^{\infty})$, which is a preference reversal. The Bellman Optimality principle is thus violated.

As shown by Example 2, $\pi \succ_s \pi'$ does not imply $(\delta,\pi) \succsim_s (\delta,\pi')$ for every π,π',δ,s. So, in policy iteration, we cannot prune policy π' on the argument it is beaten by π since π' may lead to an optimal policy (δ,π'). Similar arguments explain that a direct adaptation of value iteration for OWR optimization may fail to find the optimal policy.

The above observations constitute the deadlock to overcome to be able to find efficiently OWR-optimal solutions. This motivates us to propose a solving method based on linear programming.

Solution Method. In order to use OWR in MMDPs, we first compute the ideal point I by setting I_i as the optimal value of $\mathcal{P}$ with reward function R_i.

Although OWR is not linear, its optimization in MMDPs does not impact the dynamic of the system, which thus remains linear. Therefore, OWR is optimized under the same constraints as Program $(v\mathcal{D})$, which gives the following program $(\mathcal{D}')$:

$$(\mathcal{D}')\begin{cases} \min \displaystyle\sum_{i\in O} w_i \eta^{\lambda}_{\langle i\rangle} \\ \left.\begin{array}{l} \text{s.t. } \eta^{\lambda}_i = \lambda_i\big(I_i - \displaystyle\sum_{s\in S}\sum_{a\in A} R_i(s,a)\, x_{sa}\big) \quad \forall i\in O \\ \displaystyle\sum_{a\in A} x_{sa} - \gamma \sum_{s'\in S}\sum_{a\in A} T(s',a,s)\, x_{s'a} = \mu(s) \quad \forall s\in S \\ x_{sa}\geq 0 \quad \forall s\in S, \forall a\in A \end{array}\right\}(\mathcal{C}') \end{cases}$$

where for all $i\in O$, I_i is computed by optimizing objective i with Program $(\mathcal{P})$ or Program $(\mathcal{D})$. Since OWR is not linear but only piecewise-linear (one piece per permutation of objectives), a linear reformulation of $(\mathcal{D}')$ can be written.

First, denoting $L_k(\eta^{\lambda}) = \sum_{i=1}^{k}\eta^{\lambda}_{\langle i\rangle}$ and $w'_i = w_i - w_{i+1}$ for $i = 1,\ldots,n-1$, $w'_n = w_n$, $(\mathcal{D}')$ can be rewritten as:

$$\min_{\eta^{\lambda}\in E}\sum_{k\in O} w'_k L_k(\eta^{\lambda}) \tag{6}$$

where E is defined by Constraints $(\mathcal{C}')$. Moreover, as shown by [14], the quantity $L_k(\eta^{\lambda})$, for a given vector η^{λ}, can be computed by the following LP formulations:

$$L_k(\eta^{\lambda}) = \max_{(u_{ik})_{i\in O}} \{\sum_{i\in O}\eta^{\lambda}_i u_{ik} : \sum_{i\in O} u_{ik} = k,\ 0\leq u_{ik}\leq 1\} \tag{7}$$

$$= \min_{\substack{t_k\\(d_{ik})_{i\in O}}} \{k t_k + \sum_{i\in O} d_{ik} : \eta^{\lambda}_i \leq t_k + d_{ik},\ d_{ik}\geq 0\} \tag{8}$$

where (7) follows from the definition of $L_k(\eta^{\lambda})$ as the sum of the k largest values η^{λ}_i, while (8) is the dual LP with dual variable t_k corresponding to equation

$\sum_{i \in O} u_{ik} = k$ and variables d_{ik} corresponding to upper bounds on u_{ik}. Therefore, we have:

$$
\begin{aligned}
&\min_{\eta^\lambda \in E} \sum_{k \in O} w'_k L_k(\eta^\lambda) \\
&\quad = \min_{\eta^\lambda \in E} \sum_{k \in O} w'_k \min_{\substack{t_k \\ (d_{ik})_{i \in O}}} \{k t_k + \sum_{i \in O} d_{ik} : \eta^\lambda_i \leq t_k + d_{ik},\ d_{ik} \geq 0\} && (9) \\
&\quad = \min_{\eta^\lambda \in E} \min_{\substack{(t_k)_{k \in O} \\ (d_{ik})_{i,k \in O}}} \{\sum_{k \in O} w'_k \big(k t_k + \sum_{i \in O} d_{ik}\big) : \eta^\lambda_i \leq t_k + d_{ik},\ d_{ik} \geq 0\} && (10)
\end{aligned}
$$

where (9) derives from (8) and (10) derives from (9) as $w'_k > 0$. Together with the LP constraints $(\mathcal{C}')$ of set E. This leads to the following linearization of $(\mathcal{D}')$:

$$
\begin{aligned}
\min\ & \sum_{k \in O} w'_k (k t_k + \sum_{i \in O} d_{ik}) \\
\text{s.t. } & \lambda_i \big(I_i - \sum_{s \in S} \sum_{a \in A} R_i(s,a)\, x_{sa}\big) \leq t_k + d_{ik} \quad \forall i, k \in O \\
& \sum_{a \in A} x_{sa} - \gamma \sum_{s' \in S} \sum_{a \in A} T(s',a,s)\, x_{s'a} = \mu(s) \quad \forall s \in S \\
& x_{sa} \geq 0 \quad \forall s \in S, \forall a \in A; \quad d_{ik} \geq 0 \quad \forall\ i, k \in O
\end{aligned}
$$

Therefore, we get an exact LP formulation of the entire OWR problem $(\mathcal{D}')$. The randomized policy characterized by the x_{sa}'s at optimum is the OWR optimal policy. Our previous observation concerning the state-dependency of the OWR optimality tells us that the OWR-optimal solution might change with μ, which differs from the classical case. When the initial state is not known, distribution μ can be chosen as the uniform distribution over the possible initial states. When the initial state s_0 is known, $\mu(s)$ should be set to 1 when $s = s_0$ and to 0 otherwise. The solution found by the linear program does not specify which action to choose for the states that receive a null weight and that are not reachable from the initial state as they do not impact the value of the OWR-optimal policy.

5 Experimental Results

We tested our solving method on the navigation problem over a grid $N \times N$ ($N = 20, 50$ or 100 in our experiments). In this problem, a robot has four possible actions: Left, Up, Right, Down. The transition function models the fact that when moving, the robot may deviate from its trajectory with some fixed probability because it does not have a perfect control of its motor.

We ran four series of experiments with 100 instances each time. Unless otherwise stated, the parameters are chosen as follows. Rewards are two-dimensional vectors whose components are randomly drawn within interval $[0, 1]$. The discount factor is set to 0.9 and the initial state is set arbitrarily to the upper left corner of the grid. We set $w = (2/3, 1/3)$ (normalized vector obtained from $(1, 1/2)$) and $\lambda = (1, 1)$.

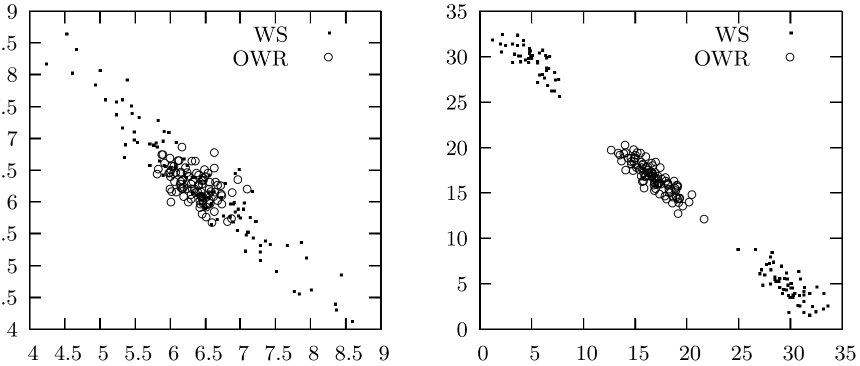

Fig. 5. 1st series (left), 2nd series (right) of experiments

As criteria are generally conflicting in real problems, for the first set of experiments, to generate realistic random instances, we simulate conflicting criteria with the following procedure: we pick one criterion randomly for each state and action and its value is drawn uniformly in $[0, 0.5]$ and the value of the other is drawn in $[0.5, 1]$. The results are represented on Figure 5 (left). One point (a dot for WS and a circle for OWR) represents the optimal value function in the initial state for one instance. Naturally, for some instances, WS provides a balanced solution but in most cases, WS gives a bad compromise solution. Figure 5 (left) shows that we do not have any control on tradeoffs obtained with WS. On the contrary, when using OWR, the solutions are always balanced.

To confirm the effectiveness of our approach, we ran a second set of experiments on pathological instances of the navigation problem. All the rewards are drawn randomly as for the first set of experiments. Then, in the initial state, for each action that does not move to a wall, we choose randomly one of the criteria and add a constant (here, arbitrarily set to 5). Then by construction, the value functions of all non-dominated deterministic policies in the initial state are unbalanced. The results are shown on Figure 5 (right). Reassuringly, we can see that OWR continues to produce fair solutions on the contrary to WS.

Our approach is still effective in higher dimensions. We ran a third set of experiments with three objectives as in higher dimensions, the experimental results would be difficult to visualize and as in dimension three, one can already show that OWR can be more effective than Minmax Regret or Augmented Tchebycheff. This last point could not have been shown in dimension two. In this third set of experiments, we set $w = (9/13, 3/13, 1/13)$ (normalized vector obtained from $(1, 1/3, 1/9)$) and $\lambda = (1, 1, 1)$. The random rewards are generated in order to obtain pathological instances in the spirit of the previous series of experiments. We set the initial state in the middle of the grid as we need to change the rewards of three actions. First, all rewards are initialized as in the first series of experiments (one objective drawn in $[0.5, 1]$, the other two in $[0, 0.5]$). In the initial state, for a first action, we add a constant C (here, $C = 5$) to the first component of its reward and a smaller constant c (here, $c = \frac{4}{5}C$) to its second

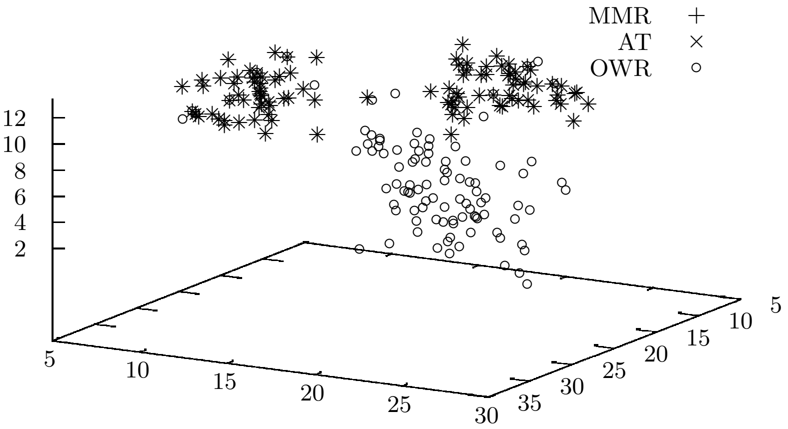

Fig. 6. Experiments with 3 objectives

one. For a second action, we do the opposite. We add c to its first component and C to its second one. For a third action, we add 5 to its third component and we subtract $2C$ from one of its first two ones chosen randomly. In such an instance, a policy choosing the third action in the initial state would yield a very low regret for the third objective, but the regrets for the first two objectives would not be balanced. In order to obtain a policy which yields a balanced profile on regrets, one needs to consider the first two actions.

The results of this set of experiments are shown on Figure 6. MMR stands for Minmax Regret and AT for Augmented Tchebycheff. Each point corresponds to the value of the optimal (w.r.t. MMR, AT or OWR) value function in the initial state of a random instance. One can notice that MMR and AT give the same solutions as both criteria are very similar. In our instances, it is very rare that one needs the augmented part of AT. Furthermore, one can see that the OWR-optimal solutions are between those optimal for MMR and AT. Although the OWR-optimal solutions are weaker on the third dimension, they fairly take into account potentialities on each objective and are better on at least one of the first two objectives.

For the last series of experiments, we tested our solution method with different scaling factors on the same instances as in the second series. With $\lambda = (1.75, 1)$ (resp. $\lambda = (1, 1.75)$), one can observe on the left (resp. right) hand side of Figure 7 that the obtained optimal tradeoffs with OWR now slightly favor the first (resp. second) objective as it could be expected.

We also perform experiments with more than three objectives. In Table 2, we give the average execution time in function of the problem size. The experiments were run using CPLEX 12.1 on a PC (Intel Core 2 CPU 2.66Ghz) with 4GB of RAM. The first row (n) gives the number of objectives. Row Size gives the number of states of the problem. Row TW gives the execution time for WS approach while row TO gives the execution time for OWR. All the times are given in

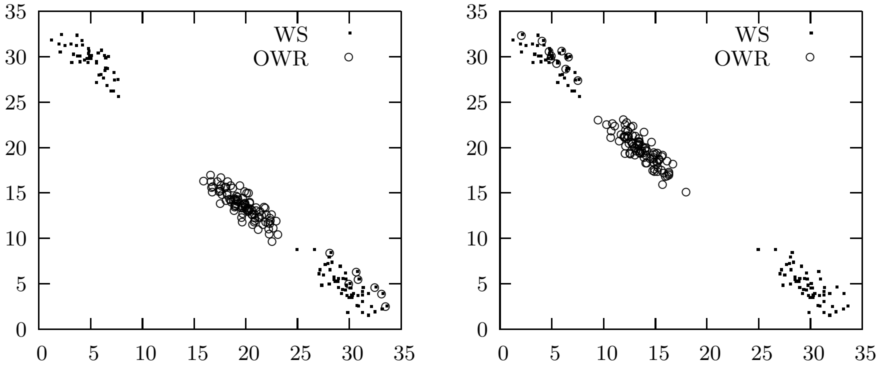

Fig. 7. 4th series of experiments (left: $\lambda = (1.75, 1)$, right: $\lambda = (1, 1.75)$)

Table 2. Average execution time in seconds

n	2			4			8			16		
Size	400	2500	10000	400	2500	10000	400	2500	10000	400	2500	10000
TW	0.2	5.2	147.6	0.10	5.1	143.7	0.1	4.7	146.0	0.12	4.9	143.6
TO	0.4	13.6	416.2	0.65	27.6	839.4	1.4	55.4	1701.7	3.10	111.5	3250.4

seconds as averages over 20 experiments. The OWR computation times increase proportionally to the number of criteria. Nevertheless, due to the huge number of variables x_{sa}'s, one may need to apply some column generation techniques [4] for larger problems.

6 Conclusion

We have proposed a method to generate fair solutions in MMDPs with OWR. Although this scalarizing function is not linear and cannot be optimized using value and policy iterations, we have provided an LP-solvable formulation of the problem. In all the experiments performed, OWR significantly outperforms the weighted sum concerning the ability to provide policies having a well-balanced valuation vector, especially on difficult instances designed to exhibit conflicting objectives. Moreover, introducing scaling factors λ_i in OWR yields deliberately biased tradeoffs within the set of Pareto-optimal solutions, thus providing full control to the decision maker in the exploration of policies.

Acknowledgements. The research by W. Ogryczak was partially supported by European Social Fund within the project Warsaw University of Technology Development Programme. The research by P. Perny and P. Weng was supported by the project ANR-09-BLAN-0361 GUaranteed Efficiency for PAReto optimal solutions Determination (GUEPARD).

References

1. Altman, E.: Constrained Markov Decision Processes. CRC Press, Boca Raton (1999)
2. Boutilier, C.: Sequential optimality and coordination in multiagent systems. In: Proc. IJCAI (1999)
3. Chatterjee, K., Majumdar, R., Henzinger, T.: Markov decision processes with multiple objectives. In: Durand, B., Thomas, W. (eds.) STACS 2006. LNCS, vol. 3884, pp. 325–336. Springer, Heidelberg (2006)
4. Desrosiers, J., Luebbecke, M.: A primer in column generation. In: Desaulniers, G., Desrosier, J., Solomon, M. (eds.) column generation, pp. 1–32. Springer, Heidelberg (2005)
5. Furukawa, N.: Vector-valued Markovian decision processes with countable state space. In: Recent Developments in MDPs, vol. 36, pp. 205–223 (1980)
6. Geoffrion, A.: Proper efficiency and the theory of vector maximization. J. Math. Anal. Appls. 22, 618–630 (1968)
7. Guestrin, C., Koller, D., Parr, R.: Multiagent planning with factored MDPs. In: NIPS (2001)
8. Hansen, P.: Bicriterion Path Problems. In: Multiple Criteria Decision Making Theory and Application, pp. 109–127. Springer, Heidelberg (1979)
9. Kostreva, M., Ogryczak, W., Wierzbicki, A.: Equitable aggregations and multiple criteria analysis. Eur. J. Operational Research 158, 362–367 (2004)
10. Littman, M.L., Dean, T.L., Kaelbling, L.P.: On the complexity of solving Markov decision problems. In: UAI, pp. 394–402 (1995)
11. Llamazares, B.: Simple and absolute special majorities generated by OWA operators. Eur. J. Operational Research 158, 707–720 (2004)
12. Marshall, A., Olkin, I.: Inequalities: Theory of Majorization and its Applications. Academic Press, London (1979)
13. Mouaddib, A.: Multi-objective decision-theoretic path planning. IEEE Int. Conf. Robotics and Automation 3, 2814–2819 (2004)
14. Ogryczak, W., Sliwinski, T.: On solving linear programs with the ordered weighted averaging objective. Eur. J. Operational Research 148, 80–91 (2003)
15. Puterman, M.: Markov decision processes: discrete stochastic dynamic programming. Wiley, Chichester (1994)
16. Steuer, R.: Multiple criteria optimization. John Wiley, Chichester (1986)
17. Viswanathan, B., Aggarwal, V., Nair, K.: Multiple criteria Markov decision processes. TIMS Studies in the Management Sciences 6, 263–272 (1977)
18. White, D.: Multi-objective infinite-horizon discounted Markov decision processes. J. Math. Anal. Appls. 89, 639–647 (1982)
19. Yager, R.: On ordered weighted averaging aggregation operators in multi-criteria decision making. IEEE Trans. on Syst., Man and Cyb. 18, 183–190 (1988)
20. Yager, R.: Decision making using minimization of regret. Int. J. of Approximate Reasoning 36, 109–128 (2004)