International Journal of Information Technology & Decision Making
Vol. 12, No. 5 (2013) 1021–1053

DOI: 10.1142/S0219622013400075

A COMPROMISE PROGRAMMING APPROACH TO MULTIOBJECTIVE MARKOV DECISION PROCESSES

WLODZIMIERZ OGRYCZAK
ICCE, Warsaw University of Technology, Poland
w.ogryczak@ia.pw.edu.pl

PATRICE PERNY
LIP6, University Pierre and Marie Curie
Paris, France

PAUL WENG
LIP6, University Pierre and Marie Curie
Paris, France

A Markov decision process (MDP) is a general model for solving planning problems under uncertainty. It has been extended to multiobjective MDP to address multicriteria or multiagent problems in which the value of a decision must be evaluated according to several viewpoints, sometimes conflicting. Although most of the studies concentrate on the determination of the set of Pareto-optimal policies, we focus here on a more specialized problem that concerns the direct determination of policies achieving well-balanced tradeoffs. To this end, we introduce a reference point method based on the optimization of a weighted ordered weighted average (WOWA) of individual disachievements. We show that the resulting notion of optimal policy does not satisfy the Bellman principle and depends on the initial state. To overcome these difficulties, we propose a solution method based on a linear programming (LP) reformulation of the problem. Finally, we illustrate the feasibility of the proposed method on two types of planning problems under uncertainty arising in navigation of an autonomous agent and in inventory management.

Keywords: Multiobjective optimization; Markov decision processes; compromise programming; reference point method; ordered weighted average; linear programming.

1. Introduction

The framework of Markov decision processes (MDPs) provides a useful mathematical model for representing and solving sequential decision-making problems under uncertainty. This model introduced fifty years ago (by Bellman[1] and Howard[2]) is used in many applications of Operations Research (investment planning, inventory systems management, manufacturing, resource allocation)[3,4] and Artificial Intelligence (path-planning, game search, trading agents, robotics and reinforcement learning).[5,6] Several variants of MDPs have been already considered and investigated, depending on the nature of the set of states (finite or not), the nature of the

reward function (quantitative or qualitative, scalar-valued or vector-valued), the nature of uncertainty (probabilistic, possibilistic, ordinal) and the observability of the system (partial or total), etc. In this paper we deal with the classical case (finite set of states (fully observable), additive rewards, probabilistic uncertainty) except that we consider several reward functions, possibly conflicting, each providing a different viewpoint on the value of a policy, thus leading to a multiobjective version of the problem. Our aim is to propose an efficient method in such multiobjective MDPs to generate compromise solutions, i.e. whose profile is well balanced in a certain sense that we will define.

This study is motivated by the numerous practical planning problems in which several viewpoints must be considered for the selection of actions. Many problems indeed involve several criteria, possibly conflicting, and preference analysis over policies must incorporate all of them during the optimization process. For example, in path-planning problems under uncertainty, we may want to optimize various features like travel duration, distance, energy consumption and risk simultaneously, rather than focusing on a single aspect. Even when all criteria can be expressed in monetary terms, it is often better to keep them separate because they refer to various type of consequences that do not compensate each other and possibly concern different stakeholders. For instance, in the management of inventory systems, it is usual to consider costs attached to storage, but also opportunity losses and supply costs as well, each of them impacting differently on the system. Thus, a policy entailing small storage costs but numerous shortage periods will not be equivalent to a solution with higher storage costs but no shortage. Other examples can be found in Artificial Intelligence as well. For instance, multi-agents planning is another area where multiobjective optimization is natural. In such problems, several agents must cooperate but each of them has its own value system and its own objective. Hence, a reward function must be defined for every agent, and the goal is to find a collective policy that fairly shares satisfaction among individuals. We can therefore distinguish two main possible contributions of multiobjective optimization to planning under uncertainty: one due to the need of handling several criteria in some planning problems, the other linked to multi-agents planning activities. This explains the current interest for multiobjective (multicriteria or multiagent) extensions of MDPs in the literature.[7–11]

When several objectives must be optimized simultaneously, most of the studies on MDPs concentrate on the determination of the entire set of Pareto-optimal policies, i.e. policies having a reward vector that cannot be improved on a criterion without being downgraded on another criterion. However, the size of the Pareto-optimal deterministic policies 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. In practice however, there is generally no need to determine the entire set of Pareto-optimal policies, but only specific compromise policies achieving a good

tradeoff between the possibly conflicting objectives. A similar statement could be made when one considers all randomized policies.

The search for a compromise solution is natural in multicriteria decision support[14,15] and has counterparts in other optimization contexts involving several dimensions. In the context of multi-agents optimization problems, the notion of compromise refers to the idea of fairness.[12,13] In all these cases, the quality of the compromise achieved can be measured using a scalarizing function discriminating between Pareto-optimal solutions. This function must be optimized to generate an optimal compromise solution, as is usually done in interactive methods for multiobjective optimization.[14,16]

Motivated by such examples, we introduce in this paper a compromise programming approach for the determination of well-balanced policies in multiobjective MDPs. It is based on the optimization of a weighted ordered weighted average (WOWA) used to scalarized expected reward vectors attached to policies. We discuss the technical problems to overcome when using such a function in a multi-stage decision problem and provide solution methods based on linear programming. The paper is organized as follows:

In Sec. 2, we recall the basic notions related to MDPs and their multiobjective extension. In Sec. 3, we introduce WOWA as a scalarizing function to generate compromise solutions. Section 4 is devoted to the search of WOWA-optimal policies in multiobjective MDPs. Finally, Sec. 5 presents some experimental results showing the effectiveness of our approach in generating compromise policies.

2. Multiobjective Markov Decision Processes

In this section, we first recall the definition of the standard MDP and some basic related notions. We then present its extension to the multiobjective case.

2.1. *Markov decision processes*

The model of MDPs has become an important framework for representing and solving sequential decision problems under uncertainty. In such problems, a decision-maker (DM) repeatedly faces a choice problem (a finite or an infinite number of times). In a choice problem, the DM has to pick an action among a set of actions. Each action has uncertain consequences and affects the future choice problems the DM faces. A solution of this sequential problem would be to determine in advance a sequence of choices optimal with respect to a preference structure, e.g. optimizing some costs or rewards.

A MDP[17] is described as a tuple (S, A, T, R) where:

- S is a finite set of states,
- A is a finite set of actions,

- T: $S \times A \to \mathbf{Pr}(S)$ is a transition function, stating for each state and each action, a probability distribution over states (in the sequel, we write $T(s, a, s')$ for $T(s, a)(s')$),
- r: $S \times A \to \mathbb{R}$ is a reward function giving the immediate reward for executing a given action in a given state.

Given a finite or infinite horizon (i.e. the number of decisions to be made), this tuple models a sequential decision problem. In such a model, one wants to find the preferred sequence of decisions with respect to some preference representation. We assume in this paper that the horizon is infinite.

In this framework, a *decision rule* δ is a procedure that determines which action to choose in each state. The procedure can be *deterministic* and δ is then defined as a function $\delta : S \to A$. More generally, the procedure can be *randomized* and δ is then defined as a function $\delta : S \to \mathbf{Pr}(A)$ 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 all the decision rules are deterministic and *randomized* otherwise. If the same decision rule δ is applied at each step, the policy is said to be *stationary* and is denoted δ^∞.

In standard MDPs, a policy π is valued by a function $v^\pi : S \to \mathbb{R}$, called *value function*, which gives the expected discounted total reward obtained by applying π in each initial state. For $\pi = (\delta_0, \delta_1, \ldots, \delta_t, \ldots)$, they are given by: $\forall s \in S$, $\forall h > 0$, $\forall t = 1, \ldots, h$,

$$\begin{aligned} v_0^\pi(s) &= 0 \\ v_t^\pi(s) &= r(s, \delta_{h-t}(s)) + \gamma \sum_{s' \in S} T(s, \delta_{h-t}(s), s') v_{t-1}^\pi(s'), \end{aligned} \tag{1}$$

where $\gamma \in [0, 1[$ is the discount factor. Value function v_h^π defines the value of policy π at horizon h. This sequence converges to the value function of π at the infinite horizon. A discount factor γ strictly lower than 1 guarantees v^π is well defined at the infinite horizon.

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

$$\forall s \in S, \quad v^*(s) = \max_{a \in A} r(s, a) + \gamma \sum_{s' \in S} T(s, a, s') v^*(s'). \tag{2}$$

There are three main approaches for solving MDPs. Two are based on dynamic programming: value iteration and policy iteration. The last approach is based on linear programming. We now recall value iteration and the linear programming approach as they are needed for the exposition of our results.

Value iteration consists of computing the solution of the Bellman equations using the following recursive sequence: $\forall s \in S, \forall t > 0$,

$$\begin{aligned} v_0^*(s) &= 0 \\ v_t^*(s) &= \max_{a \in A} r(s,a) + \gamma \sum_{s' \in S} T(s,a,s') v_{t-1}^*(s'). \end{aligned}$$

This sequence converges to the optimal value function and the optimal stationary policy can be recovered by a greedy optimization.

VALUE ITERATION

1: $\forall s \in S, v(s) \leftarrow 0$
2: **repeat**
3: **for all** $s \in S$ **do**
4: **for all** $a \in A$ **do**
5: $q(s,a) \leftarrow r(s,a) + \gamma \sum_{s' \in S} T(s,a,s') v(s')$
6: **end for**
7: $v(s) \leftarrow \max_{a \in A} q(s,a)$
8: **end for**
9: **until** convergence of v

An MDP can also be solved by linear programming. The Bellman equations state that functions satisfying the following inequalities are upper bounds of the optimal value function:

$$\forall s \in S,\ \forall a \in A, \quad v(s) \geq r(s,a) + \gamma \sum_{s' \in S} T(s,a,s') v(s').$$

The linear program can then be written as follows:

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

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. However, we will assume here that μ is normalized.

The dual of this program writes as follows:

$$(\mathcal{D}) \quad \left\{ \begin{array}{ll} \max \displaystyle\sum_{s \in S} \sum_{a \in A} r(s,a) x(s,a) & \\ \text{s.t.} \displaystyle\sum_{a \in A} x(s,a) - \gamma \sum_{s' \in S} \sum_{a \in A} x(s',a)\, T(s',a,s) = \mu(s) & \forall s \in S \\ x(s,a) \geq 0 & \forall s \in S,\ \forall a \in A \end{array} \right\}. \quad (\mathcal{C})$$

Program ($\mathcal{D}$) has a nice property, it separates the dynamics of the system (in the constraints) and the preference representation (in the objective function).

To give an interpretation to variables $x(s, a)$, we recall the two following propositions that relate feasible solutions of the dual linear program to stationary randomized policies in the MDP.[17]

Proposition 1. *For a policy π, if x^π is defined as*:

$$x^\pi(s, a) = \sum_{t=0}^{\infty} \gamma^t p_t^\pi(s, a) \quad \forall s \in S,\ \forall a \in A, \tag{3}$$

where $p_t^\pi(s, a)$ is the probability of reaching state s and choosing action a at step t, then x^π is a feasible solution of the dual linear program.

Note that $\forall s \in S,\ \mu(s) = \sum_{a \in A} p_0^\pi(s, a)$.

Proposition 2. *If $x(s, a)$ is a solution of the dual problem, then the stationary randomized policy δ^∞, defined by*:

$$\delta(s, a) = \frac{x(s, a)}{\sum_{a \in A} x(s, a)} \quad \forall s \in S,\ \forall a \in A \tag{4}$$

defines values $x^{\delta^\infty}(s, a)$ as in Eq. (3), that are equal to $x(s, a)$.

Thus, there is a one-to-one mapping between stationary randomized policies and the solutions x satisfying constraints ($\mathcal{C}$). Moreover, the basic solutions of the dual program $\mathcal{D}$ correspond to stationary deterministic policies. The stationary randomized policies are in the convex hull of the basic solutions. Note that in an MDP, any feasible value function can be obtained with a stationary randomized policy.[17] Besides, for any x and δ defined as in Proposition 2, the expectation of the value function of δ^∞ with respect to μ can be computed as follows:

$$\sum_{s \in S} \sum_{a \in A} r(s, a) x(s, a) = \sum_{s \in S} \mu(s) v^{\delta^\infty}(s). \tag{5}$$

2.2. *Multiobjective MDP*

MDP has been extended to take into account multiple objectives or criteria. A multiobjective MDP (MMDP) is defined as an MDP with the reward function replaced by:

- R: $S \times A \to \mathbb{R}^n$ where n is the number of criteria, $R(s, a) = (R_1(s, a), \ldots, R_n(s, a))$ and $R_i(s, a)$ is the immediate reward for criterion i.

Now, a policy π is valued by a value function $V^\pi : S \to \mathbb{R}^n$, which gives the expected discounted total reward vector in each state and can be computed with a vectorial version of (1) where additions and multiplications are componentwise.

To compare the value of policies in a given state s, the basic model adopted in most previous studies[18–20] is *Pareto dominance* defined as follows: For any two policies π, π', for any state s, $V^{\pi}(s)$ Pareto-dominates $V^{\pi'}(s)$, denoted by $V^{\pi}(s) \succ_P V^{\pi'}(s)$ if and only if:

$$V^{\pi}(s) \neq V^{\pi'}(s) \quad \text{and} \quad \forall i = 1, \ldots, n,\ V_i^{\pi}(s) \geq V_i^{\pi'}(s). \tag{6}$$

As Pareto dominance is a partial relation, generally there exist many optimal vectors in a given state. For a set $X \subset \mathbb{R}^n$, the set of *Pareto-optimal* vectors of X is defined by $M(X, \succ_P) = \{x \in X : \forall y \in X,\ \text{not } y \succ_P x\}$.

Standard methods for MDPs can be extended to solve MMDPs. For the linear programming approach, this can be shown as follows. Let $(\mathcal{D}_i)$ be the dual linear program $(\mathcal{D})$ solving the MDP with reward function R_i for $i = 1, \ldots, n$. Clearly, all $(\mathcal{D}_i)$ share the same constraints $(\mathcal{C})$. Hence, as argued in Ref. 19, the multiobjective version of an MDP with reward system $R = (R_1, \ldots, R_n)$ can be solved by the following multiobjective linear program:

$$(v\mathcal{D}) \quad \left\{ \begin{array}{ll} \text{v-max} \displaystyle\sum_{s \in S} \sum_{a \in A} R(s,a) x(s,a) & \\ \text{s.t.} \displaystyle\sum_{a \in A} x(s,a) - \gamma \sum_{s' \in S} \sum_{a \in A} x(s',a)\, T(s',a,s) = \mu(s) & \forall s \in S \\ x(s,a) \geq 0 & \forall s \in S, \forall a \in A \end{array} \right\}, \quad (\mathcal{C})$$

where v-max is a vector maximization with respect to Pareto dominance. Recall that for any x satisfying $(\mathcal{C})$ and δ defined as in Proposition 2, we have $\sum_{s \in S} \sum_{a \in A} R_i(s,a) x(s,a) = \sum_{s \in S} \mu(s)\, V_i^{\delta^\infty}(s)$ for all $i = 1, \ldots, n$. Thus, solving $(v\mathcal{D})$ amounts to optimizing the following objective function $\sum_{s \in S} \mu(s)\, V(s)$, interpreted as the expectation of a vector value function V with respect to probability distribution μ. Therefore, setting $\mu(s_0) = 1$ for some s_0 and $\mu(s) = 0$ for all $s \neq s_0$, the objective function boils down to $V^{\delta^\infty}(s_0) = \sum_{s \in S} \sum_{a \in A} R(s,a) x(s,a)$.

Looking for all non-dominated solutions can be difficult and time-consuming as there could be many non-dominated solutions. In fact, there exists instances of problems where the number of solutions is exponential in the number of states. We illustrate this point by adapting an example proposed by Hansen.[21]

Example 1. Let $N > 0$. Consider the following deterministic MMDP represented in Fig. 1. It has $N + 1$ states. In each state, two actions (up or down) are possible except in the absorbing state N. The rewards are given next to the arcs representing the two actions. Here, we can take $\gamma = 1$ as state N is absorbing.

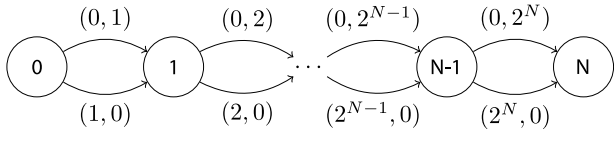

Fig. 1. Hansen graph.

In this example, there are 2^{N+1} stationary deterministic policies. Stationary deterministic policies that only differ from one another on the choice of the action in the last state N have the same value functions as the reward and the transition in those states for both actions are identical. In the initial state 0, the remaining policies induce 2^N different valuation vectors, of the form $(x, 2^N - 1 - x)$ for $x = 0, 1, \ldots, 2^N - 1$. Those different vectors are in fact all Pareto-optimal as they are on the line $x + y = 2^N - 1$. It is then infeasible in such a case to determine all non-dominated solutions.

Besides, in practice, the DM is only interested in finding one particular solution among all the non-dominated solutions that fits her preferences concerning the tradeoffs between all the criteria. Hence, it seems more natural to directly model the problem as a search for that particular solution instead of finding first all the non-dominated solutions.

We introduce the notion of *scalarizing function* that will be used to discriminate between Pareto-optimal vectors. Formally, a scalarizing function is a function ψ: $\mathbb{R}^n \to \mathbb{R}$ that defines an *overall value* $v(s_0) \in \mathbb{R}$ for an initial state s_0 from a vector value function $V : S \to \mathbb{R}^n$:

$$v(s_0) = \psi(V_1(s_0), \ldots, V_n(s_0)). \tag{7}$$

The most straightforward choice for ψ seems to be the weighted sum, which is in fact not suited for generating compromise solutions (see Example 2). In this case, $v(s_0) = \sum_{i=1}^n \lambda_i V_i(s_0)$ where $\lambda_i > 0, \forall i = 1, \ldots, n$ so as to preserve the monotonicity with respect to Pareto dominance. By linearity of the mathematical expectation and the weighted sum, optimizing $v(s_0)$ is equivalent to solving the standard MDP obtained from the MMDP where the reward function is defined as: $r(s, a) = \sum_{i=1}^n \lambda_i R_i(s, a), \forall s \in S, \forall a \in A$. In that case, an optimal stationary deterministic policy exists and standard solution methods can then be applied. However, using a weighted sum is not a good procedure for reaching balanced solutions as weighted sum is fully compensatory operator that does not encode the idea of balanced solutions. This is well illustrated by the two following examples:

Example 2. Consider the deterministic MMDP depicted in Fig. 2 with two criteria using the same valuation scale. In this MDP, there only exists two deterministic stationary policies depending on the choice of the action in state 1. They are thus denoted a and b, respectively. Their value functions are given by: $V^a(1) = (1/(1-\gamma), 9/(1-\gamma))$ and $V^b(1) = (5/(1-\gamma), 5/(1-\gamma))$. If the rewards of the agents are simply summed (i.e. we give equal weights to each criterion), then

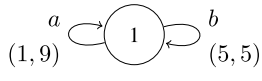

Fig. 2. MMDP of Example 2.

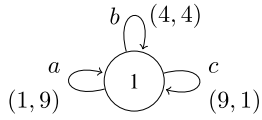

Fig. 3. MMDP of Example 3.

both policies are optimal and have the same value functions. However, the policy choosing action b yields a much better balanced vector and should be preferred.

From the previous example, one could think that by choosing appropriate weights, one could reach a balanced solution. This is not the case. There exists instances where well-balanced solutions cannot be obtained by optimizing a weighted sum.

Example 3. Consider the MMDP represented in Fig. 3.

Here, there are three stationary deterministic policies. They are denoted a, b and c respectively. Their value functions are given by: $V^a(1) = (1/(1-\gamma), 9/(1-\gamma))$, $V^b(1) = (4/(1-\gamma), 4/(1-\gamma))$ and $V^c(1) = (9/(1-\gamma), 1/(1-\gamma))$. By giving equal weights to both criteria, the policy that chooses action a and the one choosing action c are equivalent. As soon as the weights are different, only one of those two policies is optimal and it therefore yields an unbalanced valuation vector. However, the policy choosing action b is not dominated by the other policies. As it yields a much better balanced valuation vector, one could arguably prefer that solution. Yet, it cannot be obtained by a weighted sum.

In the next section, we introduce a a scalarizing function well suited for the determination of well-balanced solutions.

3. Search for Compromise Solutions

3.1. *Reference point method*

In multiobjective optimization, the question of finding balanced solutions among the non-dominated ones is a crucial issue. The standard way of generating compromise solutions in the Pareto-optimal set is resorting to the so-called reference point approach that consists in finding an attainable outcome vector that in some manner minimizes a distance to a prescribed reference point.[14,22,23] This can be achieved with the so-called quasi-satisficing approach. A good formalization of the quasi-satisficing approach to multiobjective optimization was proposed and developed mainly by Wierzbicki[24] as the reference point method (RPM).

Within the RPM the DM specifies requirements in terms of reference levels, i.e. by introducing reference (target) values for several individual outcomes. Depending on the specified reference levels, a particular scalarizing disachievement

function is built which may be directly interpreted as expressing disutility to be minimized. Minimization of the scalarizing disachievement function generates a Pareto-optimal solution to the multiobjective problem. The scalarizing disachievement function can be viewed as a two-stage transformation of the original outcomes. First, the strictly monotonic individual (partial) disachievement functions are built to measure individual performance with respect to given reference levels. Having all the outcomes transformed into a uniform scale of individual disachievements they are aggregated at the second stage to form a unique scalarization. The RPM is based on the so-called augmented (or regularized) min−max aggregation. Thus, the worst individual disachievement is essentially optimized but the optimization process is additionally regularized with the term representing the average disachievement. The min−max aggregation guarantees fair treatment of all individual disachievements by implementing an approximation to the Rawlsian principle of justice. While building the scalarizing disachievement function, it is assumed that DM prefers a solution with all individual outcomes y_i satisfying the corresponding reference levels to any solution with at least one individual outcome worse than its reference level. That means, the minimization of the scalarizing disachievement function must enforce reaching the reference levels prior to further improving of criteria. Thus, similar to the goal programming approaches, the reference levels are treated as targets but following the quasi-satisficing approach they are interpreted consistently with basic concepts of Pareto optimality in the sense that even when the target point can be reached, the better solution compared to it, the more preferred the solution is.[25]

The generic scalarizing disachievement function takes the following form[24]:

$$\psi(\mathbf{y}) = (1-\varepsilon)\max_{1\le i\le n}\{\sigma_i(y_i)\} + \frac{\varepsilon}{n}\sum_{i=1}^{n}\sigma_i(y_i), \tag{8}$$

where ε is an arbitrary small positive number and $\sigma_i : \mathbb{R} \to \mathbb{R}$, for $i = 1, 2, \ldots, n$, are the individual (partial) disachievement functions measuring actual disachievement of the individual outcomes y_i with respect to the corresponding reference levels. Let η_i denote the individual disachievement for the ith outcome ($\eta_i = \sigma_i(y_i)$) and $\eta = (\eta_1, \eta_2, \ldots, \eta_n)$ represent the disachievement vector. The scalarizing disachievement function (8) is essentially defined by the worst individual disachievement but additionally regularized with the sum of all individual disachievements. The regularization term is introduced only to guarantee the solution efficiency in the case when the minimization of the main term (the worst individual disachievement) results in a non-unique optimal solution.

Various functions σ_i provide a wide modeling environment for measuring individual disachievements.[22,26–28] The basic RPM model is based on a single vector of reference levels, the aspiration vector $\mathbf{r}^a$. Real-life applications of the RPM methodology usually deal with more complex individual disachievement functions defined with more than one reference point[26] which enriches the preference models and

simplifies the interactive analysis. In particular, the models taking advantages of two reference vectors: vector of aspiration levels $\mathbf{r}^a$ and vector of reservation levels $\mathbf{r}^r$[29] are used, thus allowing the DM to specify requirements by introducing acceptable and required values for several outcomes. The individual disachievement function σ_i can be interpreted then as a measure of the DM's dissatisfaction with the current value of outcome for the ith criterion. It takes value $\eta_i = 0$ for $y_i = r_i^a$, and $\eta_i = 1$ for $y_i = r_i^r$. Various functions can be built meeting those requirements. We use the piecewise linear individual disachievement function originally introduced in an implementation of the RPM system for the multiple criteria transshipment problems with facility location[30]:

$$\sigma_i(y_i) = \begin{cases} \beta \dfrac{y_i - r_i^r}{r_i^r - r_i^a} + 1, & \text{if } y_i \text{ is worst than } r_i^r \\ \dfrac{y_i - r_i^a}{r_i^r - r_i^a}, & \text{if } y_i \text{ is between } r_i^a \text{ and } r_i^r \\ \alpha \dfrac{y_i - r_i^a}{r_i^r - r_i^a}, & \text{if } y_i \text{ is better than } r_i^a, \end{cases} \tag{9}$$

where α and β are arbitrarily defined parameters satisfying $0 < \alpha < 1 < \beta$. Parameter $\beta > 1$ represents increase of the DM's dissatisfaction connected with outcomes worse than the reservation level while parameter α represents additional decrease of the dissatisfaction (below zero) when a criterion generates outcomes better than the corresponding aspiration level. These disachievement functions are well defined for any type of criteria, either maximized or minimized (see Fig. 4). Indeed, assuming that $r_i^a > r_i^r$ for maximized criteria and respectively $r_i^a < r_i^r$ for criteria being minimized, the individual disachievement function (9) takes values $\sigma_i(y_i) = |y_i - r_i^a|/|r_i^a - r_i^r|$ if y_i is between r_i^a and r_i^r, $\sigma_i(y_i) = \beta|y_i - r_i^r|/|r_i^a - r_i^r| + 1$ if y_i is worst than r_i^r, and $\sigma_i(y_i) = -\alpha|y_i - r_i^a|/|r_i^a - r_i^r|$ when y_i is better

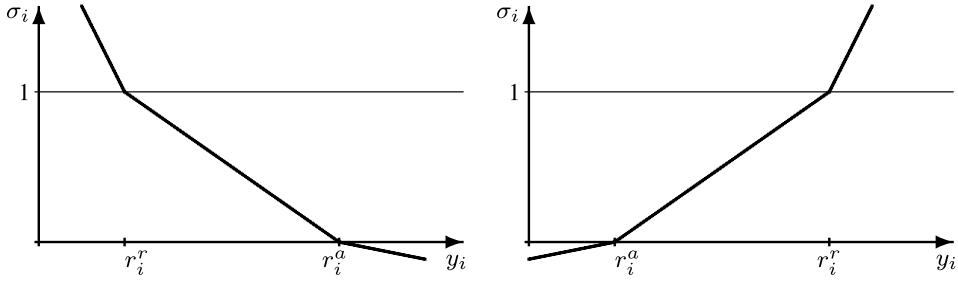

Fig. 4. Individual disachievement function (9): (a) maximized outcome ($r_i^a > r_i^r$), (b) minimized outcome ($r_i^a < r_i^r$).

than r_i^a. They can easily be extended to enable preference specification with more than two reference points.[31]

In a simplified approach, the reference points can be chosen as follows: the aspiration levels r^a as the *ideal point* (i.e. the lowest upper bound of the Pareto solutions when maximizing an objective and the greatest lower bound when minimizing an objective), and the reservation levels r^r as the *nadir point* (i.e. the highest lower bound of the Pareto solutions when maximizing and the lowest upper bound when minimizing). With such an approach the scalarizing function (8) with individual disachievements (9) takes the form of the augmented Tchebycheff scalarization:

$$\psi(y) = (1-\varepsilon)\max_{i=1\ldots n}\frac{|I_i - y_i|}{|I_i - N_i|} + \frac{\varepsilon}{n}\sum_{i=1}^{n}\frac{|I_i - y_i|}{|I_i - N_i|}, \tag{10}$$

where I is ideal point and N the nadir point. It defines the relative Tchebycheff distance from the ideal point. In practice, it is recommended for technical reasons to use,[22] instead of I, a point I', in the neighborhood of I that Pareto-dominates I. Moreover, one uses an approximation of the nadir point[32,33] instead of N, which is difficult to obtain when the number of criteria is greater than 2. Also, the use of various reference points allows one to model various compromise solutions.

3.2. *Reference point method with ordered aggregations*

The min−max aggregation is crucial for allowing the RPM to generate various compromise Pareto optimal solutions while the regularization is necessary to guarantee that only Pareto optimal solutions are obtained. The regularization by the average disachievement, applied in the standard RPM method (8), is very simple but it may disturb the basic min−max model. Actually, the only consequent regularization of the min−max aggregation is the lexmin order or the more practical ordered weighted averaging (OWA) aggregation with monotonic weights. In the OWA aggregation[34] of a vector the weights are assigned to the ordered values (i.e. to the largest value, the second largest and so on) rather than to the specific coordinates. The OWA aggregation with monotonic weights combines all the individual disachievements allocating the largest weight to the worst disachievement, the second largest weight to the second worst disachievement, the third largest weight to the third worst disachievement, and so on. This is mathematically formalized as follows. Let $\langle\eta\rangle = (\eta_{\langle 1\rangle}, \eta_{\langle 2\rangle}, \ldots, \eta_{\langle n\rangle})$ denote the vector obtained from η by rearranging its components in the non-increasing order. That means $\eta_{\langle 1\rangle} \geq \eta_{\langle 2\rangle} \geq \cdots \geq \eta_{\langle n\rangle}$ and there exists a permutation τ of set $\{1, \ldots, n\}$ such that $\eta_{\langle i\rangle} = \eta_{\tau(i)}$ for $i = 1, \ldots, n$. The standard min−max aggregation depends on the minimization of $\eta_{\langle 1\rangle}$ as it ignores the values of $\eta_{\langle i\rangle}$ for $i \geq 2$. In order to take into account all the disachievement values, one needs to minimize a weighted combination of the ordered disachievements thus representing the OWA aggregation. Note that the weights are then assigned to the

specific positions within the ordered disachievement vector rather than to the individual disachievements themselves.

With the OWA aggregation one gets the following scalarizing disachievement function to be minimized:

$$\mathrm{OWA}_{\omega}(\eta) = \sum_{i=1}^{n} \omega_i \eta_{\langle i \rangle} \quad \text{where } \eta_i = \sigma_i(y_i) \quad \forall i = 1, \ldots, n, \tag{11}$$

where $\omega_1 > \omega_2 > \cdots > \omega_n > 0$ are positive and strictly decreasing weights. Actually, they should be rather strongly decreasing to represent the regularization of the min−max order. When the differences among weights tend to infinity, the OWA aggregation approximates the leximax ranking of the ordered outcome vectors.[35] Certainly, any finite differences small enough to allow for numerical computation of the OWA values provides a proper scalarizing disachievement function. We recommend the use of geometric decreasing series to define the OWA RPM weights ω_i.

Example 4. Let us consider three alternative feasible solutions among which one wants to select one according to four criteria. Table 1 presents for all the solutions the corresponding individual disachievements defined according to the aspiration/reservation model (9) thus allocating 0 to outcomes reaching the corresponding aspiration level and 1 to those reaching the reservation level. Solution S1 oversteps the aspiration levels (disachievement values −0.2) for two criteria while failing to reach the corresponding aspiration levels for two other criteria (disachievement values 0.7). Solution S2 approaches the aspiration levels for the first three criteria (disachievement values 0.11) while clearly failing to reach only the aspiration level for the fourth criterion (disachievement value 0.7). Solution S3 essentially fails to reach all the aspiration levels, though being a little bit closer for the fourth criterion (disachievement value 0.6). All the solutions are Pareto optimal. They generate the same worst disachievement value 0.7 and therefore, while applying the standard RPM (8), the final selection depends on the average disachievement (regularization term). Actually, solution S1 will be selected as better than S2 and S3.

One may notice that the application of the OWA aggregation with decreasing weights $\omega = (0.5, 0.3, 0.15, 0.05)$ enables selection of solution S2 from Table 1. One may notice that $\mathrm{OWA}_{\omega}(\eta(S2)) < \mathrm{OWA}_{\omega}(\eta(S3))$ for any positive weights ω which means that solution S2 will be always treated as better than S3.

When it is possible the OWA RPM optimization (11) generates a Pareto-optimal solution with individual disachievements all equal, otherwise it generates another

Table 1. Sample disachievements for Example 4.

Sol.	η_1	η_2	η_3	η_4	Max	aver.	$\eta_{\langle 1 \rangle}$	$\eta_{\langle 2 \rangle}$	$\eta_{\langle 3 \rangle}$	$\eta_{\langle 4 \rangle}$	$\mathrm{OWA}_{\omega}(\eta)$
S1	0.7	−0.2	−0.2	0.7	0.7	0.25	0.7	0.7	−0.2	−0.2	0.520
S2	0.11	0.11	0.11	0.7	0.7	0.26	0.7	0.11	0.11	0.11	0.405
S3	0.4	0.3	0.7	0.6	0.7	0.50	0.7	0.6	0.4	0.3	0.605

Pareto-optimal solution but still providing equitability of individual disachievements with respect to the Pigou−Dalton principle of transfers.[13] That means, a transfer of a small amount from an individual disachievement to any relatively worse-off individual disachievement results in a more preferred disachievement vector, i.e. whenever $\eta_i < \eta_j$ and $0 < \varepsilon \leq \eta_j - \eta_i$, then $\eta + \varepsilon \mathbf{e}_i - \varepsilon \mathbf{e}_j$ is strictly preferred to η where $\mathbf{e}_i$ denotes the ith unit vector.

Note that the standard RPM model with the scalarizing disachievement function (8) can be expressed as the following OWA model: $\max((1 - \frac{(n-1)\varepsilon}{n})\eta_{\langle 1\rangle} + \frac{\varepsilon}{n}\sum_{i=2}^{n}\eta_{\langle i\rangle})$. Hence, the standard RPM model exactly represents the OWA aggregation (11) with strictly decreasing weights in the case of $n = 2$ ($\omega_1 = 1 - \varepsilon/2$ and $\omega_2 = \varepsilon/2$). For $n > 2$ it abandons the differences in weighting of the second largest disachievement, the third largest one, etc. ($\omega_2 = \cdots = \omega_n = \varepsilon/n$). The OWA RPM model (11) enables to distinguish weights[36] by introducing decreasing series (e.g. geometric ones).

When applying the OWA RPM scalarizing function (11) with individual disachievements (9) built for the ideal point I and the nadir point N as the reference levels:

$$\psi(y) = \mathrm{OWA}_{\omega}(\eta) = \sum_{i=1}^{n} \omega_i \eta_{\langle i\rangle} \quad \text{where } \eta_i = \frac{|I_i - y_i|}{|I_i - N_i|} \quad \forall i = 1, \ldots, n, \tag{12}$$

we receive a special case of the ordered weighted regret optimization.[11]

Typical RPM model allows weighting of several disachievements only by straightforward rescaling of the disachievement values. The OWA RPM model enables one to introduce importance weights to affect disachievement importance by rescaling accordingly its measure within the distribution of disachievements as defined in the so-called WOWA aggregation.[37] Let $\omega = (\omega_1, \ldots, \omega_n)$ be a set of the OWA weights and let $\lambda = (\lambda_1, \ldots, \lambda_n)$ be an additional importance weighting vector such that $\lambda_i \geq 0$ for $i = 1, \ldots, n$ and $\sum_{i=1}^{n} \lambda_i = 1$. The corresponding WOWA aggregation of disachievements $\eta = (\eta_1, \ldots, \eta_m)$ is defined as follows[37]:

$$\begin{aligned} \mathrm{WOWA}_{\omega,\lambda}(\eta) &= \sum_{i=1}^{n} w_i(\lambda, \eta)\eta_{\langle i\rangle} \quad \text{where} \\ w_i(\lambda, \eta) &= \varphi\left(\sum_{k \leq i} \lambda_{\tau(k)}\right) - \varphi\left(\sum_{k<i} \lambda_{\tau(k)}\right) \end{aligned} \tag{13}$$

with φ a monotone increasing function that interpolates points $(\frac{i}{m}, \sum_{k\leq i}\omega_k)$ together with the point (0.0) and τ representing the ordering permutation for η (i.e. $\eta_{\tau(i)} = \eta_{\langle i\rangle}$). Moreover, function φ is required to be a straight line when the point can be interpolated in this way, thus allowing the WOWA to cover the standard weighted mean with weights λ_i as a special case of equal OWA weights ($\omega_i = 1/n$ for $i = 1, \ldots, n$). Actually, within the theory of decisions under uncertainty where importance weights λ_i may represent scenarios probabilities, the WOWA aggregation

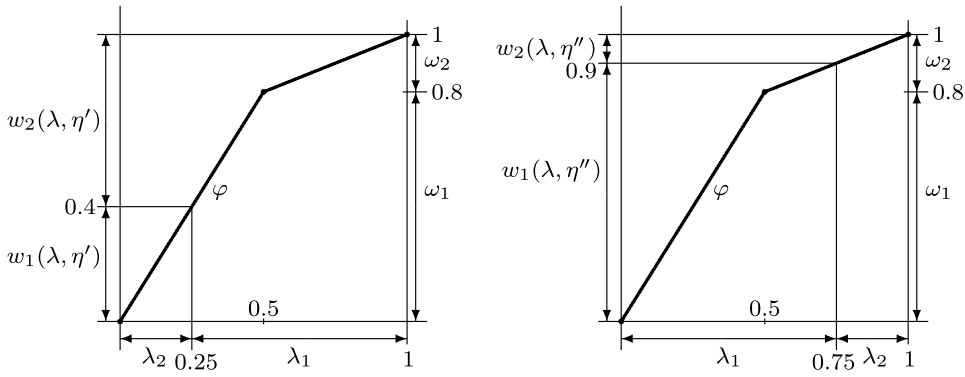

Fig. 5. Definition of WOWA weights w_i: (left) for vector $\eta' = (0.1, 0.2)$, (right) for vector $\eta'' = (0.2, 0.1)$.

is a special case of the rank dependent utility[39] with a piecewise linear probability weighting function φ defined by the importance weights.

Example 5. Consider two disachievements vectors $\eta' = (0.1, 0.2)$ and $\eta'' = (0.2, 0.1)$. While introducing preferential weights $\omega = (0.8, 0.2)$, one may calculate the OWA aggregations: $\mathrm{OWA}_\omega(\eta') = \mathrm{OWA}_\omega(\eta'') = 0.8 \cdot 0.2 + 0.2 \cdot 0.1 = 0.18$, thus equally valuating both disachievement vectors. Let us assume there are importance weights $\lambda = (0.75, 0.25)$ for particular disachievements, which means that results under the first disachievement are 3 times more important than those related to the second criterion. To take into account the importance weights in the WOWA aggregation (13) we introduce the piecewise linear function φ:

$$\varphi(\xi) = \begin{cases} 0.8\xi/0.5 & \text{for } 0 \le \xi \le 0.5 \\ 0.8 + 0.2(\xi - 0.5)/0.5 & \text{for } 0.5 < \xi \le 1.0 \end{cases}$$

and calculate weights w_i according to Formula (13) as illustrated in Fig. 5. Actually, $w_1(\lambda, \eta') = \varphi(\lambda_2) = 0.4$ and $w_2(\lambda, \eta') = 1 - \varphi(\lambda_2) = 0.6$ while $w_1(\lambda, \eta'') = \varphi(\lambda_1) = 0.9$ and $w_2(\lambda, \eta'') = 1 - \varphi(\lambda_1) = 0.1$. Hence, $\mathrm{WOWA}_{\omega,\lambda}(\eta') = 0.4 \cdot 0.2 + 0.6 \cdot 0.1 = 0.14$ and $\mathrm{WOWA}_{\omega,\lambda}(\eta'') = 0.9 \cdot 0.2 + 0.1 \cdot 0.1 = 0.19$.

The WOWA may be expressed with a more direct formula where preferential (OWA) weights ω_i are applied to the averages of the corresponding portions of ordered disachievements (quantile intervals) according to the distribution defined by importance weights λ_i.[40] Note that one may alternatively compute the WOWA values by using rational importance weights to replicate the corresponding disachievements and then calculate the OWA aggregations.

Example 6. (Example 5 continued) In the case of the previous example (with importance weights $\lambda = (0.75, 0.25)$), we need to consider three copies of disachievement η_1 and one copy of disachievement η_2 thus generating vectors $\tilde{\eta}' = (0.1, 0.1, 0.1, 0.2)$ and $\tilde{\eta}'' = (0.2, 0.2, 0.2, 0.1)$ of four equally important disachievements. Original preferential weights must be then applied respectively to the

average of the two smallest outcomes and to the average of two largest outcomes. Indeed, we get $\mathrm{WOWA}_{\omega,\lambda}(\eta') = 0.8 \cdot 0.15 + 0.2 \cdot 0.1 = 0.14$ and $\mathrm{WOWA}_{\omega,\lambda}(\eta'') = 0.8 \cdot 0.2 + 0.2 \cdot 0.15 = 0.19$.

This approach can be generalized to any real (possibly irrational) importance weights and the WOWA aggregation can be equivalently defined as follows[41]:

$$\mathrm{WOWA}_{\omega,\lambda}(\eta) = \sum_{i=1}^{n} \omega_i n \int_{\frac{i-1}{n}}^{\frac{i}{n}} \overline{F}_{\eta}^{(-1)}(\xi)\, d\xi, \tag{14}$$

where $\overline{F}_{\eta}^{(-1)}$ is the stepwise function $\overline{F}_{\eta}^{(-1)}(\xi) = \eta_{\langle k \rangle}$ for $\sum_{j<k} \lambda_{\tau(j)} < \xi \leq \sum_{j \leq k} \lambda_{\tau(j)}$, for $k = 1, \ldots, n$ with τ representing the ordering permutation for η (i.e. $\eta_{\tau(k)} = \eta_{\langle k \rangle}$). It can also be mathematically formalized as the quantile function defined as the left-continuous inverse of the cumulative distribution function, i.e. $\overline{F}_{\eta}^{(-1)}(\xi) = \sup\{\nu : \overline{F}_{\eta}(\nu) \geq \xi\}$ for $0 < \xi \leq 1$ with $\overline{F}_{\eta}(\nu) = \sum_{i=1}^{n} \lambda_i \zeta_i(\nu)$ where $\zeta_i(\nu) = 1$ if $\eta_i \geq \nu$ and 0 otherwise.

Example 7. (Example 5 continued) In the case of two disachievement vectors $\eta' = (0.1, 0.2)$ and $\eta'' = (0.2, 0.1)$ (with importance weights $\lambda = (0.75, 0.25)$) from Example 5

$$\overline{F}_{\eta'}^{(-1)}(\xi) = \begin{cases} 0.2 & \text{for } 0 < \xi \leq 0.25 \\ 0.1 & \text{for } 0.25 < \xi \leq 1 \end{cases} \quad \text{and} \quad \overline{F}_{\eta''}^{(-1)}(\xi) = \begin{cases} 0.2 & \text{for } 0 < \xi \leq 0.75 \\ 0.1 & \text{for } 0.75 < \xi \leq 1. \end{cases}$$

Hence, calculating the corresponding WOWA aggregations according to formula (14) one gets

$$\begin{aligned} \mathrm{WOWA}_{\omega,\lambda}(\eta') &= 0.8 \cdot 2(0.2 \cdot 0.25 + 0.1 \cdot 0.25) + 0.2 \cdot 2(0.1 \cdot 0.5) \\ &= 0.8 \cdot 0.15 + 0.2 \cdot 0.1 = 0.14 \end{aligned}$$

and

$$\begin{aligned} \mathrm{WOWA}_{\omega,\lambda}(\eta'') &= 0.8 \cdot 2(0.2 \cdot 0.5) + 0.2 \cdot 2(0.2 \cdot 0.25 + 0.1 \cdot 0.25) \\ &= 0.8 \cdot 0.2 + 0.2 \cdot 0.15 = 0.19. \end{aligned}$$

Applying the WOWA aggregation defined in (14), with decreasing OWA weights ω to individual disachievements ($\psi(\eta) = \mathrm{WOWA}_{\omega,\lambda}(\eta)$), we get the WOWA RPM optimization model[42]:

$$\min \mathrm{WOWA}_{\omega,\lambda}(\eta), \quad \eta_i = \sigma_i(y_i) \ \ \forall i = 1, \ldots, n \tag{15}$$

with piecewise linear individual disachievement function σ_i defined according to (9).

Proposition 3. *For any reference levels $r_i^a \neq r_i^r$, any positive weights ω and λ, if $\bar{y}$ is an optimal solution of the corresponding problem (15), then $\bar{y}$ is a Pareto-optimal solution of the corresponding multiple criteria optimization problem.*

Proposition 3 states that WOWA RPM-optimal solutions are Pareto-optimal. It follows from strict monotonicity of the WOWA aggregation.[43]

Due to the importance weights, the WOWA aggregation allows one to distinguish various individual disachievements. Therefore, contrary to the OWA aggregation, it does not generally provide any direct equitability of individual disachievements. However, still in the case of decreasing OWA weights ω it can be considered equitable with respect to the importance weighted disachievements in the sense that $\eta + \frac{\varepsilon}{\lambda_i}\mathbf{e}_i - \frac{\varepsilon}{\lambda_j}\mathbf{e}_j$ is preferred to η (i.e. $\mathrm{WOWA}_{\omega,\lambda}(\eta + \frac{\varepsilon}{\lambda_i}\mathbf{e}_i - \frac{\varepsilon}{\lambda_j}\mathbf{e}_j) \leq \mathrm{WOWA}_{\omega,\lambda}(\eta)$) whenever $\eta_i < \eta_j$ and $0 < \varepsilon < (\eta_j - \eta_i)\min\{\lambda_i, \lambda_j\}$.[44]

4. Solution Method

4.1. *RPM optimality*

Writing the WOWA RPM optimization model (15) for solving an MMDP, the best compromise solution $V^{C*}: S \to \mathbb{R}^n$, called (WOWA) RPM-optimal, can then be computed with:

$$V^{C*} = \operatorname*{argmin}_{V} \psi\left(\sum_{s\in S} \mu(s)\, V(s)\right), \tag{16}$$

where μ is a probability distribution over initial states and

$$\psi(y) = \mathrm{WOWA}_{\omega,\lambda}(\eta), \quad \eta_i = \sigma_i(y_i) \quad \forall i = 1, \ldots, n \tag{17}$$

with piecewise linear individual disachievement function σ_i defined according to (9).

In an MMDP, the aspiration and reservation levels can be set with respect to the ideal and nadir points. Here, they are respectively the lowest upper bound and the greatest lower bound of nondominated solutions in the criterion space. In fact, in practice, one uses an approximation of the nadir point as it is generally difficult to determine exactly.[32] The ideal point for an MMDP can be computed by solving a standard MDP successively with reward function R_i for $i = 1, \ldots, n$. We denote V^{i*} the vectorial value function optimal for the ith criterion. In a state s, the ideal point then can be formally defined as follows:

$$v_i^I = \sum_{s\in S} \mu(s)\, V_i^{i*}(s) \quad \forall i = 1, \ldots, n.$$

The approximated nadir point is calculated with the V^{i*}'s as

$$v_i^N = \sum_{s\in S} \mu(s) \min_{j=1\ldots n} V_i^{j*}(s) \quad \forall i = 1, \ldots, n.$$

Now, one way to define the aspiration and the reservation levels is as follows by setting two parameters q^a and q^r in interval $[0, 1]$:

$$r_i^a = v_i^N + q^a(v_i^I - v_i^N) \quad \text{and} \quad r_i^r = v_i^N + q^r(v_i^I - v_i^N).$$

Example 8. Continuing Example 3, the ideal point in state 1 is $(9/(1-\gamma), 9/(1-\gamma))$, the approximated nadir point is $(1/(1-\gamma), 1/(1-\gamma))$ and λ can be taken

$((1-\gamma)/8, (1-\gamma)/8)$. Then the RPM-optimal value, which is close to $1/2$ when ε is close to 0, is reached by the policy choosing action b.

This simple example explains why the RPM model is preferred to any weighted sum in multiobjective optimization. Besides, as mentioned previously, the quality of compromise solutions found is even better if we consider randomized policies.

In the next subsections, we present the problems that we need to overcome for finding RPM-optimal (possibly randomized) policies and we propose a solution method for compromise search in MMDPs.

4.2. *RPM optimality is state-dependent*

In a standard MDP, an optimal (w.r.t. the standard preference structure, i.e. maximizing the expectation of discounted total rewards) policy is optimal in every initial state. In an MMDP, a Pareto-optimal policy (i.e. solution of $(v\mathcal{D})$) is Pareto-optimal in every initial state. However, here, the optimality notion based on the RPM scalarizing disachievement function depends on the initial state, i.e. a best compromise policy in a given initial state may not be a best compromise solution in another state. We illustrate this point with a simple example.

Example 9. Consider a deterministic MMDP defined by (S, A, T, R) where $S = \{0, 1, 2\}$, $A = \{\text{Up}, \text{Down}\}$, T and R are represented in Fig. 6. Actions Up (resp. Down) are represented by the arcs above (resp. below). The bi-dimensional rewards are given near the arcs. As state 2 is an absorbing state, one can set the discount factor γ to 1.

Let us consider the simplest RPM scalarizing disachievement function (17) with $\omega_1 = 1 - \varepsilon/2$, $\omega_2 = \varepsilon/2$, $\lambda_1 = \lambda_2 = 1/2$, the aspiration levels set to $r^a = (20, 20)$ and the reservation levels to $r^r = (0, 0)$.

Now, taking $\mu(1) = 1$ and $\mu(0) = \mu(2) = 0$ in (16), the compromise solution in state 1 is given by $(5, 5)$, obtained by choosing action Down. However, taking $\mu(0) = 1$ and $\mu(1) = \mu(2) = 0$, action Up in state 0 followed by action Up in state 1 (valued by $(10, 10)$) yields a better compromise than action Up in state 0 followed by action Down in state 1 (valued by $(5, 15)$).

From this observation, we can conclude that the best compromise policy viewed from one state is not necessarily a best compromise solution viewed from another state.

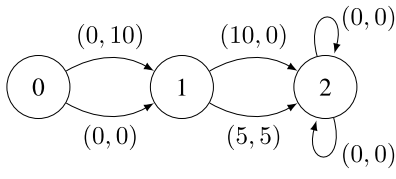

Fig. 6. MMDP of Example 9.

Therefore, to use the RPM optimality as defined by (16), one needs to know the initial state. This is not, in our opinion, a very demanding requirement as for most problems, this information is available. Moreover, when the initial state is unknown, one can instead consider the average of a value function over all possible initial states.

4.3. *Dynamic inconsistency*

Due to the nonlinearity of the RPM scalarizing function, we cannot transform the MMDP into a standard MDP. More specifically, solving an MDP obtained by aggregating the vector rewards of an MMDP with the scalarizing function is not equivalent to optimization (16) in the original MMDP.

Example 9 shows that contrary to standard MDPs, one does not have the following fundamental property:

$$\pi \succ \pi' \Rightarrow (\delta, \pi) \succsim (\delta, \pi'),$$

where π, π' are two policies, δ is a decision rule, (δ, π) (resp. (δ, π')) is the policy consisting of applying first decision rule δ then policy π (resp. π') in the next steps and relation $\succ$ (resp. $\succsim$) is the strict (resp. weak) preference relation over policies induced by the RPM scalarizing function. Without this property, one cannot eliminate dominated subpolicies as they can be improved later. Indeed, if $\pi \succ \pi'$, it could happen that for some δ, one can get a preference reversal $(\delta, \pi') \succ (\delta, \pi)$, thus, making impossible to prune π' using π in a solution method based on dynamic programming.

Example 10. As a consequence, we can no longer rely on algorithms based on dynamic programming, such as value iteration. Indeed, the natural counterpart of value iteration for scalarizing function optimization would have been:

$\forall s \in S, V(s) \leftarrow (0, \ldots, 0)$
repeat
  for all $s \in S$ **do**
    for all $a \in A$ **do**
      $Q(s,a) \leftarrow R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') V(s')$
    end for
    $V(s) \leftarrow \operatorname{argmin}_{a \in A} \psi(Q(s,a))$
  end for
until convergence of V

where ψ is defined by Eq. (15). Let us apply this algorithm on an example.

Consider again the MMDP defined in Example 9. As the only valuation vector that can be obtained in the absorbing state 2 is $(0,0)$, we will skip the computation in that state.

Initialize $V_0(0) = V_0(1) = (0, 0)$. In state 0, the value of applying action Up is given by $Q_1(0, \text{Up}) = (0, 10)$. For action Down, we get $Q_1(0, \text{Down}) = (0, 0)$. Obviously, the best action in state 0 is Up and $V_1(0) = (0, 10)$. In state 1, the value of applying action Up is given by $Q_1(1, \text{Up}) = (10, 0)$. For action Down, we get $Q_1(1, \text{Down}) = (5, 5)$. Here, the best action is Down and $V_1(1) = (5, 5)$.

At the second iteration of the algorithm, we obtain: In state 0, the value of applying action Up is given by $Q_2(0, \text{Up}) = (5, 15)$. For action Down, we get $Q_2(0, \text{Down}) = (5, 5)$. Obviously, the best action is Up and $V_2(1) = (5, 15)$. In state 1, we are in the same situation as at the previous iteration. Therefore, we get $V_2(1) = (5, 5)$ and Down is the best action.

At the third iteration, one can see that the value function has converged and yields a dominated policy as the policy choosing action Up in every state is the best compromise policy.

This last observation motivates us to search for a solution method based on the multiobjective linear program $v\mathcal{D}$.

4.4. *Solution method*

Searching for RPM optimal solution in the set of stationary randomized policies can be better than restricting the search to the set of deterministic policies. As an illustration, consider Fig. 7 which represents value functions of deterministic policies in an initial state for a given MMDP ($n = 2$). Assume that point c is the RPM optimal solution. Now, if we consider randomized policies, we can do much better as shown in Fig. 8.

The valuation vectors of randomized policies are in the convex hull of the valuation vectors of deterministic policies, represented by the gray zone. The dark gray zone represents all feasible valuation vectors that are preferred to point c. The dotted lines linking points a, b and d represent all Pareto-optimal valuation vectors. It is then easy to see that point c is dominated by all the randomized policies that are

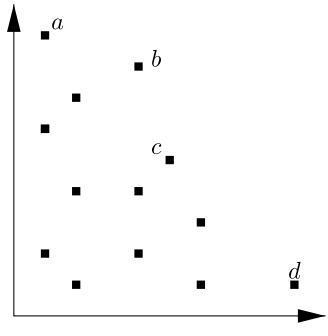

Fig. 7. Valuation vectors.

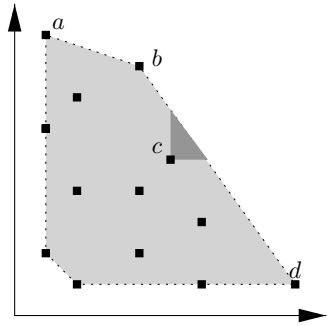

Fig. 8. Better solutions.

both in the dark gray zone and on the dotted line. As any RPM-optimal solution is also Pareto-optimal, one can find better solutions than c when considering also the set of randomized policies.

For this reason, we now focus on the search of an RPM-optimal randomized policy. Although the RPM scalarizing function is not linear, the dynamics of an MMDP remains identical to a standard MDP and thus is linear. For finding RPM-optimal solutions, it is therefore possible to adapt the linear program proposed for finding Pareto-optimal solutions in an MMDP. The WOWA RPM-optimal policy defined according to (16) and (17) can be identified by solving the following optimization problem:

$$\begin{aligned}
&\min \mathrm{WOWA}_{\omega,\lambda}(\eta)\\
&\text{s.t. } \eta_i = \sigma_i(y_i) && \forall i = 1,\dots,n\\
&\quad y_i = \sum_{s\in S}\sum_{a\in A} R_i(s,a)x(s,a) && \forall i = 1,\dots,n\\
&\quad \sum_{a\in A} x(s,a) - \gamma \sum_{s'\in S}\sum_{a\in A} x(s',a)\,T(s',a,s) = \mu(s) && \forall s \in S\\
&\quad x(s,a) \geq 0 && \forall s \in S, \forall a \in A.
\end{aligned} \tag{18}$$

An important advantage of the RPM depends on its easy implementation as an expansion of the original multiple criteria model. Even complicated individual disachievement functions of the form (9) are strictly monotonic and convex, thus allowing for LP implementation.[30] Indeed, since individual disachievement function (9) is piecewise linear convex, it can be expressed in the form:

$$\begin{aligned}
\sigma_i(y_i) &= \max\left\{\beta\frac{y_i - r_i^r}{r_i^r - r_i^a} + 1,\ \frac{y_i - r_i^a}{r_i^r - r_i^a},\ \alpha\frac{y_i - r_i^a}{r_i^r - r_i^a}\right\}\\
&= \max\{\beta\varsigma_i y_i + 1 - \beta\varsigma_i r_i^r,\ \varsigma_i y_i - \varsigma_i r_i^a,\ \alpha\varsigma_i y_i - \alpha\varsigma_i r_i^a\},
\end{aligned} \tag{19}$$

where $\varsigma_i = 1/(r_i^r - r_i^a)$ for $i = 1,\dots,n$. Formula (19) guarantees LP computability with respect to outcomes y_i. Hence, due to the strict monotonicity of the WOWA

aggregation, the RPM-optimal policy defined by model (17) can be identified with the following WOWA optimization on the LP feasible set:

$$\begin{aligned}
&\min \mathrm{WOWA}_{\omega,\lambda}(\eta)\\
&\text{s.t. } \eta_i \geq \beta\varsigma_i y_i + 1 - \beta\varsigma_i r_i^r && \forall i = 1,\dots,n\\
&\quad \eta_i \geq \varsigma_i y_i - \varsigma_i r_i^a && \forall i = 1,\dots,n\\
&\quad \eta_i \geq \alpha\varsigma_i y_i - \alpha\varsigma_i r_i^a && \forall i = 1,\dots,n\\
&\quad y_i = \sum_{s\in S}\sum_{a\in A} R_i(s,a)x(s,a) && \forall i = 1,\dots,n\\
&\quad \sum_{a\in A} x(s,a) - \gamma \sum_{s'\in S}\sum_{a\in A} x(s',a)\,T(s',a,s) = \mu(s) && \forall s \in S\\
&\quad x(s,a) \geq 0 && \forall s \in S,\ \forall a \in A.
\end{aligned}$$

The WOWA criterion is, in general, hard to implement due to the pointwise ordering of individual disachievements. Nevertheless, similarly to the standard OWA optimization,[45] its minimization can be implemented with LP model,[40] in the case of positive and strictly decreasing preferential (OWA) weights $\omega_1 > \omega_2 > \cdots > \omega_n > 0$. Recall that formula (14) defines the WOWA value applying preferential weights ω_i to importance weighted averages within quantile intervals. It may be reformulated with the tail averages (Lorenz components):

$$\mathrm{WOWA}_{\omega,\lambda}(\eta) = \sum_{k=1}^{n} \overline{\omega}_k n L\left(\eta, \lambda, \frac{k}{n}\right) \quad \text{where } L(\eta,\lambda,\xi) = \int_0^{\xi} \overline{F}_{\eta}^{(-1)}(\zeta)\,d\zeta \tag{20}$$

and differential weights

$$\overline{\omega}_k = \omega_k - \omega_{k+1} \quad \text{for } k = 1,\dots,n-1 \quad \text{and} \quad \overline{\omega}_n = \omega_n. \tag{21}$$

Note that the differential weights $\overline{\omega}_i$ are positive in the case of positive and strictly decreasing preferential (OWA) weights $\omega_1 > \omega_2 > \cdots > \omega_n > 0$. Graphs of functions $L(\eta,\lambda,\xi)$ (with respect to ξ) take the form of concave piecewise linear curves, the so-called (upper) absolute Lorenz curves. Values of function $L(\eta,\lambda,\xi)$ for any $0 \leq \xi \leq 1$ can be given by optimization:

$$L(\eta,\lambda,\xi) = \max_{u_i}\left\{\sum_{i=1}^{n} \eta_i u_i : \sum_{i=1}^{n} u_i = \xi,\ 0 \leq u_i \leq \lambda_i\ i = 1,\dots,n\right\}. \tag{22}$$

Introducing dual variable t corresponding to the equation $\sum_{i=1}^{n} u_i = \xi$ and variables d_i corresponding to upper bounds on u_i one gets the following LP dual expression of $L(\eta,\lambda,\xi)$:

$$L(\eta,\lambda,\xi) = \min_{t,d_i}\left\{\xi t + \sum_{i=1}^{n} \lambda_i d_i : \eta_i \leq t + d_i,\ d_i \geq 0\ \forall i = 1,\dots,n\right\}. \tag{23}$$

Hence, the entire WOWA RPM program (17) can easily be linearized as follows:

$$
\begin{aligned}
\min\ & \sum_{k=1}^{n} \overline{\omega}_k z_k \\
\text{s.t. } & z_k = k t_k + n \sum_{i=1}^{n} \lambda_i d_{ik} && \forall k = 1, \ldots, n \\
& \eta_i \le t_k + d_{ik},\ d_{ik} \ge 0 && \forall i, k = 1, \ldots, n \\
& \eta_i \ge \beta \varsigma_i y_i + 1 - \beta \varsigma_i r_i^r && \forall i = 1, \ldots, n \\
& \eta_i \ge \varsigma_i y_i - \varsigma_i r_i^a && \forall i = 1, \ldots, n \\
& \eta_i \ge \alpha \varsigma_i y_i - \alpha \varsigma_i r_i^a && \forall i = 1, \ldots, n \\
& y_i = \sum_{s \in S} \sum_{a \in A} R_i(s,a) x(s,a) && \forall i = 1, \ldots, n \\
& \sum_{a \in A} x(s,a) - \gamma \sum_{s' \in S} \sum_{a \in A} x(s',a) T(s',a,s) = \mu(s) && \forall s \in S \\
& x(s,a) \ge 0 && \forall s \in S, \forall a \in A,
\end{aligned}
$$

where t_k for $k = 1, \ldots, n$ and y_i for $i = 1, \ldots, n$ are unbounded (unrestricted) variables. Actually, variables y_i are introduced only to represent the values $\sum_{s \in S} \sum_{a \in A} R_i(s,a) x(s,a)$, but their elimination does not simplify the model. The coefficients $\overline{\omega}_k$ are defined as differential OWA weights (21).

Our previous observation concerning the state-dependency of the RPM optimality suggests that contrary to standard MDPs, RPM-optimal solutions depend on μ. When we do not know the initial state, 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 for $s = s_0$ and to 0 otherwise. The solution found by the linear program does not specify which action to choose for any state s for which $\mu(s) = 0$ and that is not reachable from the initial state as such a state does not impact the value of the RPM-optimal policy.

4.5. *Preference modeling*

The reference point methods support preference modeling by the reference levels in the sense that various efficient solutions can be selected by appropriate setting of the reference levels. Indeed any properly efficient solutions[46] with bounded tradeoffs can be generated by the standard RPM.[47] Recall that an outcome vector $\bar{\mathbf{y}}$ is *properly nondominated with tradeoffs bounded by* Δ, if and only if for any attainable outcome vector $\mathbf{y}$ the implication

$$y_i \text{ is better than } \bar{y}_i \quad \text{and} \quad \bar{y}_k \text{ is better than } y_k \Rightarrow |y_i - \bar{y}_i| \le \Delta |\bar{y}_k - y_k| \tag{24}$$

is valid for any $i, k = 1, \ldots, n$.

Proposition 4. *For any positive importance weights* λ_i, *if* $\bar{y}$ *is properly nondominated with tradeoffs bounded by* Δ, *then there exist positive and strictly decreasing preferential* (*OWA*) *weights* $\omega_1 > \omega_2 > \cdots > \omega_n > 0$, *aspirations levels* r_i^a

and reservation levels r_i^r *such that* $\bar{y}$ *is an optimal solution of the corresponding problem* (15).

Proof. Let $\bar{y}$ be an attainable outcome vector properly nondominated with tradeoffs bounded by Δ. Let us set positive and strictly decreasing preferential (OWA) weights $\omega_1 > \omega_2 > \cdots > \omega_n > 0$ with large enough ω_1 to fulfill inequality $\Delta \leq n\bar{\lambda}\omega_1/(\alpha - \alpha n\bar{\lambda}\omega_1)$ where $\bar{\lambda} = \min_{i=1\ldots n}\lambda_i$. Further, for all $i = 1, \ldots, n$, let us set the reference levels as $r_i^a = \bar{y}_i$ and $r_i^r = \bar{y}_i - 1$ in the case of maximized criterion and $r_i^r = \bar{y}_i + 1$ in the case of minimized one. We will show that $\bar{y}$ together with disachievements $\bar{\eta}_i = \sigma_i(\bar{y}_i)$ defined according to formula (9) form an optimal solution of the corresponding RPM problem (15). Suppose there exists an attainable outcome vector y such that for its disachievements $\eta_i = \sigma_i(y_i)$, $\forall i$ one gets better scalarizing disachievement value $\mathrm{WOWA}_{\omega,\lambda}(\eta) < \mathrm{WOWA}_{\omega,\lambda}(\bar{\eta})$. Note that $\bar{\eta}_i = 0$ for all $i = 1, \ldots, n$. Hence, following formula (13):

$$\begin{aligned}\mathrm{WOWA}_{\omega,\lambda}(\eta) - \mathrm{WOWA}_{\omega,\lambda}(\bar{\eta}) &= \sum_{i=1}^{n} w_i(\lambda,\eta)\eta_{\langle i\rangle} - \sum_{i=1}^{n} w_i(\lambda,\eta)\bar{\eta}_{\langle i\rangle} \\ &= \sum_{i=1}^{n} w_i(\lambda,\eta)(\eta_{\tau(i)} - \bar{\eta}_{\tau(i)}),\end{aligned}$$

where τ is the ordering permutation for the disachievement vector η. Moreover, due to the Pareto-optimality of $\bar{y}$, $\bar{y}_{\tau(1)}$ is a better outcome value than $y_{\tau(1)}$ and $\eta_{\tau(1)} - \bar{\eta}_{\tau(1)} \geq (r^a_{\tau(1)} - r^r_{\tau(1)})(\bar{y}_{\tau(1)} - y_{\tau(1)}) \geq 0$. Further, due to formula (9):

$$\eta_{\tau(i)} - \bar{\eta}_{\tau(i)} \geq -\alpha(r_i^a - r_i^r)(y_{\tau(i)} - \bar{y}_{\tau(i)})$$

whenever $y_{\tau(i)}$ is better than $\bar{y}_{\tau(i)}$ (i.e. $(r_i^a - r_i^r)(y_{\tau(i)} - \bar{y}_{\tau(i)}) > 0$) and $\eta_{\tau(i)} - \bar{\eta}_{\tau(i)} \geq 0$ otherwise. Therefore, taking advantages of the proper efficiency inequalities (24) for $k = \tau(1)$ one gets

$$\begin{aligned}\sum_{i=2}^{n} w_i(\lambda,\eta)(\eta_{\tau(i)} - \bar{\eta}_{\tau(i)}) &\geq -\sum_{i=2}^{n} w_i(\lambda,\eta)\Delta(r^a_{\tau(1)} - r^r_{\tau(1)})(\bar{y}_{\tau(1)} - y_{\tau(1)}) \\ &\geq -w_1(\lambda,\eta)(r^a_{\tau(1)} - r^r_{\tau(1)})(\bar{y}_{\tau(1)} - y_{\tau(1)}) \\ &\geq -w_1(\lambda,\eta)(\eta_{\tau(1)} - \bar{\eta}_{\tau(1)})\end{aligned}$$

which contradicts the inequality $\sum_{i=1}^{n} w_i(\lambda,\eta)\eta_{\langle i\rangle} < \sum_{i=1}^{n} w_i(\lambda,\bar{\eta})\bar{\eta}_{\langle i\rangle}$, thus confirming optimality of $\bar{y}$ for the corresponding WOWA RPM problem (15). □

As our MMDP problems fit the LP formulation, there exists $\Delta > 0$ such that any Pareto-optimal solution is represented by the properly nondominated outcome vector with tradeoffs bounded by Δ.[46] Therefore, Proposition 4 justifies modeling preferences with reference levels as any Pareto-optimal policy can be obtained. However, in an MMDP one has very limited knowledge of criterion values and the corresponding aspiration and reservation levels can be set only roughly as a function of the ideal and nadir points. Therefore, an important advantage of the WOWA RPM-optimality formulation (16) and (17) is the use of importance weights allowing

the importance of various individual disachievements to be distinguished. Indeed, one can improve a particular disachievement by increasing the corresponding importance weight as suggested by the following proposition.

Proposition 5. *Let* $\bar{\mathbf{y}}$ *be an attainable outcome vector properly nondominated with tradeoffs bounded by* Δ. *Let* $\mathbf{y}$ *be an attainable nondominated outcome vector with coordinate* y_{i_o} *worse than* $\bar{y}_{i_o}$ *(i.e.* $\bar{\eta}_{i_o} < \eta_{i_o}$*). For any positive and strictly decreasing preferential (OWA) weights* $\omega_1 > \omega_2 > \cdots > \omega_n > 0$, *for any aspirations levels* r_i^a *and reservation levels* r_i^r, *and any criterion* i_o, *there exist importance weights* $\lambda_1, \ldots, \lambda_n$ *such that* $\mathrm{WOWA}_{\omega,\lambda}(\bar{\eta}) < \mathrm{WOWA}_{\omega,\lambda}(\eta)$.

Proof. Note that, following (20), we have

$$\mathrm{WOWA}_{\omega,\lambda}(\bar{\eta}) - \mathrm{WOWA}_{\omega,\lambda}(\eta) = \sum_{i=k}^{n} \overline{\omega}_k n \left[L\left(\bar{\eta}, \lambda, \frac{k}{n}\right) - L\left(\eta, \lambda, \frac{k}{n}\right) \right],$$

where $\overline{\omega}_k$ are positive differential OWA weights defined as (21) and

$$L(\bar{\eta}, \lambda, \xi) - L(\eta, \lambda, \xi) = \max_{u \in U(\lambda,\xi)} \sum_{i=1}^{n} \bar{\eta}_i u_i - \max_{u \in U(\lambda,\xi)} \sum_{i=1}^{n} \eta_i u_i$$

with $U(\lambda, \xi) = \{u = (u_1, \ldots, u_n) : \sum_{i=1}^{n} u_i = \xi,\ 0 \le u_i \le \lambda_i\ i = 1, \ldots, n\}$. Hence,

$$L(\bar{\eta}, \lambda, \xi) - L(\eta, \lambda, \xi) \le \sum_{i=1}^{n} \bar{\eta}_i \bar{u}_i(\xi) - \sum_{i=1}^{n} \eta_i \bar{u}_i(\xi) = \sum_{i=1}^{n} (\bar{\eta}_i - \eta_i) \bar{u}_i(\xi),$$

where $\bar{u}(\xi)$ is an optimal solution to the problem $\max_{u \in U(\lambda,\xi)} \sum_{i=1}^{n} \bar{\eta}_i u_i$.

There exists i such that $\eta_i < \bar{\eta}_i$ (otherwise y would be dominated). Then, following (24), $\bar{\eta}_i - \eta_i \le (\eta_{i_o} - \bar{\eta}_{i_o})\Delta$. As $\beta/\alpha > 1$, $\bar{\eta}_i - \eta_i \le (\eta_{i_o} - \bar{\eta}_{i_o})\Delta\beta/\alpha$ and

$$\begin{aligned} &\mathrm{WOWA}_{\omega,\lambda}(\bar{\eta}) - \mathrm{WOWA}_{\omega,\lambda}(\eta) \\ &\quad \le \sum_{k=1}^{n} \overline{\omega}_k n \left[\bar{u}_{i_o}\left(\frac{k}{n}\right) - \frac{\Delta\beta}{\alpha} \sum_{i \ne i_o} \bar{u}_i\left(\frac{k}{n}\right) \right] (\bar{\eta}_{i_o} - \eta_{i_o}) \\ &\quad \le n \left[\omega_n \lambda_{i_o} - \frac{\Delta\beta\omega_1}{\alpha} \sum_{i \ne i_o} \lambda_i \right] (\bar{\eta}_{i_o} - \eta_{i_o}) \end{aligned}$$

since $\bar{u}_{i_o}(\frac{k}{n}) \ge 0$ for all k, $\bar{u}_{i_o}(\frac{n}{n}) = \lambda_{i_o}$, and $\bar{u}_i(\frac{k}{n}) \le \lambda_i$ for all i. Thus, for sufficiently large λ_{i_o} (e.g. $\lambda_{i_o} > \Delta\beta\omega_1/(\Delta\beta\omega_1 + \alpha\omega_n)$) one gets $\mathrm{WOWA}_{\omega,\lambda}(\bar{\eta}) < \mathrm{WOWA}_{\omega,\lambda}(\eta)$.

As a side note, a better bound for λ_{i_o} can be found when $\bar{\eta}_{i_o} \ge \bar{\eta}_i$ for all i. Following (24), if $\eta_i < \bar{\eta}_i$, then $\bar{\eta}_i - \eta_i \le \Delta(\eta_{i_o} - \bar{\eta}_{i_o})$ and

$$L(\bar{\eta}, \lambda, \xi) - L(\eta, \lambda, \xi) \le \left[\bar{u}_{i_o}(\xi) - \Delta \sum_{i \ne i_o} \bar{u}_i(\xi) \right] (\bar{\eta}_{i_o} - \eta_{i_o}),$$

where $\bar{u}_{i_o}(\xi) = \min\{\xi, \lambda_{i_o}\}$ while $\bar{u}_i(\xi) \le \min\{\xi - \bar{u}_{i_o}(\xi), \lambda_i\}$ for all $i \ne i_o$. Hence, for large enough λ_{i_o} (e.g. $\lambda_{i_o} > \Delta/(1 + \Delta)$) one gets $L(\bar{\eta}, \lambda, \xi) < L(\eta, \lambda, \xi)$ for any $0 < \xi \le 1$ and thereby $\mathrm{WOWA}_{\omega,\lambda}(\bar{\eta}) < \mathrm{WOWA}_{\omega,\lambda}(\eta)$. □

Proposition 5 states that having an WOWA RPM-optimal solution with not satisfactory disachievement for criterion i_o, one may increase importance of this criterion, e.g. by setting new importance weights $\lambda'_{i_o} = (\lambda_{i_o} + \hat{\lambda})/(1 + \hat{\lambda})$ and $\lambda'_i = \lambda_i/(1 + \hat{\lambda})$ for all $i \neq i_o$. For sufficiently large increment $\hat{\lambda}$, following Proposition 5 it will exclude solution with worse disachievements for criterion i_o.

Recall Example 4, where solution S3 from Table 1 could not to be selected as the best one when using the standard OWA aggregation of individual disachievements without importance weights. One may notice that introducing high importance weight for the fourth criterion (say $\lambda_4 = 0.85$, $\lambda_1 = \lambda_2 = \lambda_3 = 0.05$) and using the same OWA weights $\omega = (0.5, 0.3, 0.15, 0.05)$ one gets $\mathrm{WOWA}_{\omega,\lambda}(\eta(S3)) < \mathrm{WOWA}_{\omega,\lambda}(\eta(S2))$ and $\mathrm{WOWA}_{\omega,\lambda}(\eta(S3)) < \mathrm{WOWA}_{\omega,\lambda}(\eta(S1))$ which enables selection of S3 as the most preferred solution.

5. Experimental Results

We tested our solution method on two different problems: the navigation problem over an $N \times N$ grid and the inventory control problem. All the experiments were run using CPLEX 12.1 on a PC (Intel Core 2 CPU 2.66 Ghz) with 4 GB of RAM. In the partial disachievement function (9), we set $\beta = 10$ and $\alpha = 0.1$. The other parameters for an objective i are set with respect to its ideal point v_i^I, computed by optimizing objective i as the single objective. We set r_i^r to 25% of v_i^I and r_i^a to 75% of v_i^I when the ideal value is positive and the reverse otherwise.

5.1. *Navigation problem*

In the navigation problem over $N \times N$ grid, the robot can choose among four actions: (L)eft, (U)p, (R)ight, (D)own. Figure 9 gives the transitions for action (R)ight. The whole transition function can then be recovered by symmetry and rotation.

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. For this problem, we ran two series of experiments. As in real problems, criteria are generally conflicting. 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

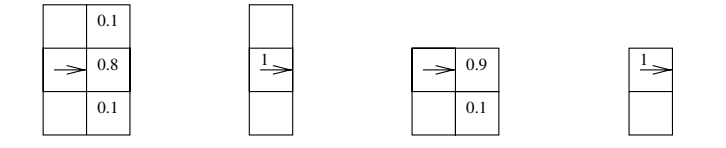

Fig. 9. Transitions.

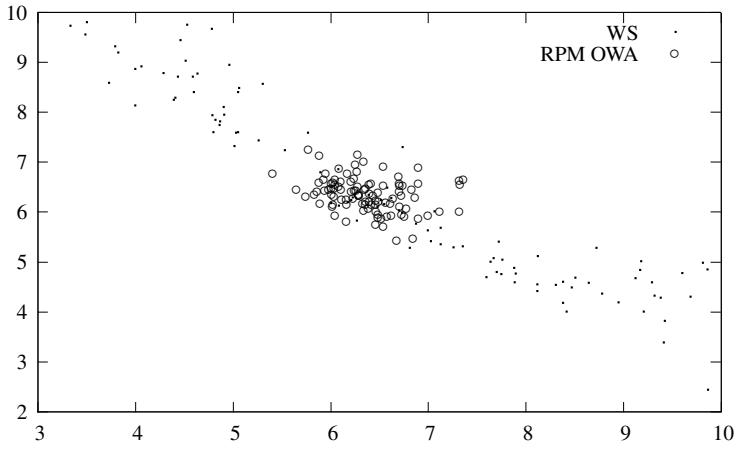

Fig. 10. First experiments.

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 over 100 experiments are shown in Fig. 10. One point on that figure (a dot when the solution is obtained by maximizing the weighted sum with equal weights and a circle when optimizing RPM OWA with exponentially decreasing weights) represents the optimal value function in the initial state in one instance. Naturally, for some instances, maximizing the weighted sum can yield a balanced solution. But, in most cases, it gives a bad compromise solution. Figure 10 shows that we do not have any control on tradeoffs obtained with a weighted sum. On the contrary, when using the RPM OWA, the profile of the solutions seems to be more balanced.

To show the effectiveness of our approach, we ran a second set of experiments on pathological instances of the navigation problem. All rewards are drawn randomly as for the first set of experiments. Then, for each action of the initial state, we choose randomly one of the criteria and add a constant (here, arbitrarily set to 5). Then by construction, the value functions of all deterministic policies in the initial state are unbalanced. The value functions of optimal policies (w.r.t. the weighted sum and RPM OWA) are represented in Fig. 11. Maximizing the weighted sum only gives very unbalanced solutions as it can only reach deterministic policies. Reassuringly, the solutions found by optimizing RPM OWA are still well balanced.

On the navigation problem, we also made some experiments when the RPM WOWA criterion is used to show a better controllability on the profile of solutions that one wants to find. This is done by setting importance weights on objectives. The results for 100 instances and three sets of importance weights are plotted in Fig. 12. The circles are the solutions found for RPM WOWA with equal weights (which is simply RPM OWA). The triangles are the obtained solutions when one gives more weight for the first objective (0.75 for the first objective and 0.25 for the second). The dots are the solutions found when more weight is given to the second objective

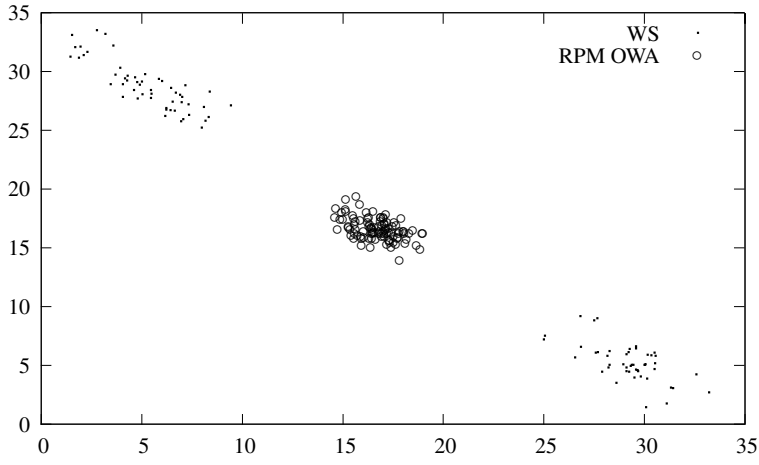

Fig. 11. Second experiments.

(0.25 for the first objective and 0.75 for the second). We can notice that the importance weights provide some controllability on the optimal tradeoffs contrary to RPM OWA.

Finally for the navigation problem, we list the average execution time as a function of the problem size in Table 2. The first column n shows the number of objectives. The second column gives the number of states of the problem. Finally, column TW gives the execution time for the weighted sum approach, column TRO corresponds to the execution time of RPM OWA and column TRW gives that of RPM WOWA. All the times are averages over 20 experiments and are given in seconds.

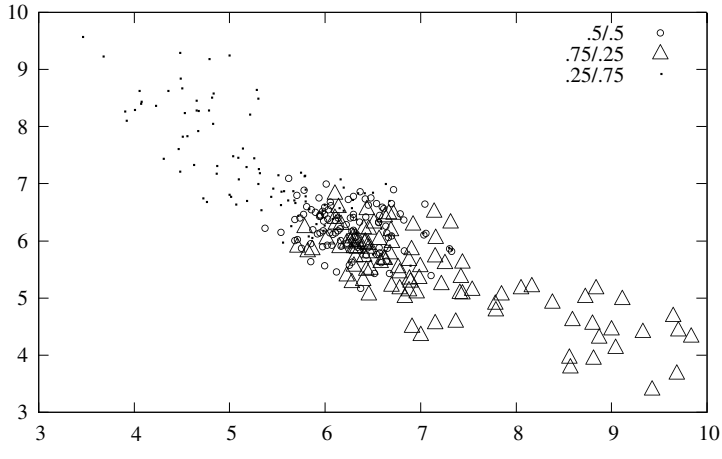

Fig. 12. Controllability.

Table 2. Average execution time in seconds.

n	Size	TW	TRO	TRW
2	400	0.17	0.48	0.46
2	2500	5.13	15.06	15.12
2	10000	151.51	417.02	422.06
4	400	0.12	0.75	0.76
4	2500	5.20	28.21	28.27
4	10000	154.00	821.27	829.83
8	400	0.12	1.30	1.30
8	2500	4.96	50.62	50.72
8	10000	158.26	1514.21	1538.19

5.2. *Inventory problem*

We have also tested our approach on the single-product inventory problem. At each time step, a warehouse manager decides how many products to order subject to a stochastic customer demand. For simplicity, we assume that there is no delivery delay. The aim of the DM is to minimize her operating costs: the cost of stocking unsold products, the cost of ordering products and the shortage cost when the demand is higher than the stock level. Although these three types of costs are expressed in monetary terms, we do not sum them as it is customary. These costs indeed refer to various types of consequences that have different impacts in terms of storage capacity dimensioning, order policy and client's satisfaction. These various consequences are not easily commensurable and may not compensate one another. For this reason, it is often interesting to keep these costs separate and to treat the inventory problem as a multiobjective one.

This problem can be modeled as an MDP. A state s represents the number of products in stock when it is positive. For modeling reasons, we assume that s can be negative, which would mean that the current stock level is null and that at the last iteration, the customer demands have not completely been met, i.e. s extra products could have been sold if the stock level were high enough at the previous iteration. So, the stock level is given by $\max(0, s)$. We assume that M is the maximum capacity of the warehouse. An action a represents the quantity of ordered products. We assume that the customer demand D is a random variable, which follows a (stationary) Poisson law in our experiments. Then, when action a is taken in state s, the next state s' can be expressed as follows:

$$s' = \min(M, \max(0, s) + a) - D$$

as the quantity of products in stock after an order cannot be higher than M. The transition function can be inferred from the probability distribution of D. Rewards $R(s, a, s')$ (depending on the next state here) are defined as triplets $(R_s(s, a, s'), R_o(s, a, s'), R_l(s, a, s'))$ where R_s represents the stock cost, R_s the order cost and R_l

Table 3. Average execution time in seconds.

M	TW	TRO	TRW
10	0.02	0.04	0.04
100	0.05	0.17	0.18
1000	9.79	36.96	36.59

the shortage cost. They are defined as follows:

$$R_s(s, a, s') = \max(0, s') * u_s, \tag{25}$$

$$R_o(s, a, s') = (a * u_o + u_f) I_{a \neq 0}, \tag{26}$$

$$R_l(s, a, s') = -\min(0, s'), \tag{27}$$

where u_s is the marginal cost of stocking one product, u_o is the marginal cost of ordering one product, u_f is the fixed cost of an order, $I_{a \neq 0} = 1$ if a is not null and $I_{a \neq 0} = 0$ otherwise. For the shortage cost, we simply count the number of unsatisfied demand units as it is difficult to estimate the costs in dollar term.

In Table 3, we list the computation times (in seconds) for the inventory problem with different values of M. All the times are averaged over 20 runs.

6. Conclusion

In this paper, we have presented a compromise programming approach to MMDPs. It relies on a reference point method designed to generate a policy yielding an expected-utility vector as close as possible to a reference point. One particularity of our approach is that the overall value of a policy is measured with the WOWA of individual disachievements, which comprises many standard scalarizing functions (weighted sum, OWA, weighted Tchebycheff norm) as special cases and provides new interesting features in terms of discrimination and controllability. We demonstrated that an RPM optimal policy depends on the initial state. Moreover, we also explained why standard schemes for MDPs based on dynamic programming do not apply to RPM WOWA optimality. Besides, we showed how one can find better solutions through considering all randomized policies. All these observations justify using mathematical programming to search for an RPM-optimal policy. Although the scalarizing function is not linear, we have provided an LP-solvable formulation of the problem. In all the experiments performed, the reference point method with ordered aggregations 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. Note that the RPM method proposed here is quite general and could be applied to any other multiobjective problem with linear constraints.

Acknowledgments

The research by W. Ogryczak was partially supported by the Polish National Budget Funds 2010−2013 for science under the grant N N514 044438.

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).

The authors are indebted to an anonymous referee for his helpful comments.

References

1. R. Bellman, A Markovian decision process, *Journal of Mathematics and Mechanics* **6** (1957) 679−684.
2. R. A. Howard, *Dynamic Programming and Markov Processes* (The M.I.T. Press, Cambridge, Mass, 1960).
3. D. J. White, A survey of applications of Markov decision processes, *Journal of the Operational Research Society* **44** (1993) 1073−1096.
4. T. M. Leschine, H. Wallenius and W. A. Verdini, Interactive multiobjective analysis and assimilative capacity based ocean disposal decision, *European Journal Operational Research* **56** (1992) 278−289.
5. O. Sigaud and O. Buffet (eds.), *Markov Decision Processes in Artificial Intelligence* (John Wiley and Sons, Hoboken, NJ, 2010).
6. S. P. M. Choi and J. Liu, Markov decision approach for time-constrained trading in electronic marketplace, *International Journal of Information Technology and Decision Making* **1** (2002) 511−524.
7. A. I. Mouaddib, Multi-objective decision-theoretic path planning, in *IEEE International Conference Robotics and Automation* Vol. 3 (2004), pp. 2814−2819.
8. K. Chatterjee, R. Majumdar and T. A. Henzinger, Markov decision processes with multiple objectives, in *The Symposium on Theoretical Aspects of Computer Science 2006*, LNCS Vol. 3884 (2006), pp. 325−336.
9. C. Boutilier, Sequential optimality and coordination in multiagent systems, in *Proceedings of the International Joint Conference on Artificial Intelligence* (Morgan Kaufmann, San Francisco, 1999), pp. 478−485.
10. C. Guestrin, D. Koller and R. Parr, Multiagent planning with factored MDPs, in *Advances in Neural Information Processing Systems (NIPS)* (2001), pp. 1523−1530.
11. W. Ogryczak, P. Perny and P. Weng, On minimizing ordered weighted regrets in multiobjective Markov decision processes, in *International Conference on Algorithmic Decision Theory*, LNAI, Vol. 6992 (2011), pp. 190−204.
12. B. Golden and P. Perny, Infinite order Lorenz dominance for fair multiagent optimization, in *International Conference on Autonomous Agents and Multiagent Systems* (2010), pp. 383−390.
13. W. Ogryczak, A. Wierzbicki and M. Milewski, A multi-criteria approach to fair and efficient bandwidth allocation, *OMEGA* **36** (2008) 451−463.
14. R. E. Steuer, *Multiple Criteria Optimization* (John Wiley and Sons, New York, 1986).
15. J. Ignatius, A. Mustafa, M. Jantan, C. P. Lim, T. Ramayah and J. Yeap Ai Leen, A multi-objective sensitivity approach to training providers evaluation and quota allocation planning, *International Journal of Information Technology and Decision Making* **10** (2011) 147−174.

16. P. Korhonen, Interactive methods, in *Multiple Criteria Decision Analysis: State of the Art Surveys*, J. Figueira, S. Greco and M. Ehrgott (eds.), (Springer, Boston, 2005), pp. 641–666.
17. M. L. Puterman, *Markov Decision Processes — Discrete Stochastic Dynamic Programming* (John Wiley and Sons, Hoboken, NJ, 1994).
18. N. Furukawa, Vector-valued Markovian decision processes with countable state space, in *Recent Developments in Markov Decision Processes*, eds., R. Hartley, L. C. Thomas, D. J. White (Academic Press, New York, 1980), pp. 205–223.
19. B. Viswanathan, V. V. Aggarwal and K. P. K. Nair, Multiple criteria Markov decision processes, *TIMS Studies in the Management Sciences* **6** (1977) 263–272.
20. D. J. White, Multi-objective infinite-horizon discounted Markov decision processes, *Journal of Mathematical Analysis and Applications* **89** (1982) 639–647.
21. P. Hansen, Bicriterion path problems, in *Multiple Criteria Decision Making: Theory and Applications*, Lecture Notes in Economics and Mathematical Systems 177, G. Fandel and T. Gal (eds.), (Springer, Berlin, Heidelberg, 1980), pp. 109–127.
22. A. P. Wierzbicki, On the completeness and constructiveness of parametric characterizations to vector optimization problems, *OR Spektrum* **8** (1986) 73–87.
23. M. Ehrgott, *Multicriteria Optimization* (Springer, Berlin, Heidelberg, 2005).
24. A. P. Wierzbicki, A mathematical basis for satisficing decision making, *Mathematical Modelling* **3** (1982) 391–405.
25. W. Ogryczak, On goal programming formulations of the reference point method, *Journal of the Operational Research Society* **52** (2001) 691–698.
26. A. P. Wierzbicki, M. Makowski and J. Wessels, *Model Based Decision Support Methodology with Environmental Applications* (Kluwer, Dordrecht, 2000).
27. K. Miettinen and M. M. Mäkelä, On scalarizing functions in multiobjective optimization, *OR Spectrum* **24** (2002) 193–213.
28. M. Kadziński and R. Słowiński, Interactive robust cone contraction method for multiple objective optimization problems, *International Journal of Information Technology and Decision Making* **11** (2012) 327–357.
29. A. Lewandowski and A. P. Wierzbicki, *Aspiration Based Decision Support Systems — Theory, Software and Applications* (Springer, Heidelberg, 1989).
30. W. Ogryczak, K. Studziński and K. Zorychta, DINAS: A computer-assisted analysis system for multiobjective transshipment problems with facility location, *Computers and Operations Research* **19** (1992) 637–647.
31. J. Granat and M. Makowski, ISAAP — Interactive specification and analysis of aspiration-based preferences, *European Journal of Operational Research* **122** (2000) 469–485.
32. M. Ehrgott and D. Tenfelde-Podehl, Computation of ideal and Nadir values and implications for their use in MCDM methods, *European Journal of Operational Research* **151** (2003) 119–139.
33. P. Korhonen, S. Salo and R. E. Steuer, A heuristic for estimating nadir criterion values in multiple objective linear programming, *Operations Research* **45** (1997) 751–757.
34. R. R. Yager, On ordered weighted averaging aggregation operators in multicriteria decision making, *IEEE Transactions on Systems, Man, and Cybernetics* **18** (1988) 183–190.
35. R. R. Yager, On the analytic representation of the Leximin ordering and its application to flexible constraint propagation, *European Journal of Operational Research* **102** (1997) 176–192.
36. W. Ogryczak and B. Kozłowski, Reference point method with importance weighted ordered partial achievements, *TOP* **19** (2011) 380–401.

37. V. Torra, The weighted OWA operator, *International Journal of Intelligent Systems* **12** (1997) 153–166.
38. V. Torra and Y. Narukawa, *Modeling Decisions Information Fusion and Aggregation Operators* (Springer, Berlin, Heidelberg, 2007).
39. J. Quiggin, *Generalized Expected Utility Theory, The Rank-Dependent Model* (Kluwer Academic, Amsterdam, 1993).
40. W. Ogryczak and T. Śliwiński, On optimization of the importance weighted OWA aggregation of multiple criteria, in *International Conference on Computer Science and its Applications*, LNCS Vol. 4705 (2007), pp. 804–817.
41. W. Ogryczak and T. Śliwiński, On efficient WOWA optimization for decision support under risk, *International Journal of Approximate Reasoning* **50** (2009) 915–928.
42. W. Ogryczak, Reference point method with importance weighted partial achievements, *Journal of Telecommunications and Information Technology* **4/2008** (2008) 17–25.
43. W. Ogryczak, Ordered weighted enhancement of preference modeling in the reference point method for multiple criteria optimization, *Soft Computing* **14** (2010) 345–360.
44. W. Ogryczak, On principles of fair resource allocation for importance weighted agents, in *IEEE International Conference on Social Informatics* (2009), pp. 57–62.
45. W. Ogryczak and T. Śliwiński, On solving linear programs with the ordered weighted averaging objective, *European Journal of Operational Research* **148** (2003) 80–91.
46. A. M. Geoffrion, Proper efficiency and the theory of vector maximization, *Journal of Mathematical Analysis and Applications* **22** (1968) 618–630.
47. A. P. Wierzbicki, Reference point approaches, in *Multicriteria Decision Making: Advances in MCDM Models, Algorithms, Theory, and Applications*, T. Gal, T. Stewart and T. Hanne (eds.), (Kluwer, Boston, 1999), pp. 9.1–9.39.