Vector-Value Markov Decision Process for multi-objective stochastic path planning

Abdel-Illah Mouaddib

Vector-Value Markov Decision Process for multi-objective stochastic path planning

Abdel-Illah Mouaddib*
GREYC – Université de Caen, Bd Maréchal Juin, Caen Cedex, France

Abstract. The problem of path planning in stochastic environments where the shortest path is not always the best one is a challenging issue required in many real-world applications such as autonomous vehicles, robotics, logistics, etc. ... In this paper, we consider the problem of path planning in stochastic environments where the length of the path is not the unique criterion to consider. We formalize this problem as a multi-objective decision-theoretic path planning and we transform this latter into 2VMDP (Vector-Valued Markov Decision Process). We show, then, how we can compute a policy balancing between different considered criteria. We describe different techniques that allow us to derive an optimal policy where it is hard to express the expected utilities, rewards and values with a unique numerical measure. Firstly, we examine different existing approaches based on preferences and we define notions of optimality with preferred solutions and secondly we present approaches based on egalitarian social welfare techniques. Finally, some experimental results have been developed to show the feasibility of the approach and the benefit of this approach on the single-objective techniques.

Keywords: Planning under uncertainty, Markov Decision Processes, multi-criteria decision making, autonomous systems

1. Introduction

In real-world robotic applications [17,25], the solution quality is frequently a function of multiple criteria. One of the applications concerned with such approaches is the multi-criteria path planning where the shortest path may not always be the most efficient means of getting from start to destination. There are many other attributes of a path that may be desirable in addition to distance. One example would be the smoothness of a path. Indeed, the smoothness of a path when using real robots is important because of resource consumption and of localization. When a robot uses odometry for localization, paths with many slopes make the localization with odometry very difficult. Also using actions to rotate is energy consuming rather than a linear movement. Another advantages to reduce the slopes in the paths is that the robot doesn't need to reduce its speed. Consequently, considering the smoothness of the path in addition to the length of the path is more effective than classical approaches where only one criterion is taken into account. The development of an effective multi-criteria selection method may be difficult. Comparing two candidate solutions using multi-criteria may need a sophisticated technique. Preferences on different solutions and criteria are an adequate way to characterize the interesting solutions. For example, the designer can prefer the smoothness and a long path rather than a short and difficult path. We mention that this example of path planning with multiple criteria length, smoothness, energy, ... will be used at many places to illustrate our claims.

The sequential decision process applied to such applications yields a result having a quality described as a vector Z of quality criteria $(z_1, z_2, \ldots, z_n)$. In the path planning applications the quality could be the length of the path and also the number of slopes in the path. Each decision improves the quality of the result according to a subset of criteria.

However, the use of vectors to represent multi-criteria quality leads to new problems where numerical, additive utility, reward and value functions are not available. Since the utility of multi-criteria quality can be represented as a *multi-dimensional utility*, we meet

*Corresponding author: Abdel-Illah Mouaddib, GREYC – Université de Caen, Bd Maréchal Juin, BP 5186, F14032 Caen Cedex, France. E-mail: mouaddib@info.unicaen.fr.

the problem of how to decide that a multi-dimensional utility dominates another since operator of comparison such as *Max* is used to compare single values but it is unable to compare vectors . This problem has been addressed in utility theory as a representation of the utility function issue [6] as well as preference elicitation [5] and pareto-dominance approaches [13]. Most of the approaches developed in utility theory select the most preferred action in a local state without considering the fluctuation in the rest of states while the approaches based on the preference elicitation are, in some cases, not enough to quantify the utility. This issue has also attracted some work in extending MDP (Markov Decision Process) to vector-value function [18,35]. These approaches based on *leximin* techniques consider that the value of criteria are of the same scale and the same nature. In addition to these classical approaches, we present an extension of *leximin* to be independent of the scale of criteria by using a ratio regret measure.

In this paper we discuss how to construct an optimal policy using a multi-dimensional utility. For that, we focused our study on the use of preference-based techniques. In fact, it is more appropriate to represent preference over states with an ordering relation rather than with additive utilities and rewards. The problem of extending decision-theoretic planning to multiple objectives application is important because it allows decision-theoretic techniques to be more general. However, this extension leads to new interesting but difficult problems. These problems have attracted the attention of new interests for dealing with multi-attribute value functions and generating optimal policy in decision process and advocating a qualitative version of decision theory [4,28,31]. The approach we develop in this paper shows how multi-objective decision-theoretic planning offers new potential for new applications of AI such as multi-objective path planning.

The paper is organized into 10 sections. After the introduction, Section 2 describes how the multi-criteria path planning can be considered as a multi-objective optimization problem and the limits of the existing techniques. Section 3 considers the stochastic aspect of the problem and presents a formal framework of multi-criteria stochastic path planning using Markov Decision Process (MDP) techniques and how to extend the existing algorithms of solving MDPs to MDPs with multi-attribute value functions, named vector-valued MDPs (2V-MDP). Section 4 shows how to consider bounded resources constraints in this model. Section 5 gives an overview of techniques of multi-criteria optimization based on different social welfare approaches and preference ordering techniques. Section 6 describes how to use the social welfare techniques in value iteration algorithms to solve 2V-MDPs. Section 7 introduces a new technique based on regret measure to solve 2V-MDPs. Section 8 presents the validation and experimental results. Section 9 gives a description of related work and in Section 10 we conclude the paper and we give the remaining issues which are let to the future works.

2. Multi-objective decision optimization problem

Let $\{O_1, O_2, \ldots, O_n\}$ be the set of objectives of a problem to achieve in an available resource T. Objectives O_i can be totally or partially ordered and organized in an acyclic graph where arcs represent the dependency between objectives. Let A be the set of actions $\{a_1, a_2, a_3, \ldots, a_n\}$ which act to achieve the objectives O_i. We assume that when an action a_i is executed, it contributes in achieving partially some objectives O_i. In a simple scenario, an action gradually achieves a unique objective and objectives are considered to be completely independent or weakly coupled. However, we focus our attention on actions that achieve more than one objective, at once, and that the objectives could be strongly connected, such in the example we presented above where the dependency between the length of the path and energy consumption for example is obvious. We present an approach that allows us to represent this problem as a multi-criteria decision. We use vector $\mathsf{Z} = (z_1, z_2, \ldots, z_n)$ of solution qualities of objectives O_i where z_i is the solution quality of achieving the objective O_i. In the context of our decision-theoretic planner, the objectives are to minimize the length of the path, the number of slopes of the path and the resource consumption.

In the following, we describe our formal framework and how to best act to solve this multi-objectives problem.

3. Multi-objective decision-theoretic planning with Markov Decision Process

3.1. Markov Decision Process

The Markov Decision Process (MDP) framework provides a formal description for modeling a large variety of stochastic, sequential decision problems. It is a well framework with well established on-line and off-

line algorithms for determining optimal behavior, such as value iteration algorithms and policy iteration algorithms [2]. The limitations of this developed framework are also well-known: compliance with Markov property, generally requires a very fine grained description of the environment, i.e. a very large number of states. Many researchers have focused their attention on the development of methods for large state spaces. However, a small attention has been paid to the reward and value functions where numerical and additive measures are not available. Indeed, several applications requires a multi-attribute value function. Before discussing the issues that should be addressed when dealing with a multi-attribute value function in MDPs, let us first recall that an MDP model is defined by a tuple $< S, A, p, r, T >$ s.t.: (1) S is a set of states, (2) A is a set of available actions, (3) p is the probability distributions where $p(s, a, s')$ describing the probability to be in state s' when applying action a in state s (this probability represents the uncertainty on the outcome of the action a, (5) rewards r(s) obtained when being in state s and (5) the horizon T is a set of stages (assumed in our case finite, but this assumption does not affect the claim of the paper) in which decisions are made. A policy δ is an application from S to A assigning an action to each possible state in the world. Solving an MDP consists in deriving an optimal policy δ^* by solving Bellman equation:

$$V^*(s) = r(s) + \max_{a \in A} \sum_{s'} p(s, a, s') V^*(s')$$

$$\delta^* = arg \max_{(\delta)} (r(s) + \sum_{s'} p(s, \delta(s), s') V^*(s')) \quad (1)$$

The optimal policy δ^* maximizes the expected value function V^* (expected value of the optimal policy) as described by Eq. (1). Deriving an optimal policy consists in maximizing the expected value in each state. This computation is based in using a max operator. If we extend the expected value function to a multiple attributes function, the global optimal policy becomes less clear. We discuss this problem in the rest of paper by defining different notions of optimality.

3.2. MDP with multi-attribute value function

The expected value of a non-terminal state in an MDP is given by the following equation:

$$V(s) = r(s) + \max_{a \in A} \sum_{s'} p(s, a, s') V(s') \quad (2)$$

This expected value function of terminal states s_T is:

$$V(s_T) = r(s_T) \quad (3)$$

The computation of state value is performed in a backward way starting from the terminal states. This computation leads to an optimal policy (solution) and assumes a single attribute value function of a state. Let us, now, assume the state s be a vector of criteria $\mathsf{Z} = (z_1, z_2, \ldots, z_n)$ where actions α modify the values of criteria. States s represent the state of the vector Z. Equations (2) and (3) should be adapted by redefining the reward value that itself can be seen as a vector of local reward functions s.t.: $r(\mathbf{Z}) = (r_1(z_1), r_2(z_2), \ldots, r_n(z_n))$ and the value function V as a vector of value functions s.t.: $V(\mathsf{Z}) = (v_1(z_1), v_2(z_2), \ldots, v_n(z_n))$. The function $r_i(z_i)$ is the reward of the value of a criterion z_i. The value function $v_i(z_i)$ is the expected gain of the value of a criterion z_i. Equation (2) adapted to our framework becomes:

$$V(\mathbf{Z}) = r(\mathbf{Z}) + \max_{a} \sum_{\mathbf{Z}'} p(\mathbf{Z}, a, \mathbf{Z}') V(\mathbf{Z}') \quad (4)$$

The value function v_i is the expected gain of the value of a criterion z_i given by: $v_i(z_i) = r_i(z_i) + \sum_{\mathbf{Z}'} p(\mathbf{Z}, a, \mathbf{Z}') v_i(z_i')$ where z_i is the state of criterion i of $\mathbf{Z}$ while z_i' is the state of criterion i of $\mathbf{Z}'$. By extension of Eq. (4), we obtain, then:

$$\begin{pmatrix} v_1(z_1) \\ v_2(z_2) \\ \ldots \\ v_n(z_n) \end{pmatrix} = \begin{pmatrix} r_1(z_1) \\ r_2(z_2) \\ \ldots \\ r_n(z_n) \end{pmatrix} + \max_a \sum_{\mathbf{Z}'} p(\mathbf{Z}, a, \mathbf{Z}'). \begin{pmatrix} v_1(z_1') \\ v_2(z_2') \\ \ldots \\ v_n(z_n') \end{pmatrix} \quad (5)$$

where $\mathbf{V}(\mathbf{Z}) = (v_1(z_1), v_2(z_2), \ldots, v_n(z_n))$ and $\mathbf{V}(\mathbf{Z}') = (v_1(z_1'), v_2(z_2'), \ldots, v_n(z_n'))$ is the value of the vector obtained after application of action a. To derive an optimal policy from this equation we need to redefine the $\max$ operator. To do that, we have to deal with different notions of optimal solutions using different notions of preferred solutions. In the same equation the probability $p(\mathbf{Z}, a, \mathbf{Z}')$ is a specific conditional probability table in dynamic Bayesian networks. The Bellman equation extended to a vector-valued function is similar to a linear system of Bellman equations. Thanks to Eq. (2), the optimal policy δ^* is subject to:

$$\left\{\begin{array}{l} v_1(z_1)=r_1(z_1)+\max_a \sum_{\mathbf{Z}'} p(\mathbf{Z},a,\mathbf{Z}')v_1(z_1') \\ v_2(z_2)=r_2(z_2)+\max_a \sum_{\mathbf{Z}'} p(\mathbf{Z},a,\mathbf{Z}')v_2(z_2') \\ \ldots \\ v_n(z_n)=r_n(z_n)+\max_a \sum_{\mathbf{Z}'} p(\mathbf{Z},a,\mathbf{Z}')v_n(z_n') \end{array}\right\}$$

Note that this system couldn't have a solution because the action optimizing criterion z_i could be different from the action optimizing criterion $z_{j\neq i}$. However, deriving a preferred solution from this linear system of equations is possible. We discuss in the rest of the paper, how to solve this system.

4. Bounded resources as a complementary objective

A multi-objective decision-theoretic planner offers new potential for real-world applications requiring solution quality as a function with multiple criteria. Such planners could be extended to deal with limited resources by considering the resource consumption minimization as an objective. The approach we present will also consider this aspect by considering a new state representation in the MDP: [**Z**,t] where **Z** is the vector of criteria and t is the resource consumed to attain vector Z. In the rest of the paper, let t be time consumed (assumption does not affect the generality of the approach). We describe an adaptation of the former of Eq. (4) to this new representation. To do that, we have to redefine the probability $p(\mathbf{Z},a,\mathbf{Z}')$ by considering uncertainty of resource consumption of actions α and we replace the reward function by a time-dependent utility. In the following, we give preliminaries on stochastic outcomes of actions and a new adaptation of the Bellman equation to our former and discussing the problem of solving the obtained MDP.

4.1. Representation of stochastic outcome of actions

The probability $p(\mathbf{Z},a,\mathbf{Z}')$ represents the stochastic outcomes of each action. Indeed, each individual action a has a characterization of its performance that maps the status of an input quality vector **Z** to a discrete probability distribution of the resource consumption c and output quality z_i that is the quality of objective O_i after the execution of an action a. This conditional probability, denoted $PP_{a,i}((z_i,c)|\mathbf{Z})$, expresses the dependency between the status of the vector and the resource consumption c and the output quality z_i of objective O_i. This probability allows us to indicate how an action contributes in improving the quality of an objective given the current state of the qualities of all the objectives of the problem. We use a discrete representation of this performance $\{\ ((z_i,c),\mathbf{Z},p)\ \}$ where p is the probability to get the couple (z_i,c) given vector Z.

The representation we use allows us to compact the joint probability distribution over the considered objectives and to represent this conditional probability compactly as in Bayesian networks. Our technique is inspired from the techniques we find in those representations [11,23,30]. Indeed, the objectives are organized in an acyclic graph where each node is an objective and the arcs are the dependency between them. With this representation, we use some well-known techniques used in Bayesian networks to define and estimate the joint probability distribution. Consequently, the probability to get a quality z_i of an objective depends on the qualities of its predecessors that we name also *parents* and we denote with $parent(i)$. This conditional probability is expressed as follows:

$$PP_{a,i}((z_i,t)|\mathbf{Z}) = Pr((z_i,t)|z_{j\in \textit{parent}(i)})$$

where Pr is the probability to get an output quality z_i and computation time t given the qualities z_j of the parents of the node i.

4.2. Dependency

Definition 1. We say that objective p depends on objective q when we have to achieve, even partially, objective q before achieving objective p. We also say that objective q preceeds objective p.

We formalize this definition by using $PP_{a,i}$ and generalizing to situations where objective p depends on objectives $r,s,\ldots,u$:

$$\mathbf{Z} = (z_1,\ldots,z_r,z_s,\ldots,z_u,\ldots,z_n)$$

$\forall\ t$ and **Z**, with $z_r=0$ or $z_s=0$ or $z_u=0$,

$$PP_{a,p}((z_p=0,t)|\mathbf{Z}) = 1 \quad (6)$$

and $\forall\ t$ and **Z**, with $z_r>0$ and $z_s>0$ and $z_u>0$,

$$PP_{a,p}((z_p>0,t)|\mathbf{Z}) > 0 \quad (7)$$

This equation means that the quality of the solution of objective p being equal to 0 when at least one objective on which p depends are not (even partially) solved because of Eq. (6).

4.3. Time-dependent utility function

Instead of a reward function $r(\mathbf{Z})$ where the resource dimension is not considered, we introduce an utility function $U(\mathbf{Z},t)$ of the output vector quality that represents the utility of the status of the quality vector after consuming t resource units. We represent this utility function $U(\mathsf{Z},t)$ as a vector of utilities functions of each dimension such that: $U(\mathbf{Z},t) = (u_1(z_1,t),u_2(z_2,t),\ldots,u_n(z_n,t))$.

4.4. Transitions

The transitions from a state [**Z**,t] to a state [**Z**′,t′] when acting with an action a is probabilistically given by the probability $Pr([\mathbf{Z}', t']|[\mathbf{Z}, t], \alpha)$. This probability is calculated by:

$$Pr([\mathbf{Z}', t' = t + c]|[\mathbf{Z}, t], \alpha) = \Pi_{i \in \{p,q,\ldots,r\}} PP_{a,i}((z_i, c)|\mathbf{Z})$$

where $\{p, q, \ldots, r\}$ are the objectives modified in $\mathbf{Z}$ after the execution of action a. Moreover, given that our agent is with limited resources T, we have to consider all transitions that lead to an over-consumption of resources ($t' > T$). For this reason, we consider those transitions by the following equation where the execution is stopped when the available resource has been fully elapsed and no modification in the vector $\mathbf{Z}$ has been done. Differently speaking, if an action leads to consumption of available resources then the current state is not changed, and all the probabilities have to be additively cumulated.

$$Pr([\mathbf{Z}, t' = T]|[\mathbf{Z}, t], a) = K. \sum_{c:c+t>T} PP_{a,i}((z_i, c)|\mathbf{Z})$$

Where K is a constant to normalize the sum.

4.5. Value function

We adapt the Bellman equation [2] to our formal framework such that:

The value of intermediate states

$$V^*([\mathbf{Z}, t]) = \max_{a \in A} Pr([\mathbf{Z}', t + c]|[\mathbf{Z}, t], a)V([\mathbf{Z}', t + c]) + Pr([\mathbf{Z}, T]|[\mathbf{Z}, t], a)V([\mathbf{Z}, T])) \quad (8)$$

The value of terminal state

$$V^*([\mathsf{Z}_{\mathsf{best}}, t]) = U(\mathsf{Z}_{\mathsf{best}}, t) \quad (9)$$

We consider for this equation domains where the computation of the best vector is possible such as path planning, navigation and exploration problems (In the path planning of our example, the best vector could be (d_{min}, e_{min}, $slope_{min}$) for length with minimal distance and with minimal consumed energy and minimal slope in the path. d_{min}, e_{min}, $slope_{min}$ are criteria that can be assessed by a reverse scale of the distance (the shorter the path the better it is) and of the consumed energy (the lower the energy consumption the better it is) and finally the best path is the one with minimal slopes. We also attain terminal states when all resources have been fully consumed:

$$V^*([\mathbf{Z}, T]) = U(\mathbf{Z}, T) \quad (10)$$

The resulting MDP is a finite-horizon that can be easily solved for a reasonable size (relatively large state spaces) using standard dynamic programming algorithms (value iteration algorithm) [2] or search algorithm AO^* (extension of A^* algorithm to AND/OR graph [21]) where the operator *max* is redefined for multi-value functions instead of a single value function. This is the issue we will discuss in the next sections.

5. Social welfare ordering and optimal policy

The multi-objective decision-theoretic planning leads to a problem of maximizing the satisfaction of each individual objective using a society's preferences. For this reason, we borrow the concepts from welfare economics [13]. Two concepts are possible: (1) using the approach maximizing the sum of all utilities of the member of society (utilitarianism concept), and (2) using the approach minimizing differences between the utilities of the member of society (egalitarianism concept). The first concept takes a sense in many application where all members contribute to an overall goal of the society. This is not the case in many multi-objective applications where objectives could potentially be conflictual. In the opposite, egalitarianism concept allows us to consider that differences of individual welfare are unjust and consequently to remove or attenuate these differences. In other words, the foremost goal of such a society is to maximize the welfare of its weakest member. We use this egalitarianism concept to deal with the multi-objective MDP. We introduce two *social welfare orderings* over multi-attributes value functions. Given the value of each individual objective (attribute of the value function), a social welfare ordering formalizes the notion of a society's preferences. In this section, we will examine the *maximin and lexicographic max-order* ordering and the relationship between the notion of optimal policy and preferred solutions using social welfare ordering [13]. In the following sections, we use definitions of general concepts of "Decision with Multiple Objectives: Preferences and Value Tradeoffs" introduced in [13] and gracefully formalized in [12]. The aim of the use of these concepts is to define a notion of optimality in MDP with vector value functions using preference ordering.

5.1. Elitist social welfare: Preference-based approaches

In this section, we examine different approaches of elitist preference-based techniques and explaining their suitability and sensitivity.

We define a preference between the different values of a criterion z_i. Let $\prec_{z_i} \subseteq D(z_i) \times D(z_i)$ be a strict partial order for each z_i. For example, we choose $>$ for smoothness and $<$ for the length of the path. We write $u \preceq v$ iff $u \prec v$ or $u = v$. We write $v_{\mathsf{S}}(z_i)$ the value of a criterion z_i in solution S.

Most of multi-criteria optimization methods consider preferences between the different values of each criterion but they do not specify that some criteria are more important than others (we would like to state a preference between a high smoothness and a short length) but we will still would like to get a solution where the length of the path is minimized while the degree of the smoothness is maximized. The main issue is to define different notions of optimal solutions with notions of preferred solutions.

Let us, now, express with preferences different notions of optimality used in multi-criteria optimization.

5.1.1. Pareto-optimal solutions

Definition 2. A solution S is Pareto-optimal solution of multi-criteria $(\mathbf{Z}, \prec_{\mathbf{z_i}})$ problem iff there is no solution S^* s.t. $v_{S^*}(z_k) \prec_{z_k} v_S(z_k)$ for a k and $v_{S^*}(z_i) \preceq_{z_i} v_S(z_i)$, for all i.

In our framework, Eq. (8) can be based on this definition to derive a Pareto-optimal solution using a preference on criteria. The adaptation of this definition to our framework consists in using the strict partial order $\prec_{z_i}$ for each z_i where the value $v_{\mathsf{S}}(z_i)$ of a criterion z_i in solution S corresponds to the value v_{z_i} of a criterion z_i in the vector $V(Z, t)$.

Claim 1. An optimal policy of an MDP given a multi-criteria value $V(\mathbf{Z}, t)$, a performance profile PP_a and a preference on criteria $\prec_{z_i}$, is the pareto-optimal solution of multi-criteria $(\mathbf{Z}, \prec_{\mathbf{z_i}})$ problem, at each state.

Proof: This claim is trivial since Eq. (8) selects the maximal value that can be redefined by a pareto-optimal solution. The Bellman equation with multi-attributes value is as follows:

$$V^* = \max_{\alpha} \{(v'_{\alpha,z_1}, v'_{\alpha,z_2}, \ldots, v'_{\alpha,z_n})\},$$

where

$$v'_{\alpha,z_i} = Pr([\mathbf{Z}', t+c]|[\mathbf{Z}, t], \alpha) v_{q'_i} + Pr([\mathbf{Z}, T]|[\mathbf{Z}, t], \alpha) v_{z_i}$$

Let Z_A be defined as follows:

$Z_A = \{(v'_{\alpha,z_1}, v'_{\alpha,z_2}, \ldots, v'_{\alpha,z_n}) \mid \forall\ \alpha \in A$, and $v'_{\alpha,z_i} \in D(\mathsf{z_i})\}$ is the set of vectors $(v'_{\alpha,z_1}, v'_{\alpha,z_2}, \ldots, v'_{\alpha,z_n})$. In our formal framework we consider that each criterion z_i has a domain $D(z_i)$. Since $v'_{\alpha,z_i} \in D(z_i)$ we can say that $Z_A \subset \mathbf{Z}$.

The permutation $\pi(Z)$ corresponds to the rearrangement respecting an increasing order. This permutation can be then applied to Z_A. Then, we can apply the preference on criteria $\prec_{z_i}$ to $\pi(Z_A)$. Consequently, the absolute solution of the multi-criteria $(Z_A, \prec_{z_i})$ with a permutation π is the optimal solution of the MDP.

We can thus use $\prec_{z_i}$ partial order to compare vectors in Z_A. Consequently, $\max_\alpha$ operator in the previous equation is defined here from the Pareto-optimal solution definition. In some situations, we cannot establish a preference order between criteria (for example vectors (5,3,6) and (4,4,5) could be difficult to compare) but we would like to be able to find compromises between them. The weighted sums are often used to achieve those compromises. Surprisingly, those methods do not necessarily produce the best solutions (the ones offering the best compromises).

Sometimes, it is natural to specify preferences between different criteria as well. For example, we prefer a smoothest path rather than a shortest one. We therefore introduce preferences between criteria in form of a strict partial order $\prec_{\mathbf{Z}} \subseteq \mathbf{Z} \times \mathbf{Z}$.

We aggregate preferences between and on criteria to preferences between assignments of criteria of the form $z_i = v$. Let $\prec$ be the relation satisfying following two conditions: (1) if $u \prec_{z_i} v$ then $z_i = u \prec z_i = v$ and (2) if $z_i \prec_{\mathbf{Z}} z_j$ then $z_i = u \prec z_j = v$, $\forall u, v$. The second condition means that if criterion z_i is preferred to criterion z_j then any assignment to z_i is preferred to any assignment to z_j.

5.1.2. Ranking of criteria

A ranking of criteria is considered to derive a lexicographic optimal solution. This interpretation leads to the R-preferred solution [12].

Definition 3. A solution S^* is a R-preferred solution of $(\mathbf{Z}, \prec)$ if there exists a permutation π s.t. (1) π respects $\prec_{\mathbf{Z}}$ (i.e. $z_i \prec_{\mathbf{Z}} z_j$ implies $\pi_i < \pi_j$) and (2) there is no other solution S satisfying $V_{\mathsf{S}}(\pi(\mathbf{Z})) \prec_{\mathbf{lex}} \mathbf{V}_{\mathsf{S}^*}(\pi(\mathbf{Z}))$

Intuitively, R-preference compares solutions according to an ordering of criteria. Let consider the example of preferences on car. R-preference can express the fact that cars are compared first according to the price criterion and second according to the color criterion.

Claim 2. An optimal policy of an MDP given a multi-criteria value function with a $\prec$ preference between criteria, a performance profile PP_α and a permutation π is the R-preferred solution of (**Z**, $\prec$), at each state

Proof: The proof is similar to the previous ones using Z_A.

5.1.3. Elitist social welfare

An interesting case in the social welfare that we examine is where the welfare of the society is evaluated on the basis of the happiest member. In such elitist society, the decision process supports the champion (the happiest member). In our formal framework, we represent the multi-dimensional utility by the utility of the maximal quality of the vector.

Definition 4. An *optimistic utility* associated with a state [**Z**,t] can be defined as:

$$u^{Opt}(\mathbf{Z},t) = \max_{i\in\{1,2,\ldots,n\}}(u_1(z_1,t),u_2(z_2,t),\ldots,u_n(z_n,t)) \quad (11)$$

The policy δ_{max} using Eq. (8) based on u^{Opt} means that we maximize the utility of the most satisfied criterion. This approach focuses its processing in improving the criterion of the highest quality. This approach has a little sense but it could be appropriate for some situations. A typical scenario is where different agents are launched with the same goal, with the aim that at least one agent achieves that goal (no matter what happens to the others). We don't consider this approach for our problem because it means in our case that only one criterion is considered for optimization while we are interested in solutions with the best balance between all criteria.

5.2. Egalitarian social welfare approaches

In this section we consider each criterion of the multi-dimensional value function as a member of society criteria and we show how social welfare ordering contributes in defining measures that allow us to make expectation. Those measures allow us to compare between the utilities of two quality vectors (seen here as a society) and try to guide the decision process using Eqs (7), (8) and (9). In the following, we give two social welfare ordering over multi-dimensional value: the maximin and lexicographic order.

5.2.1. Egalitarian social welfare

The aim of an egalitarian social welfare is to maximize the welfare of its weakest member. In that sense, we can measure the social welfare by measuring the welfare of the member who is worst off. In our formal framework, this idea leads to the definition of the following egalitarian social welfare function

Definition 5. The multi-dimensional utility is a vector of single utility of each dimension (u_i). An *egalitarian social welfare* associated with a state [**Z**,t] can be defined as:

$$u^{sw}(\mathbf{Z},t) = \min_i(U(\mathbf{Z},t)) = \min_{i\in 1,\ldots,n}(u_1(z_1,t),u_2(z_2,t),\ldots,u_n(z_n,t)) \quad (12)$$

The function $u^{sw}(\mathbf{Z},t)$ gives rise to a social preference ordering over different quality vector states. State [**Z**,t] is strictly preferred over state [**Z′**,t] iff $u^{sw}(\mathbf{Z},t) > u^{sw}(\mathbf{Z}',t)$. This ordering is well-known by the *maximin*-ordering name. The computed policy δ_{sw} using u^{sw} allows us to maximize the egalitarian social welfare. This policy is based on a sequence of actions that maximizes the welfare of the weakest criterion.

5.2.2. Lexicographic optimal solutions

The *maximin*-ordering induced by u^{sw} only takes into account the welfare of the currently weakest component, but is insensitive to utility fluctuation in the rest of the vector. To allow for a finer distinction of the social welfare for different quality vectors, we use the so-called *leximin*-ordering. The principle consists in rearranging the elements of the vector in increasing order and using a lexicographic ordering over the rearranged vectors.

Definition 6. Let $\mathcal{V}_{\mathsf{S}}(\pi(\mathbf{Z})) = (v_{\mathsf{S}}(z_{\pi_1}), v_{\mathsf{S}}(z_{\pi_2}), \ldots, v_{\mathsf{S}}(z_{\pi_n}))$. A solution S^* is an absolute solution of (**Z**, $\prec_{z_i}$) iff there is no other solution S s.t. $\mathcal{V}_{\mathsf{S}}(\pi(\mathbf{Z})) \prec_{lex} \mathcal{V}_{\mathsf{S}^*}(\pi(\mathbf{Z}))$ and $\prec_{lex}$ is a lexicographic order.

We mean by an absolute solution that some criteria have an absolute priority over other criteria. Different ranking (or permutation) lead to different absolute solutions which are all Pareto-optimal. An absolute solution can be determined by solving a sequence of single-criteria optimization problems starting with the most important criterion.

The lexicographic order in our framework considers the multi-attribute value function $\mathcal{V}(\pi(\mathbf{Z}),t) = (v_{z_{\pi_1}}, v_{z_{\pi_2}}, \ldots, v_{z_{\pi_n}})$.

Claim 3. An optimal policy of an MDP given a multi-criteria value function $\mathcal{V}(\pi(\mathbf{Z}), t)$, a performance profile PP_α, π a permutation and a lexicographic order $\prec_{lex}$, is the absolute solution of multi-criteria $(\mathbf{Z}, \prec_{\mathbf{z_i}})$ problem with the permutation π and a lexicographic order $\prec_{lex}$ s.t. $\mathcal{V}_{\mathsf{S}^*}(\pi(\mathbf{Z}), t) \prec_{lex} \mathcal{V}_{\mathsf{S}}(\pi(\mathbf{Z}), t)$, at each state.

Proof: In Eq. (8), we have:

$$\mathcal{V}^* = \max_{\alpha}\{(v'_{\alpha,z_1}, v'_{\alpha,z_2}, \ldots, v'_{\alpha,z_n})\},$$

where

$$\begin{aligned} v'_{\alpha,z_i} &= Pr([\mathbf{Z}', t+c]|[\mathbf{Z}, t], \alpha) v_{q'_i} \\ &\quad + Pr([\mathbf{Z}, T]|[\mathbf{Z}, t], \alpha) v_{z_i} \end{aligned}$$

$$\begin{aligned} Z_A = \{(v'_{\alpha,z_1}, v'_{\alpha,z_2}, \ldots, v'_{\alpha,z_n}) \mid \forall\, \alpha \in A, \\ \text{and } v'_{\alpha,z_i} \in D(\mathsf{z_i})\} \end{aligned}$$

is the set of vectors $(v'_{\alpha,z_1}, v'_{\alpha,z_2}, \ldots, v'_{\alpha,z_n})$. In our formal framework we consider that each criterion z_i has a domain $D(z_i)$. Since $v'_{\alpha,z_i} \in D(z_i)$ we can say that $Z_A \subset \mathbf{Z}$. The permutation $\pi(Z)$ can be then applied to Z_A. Then, we can apply the lexicographic order $\prec_{lex}$ to $\pi(Z_A)$. Consequently, the absolute solution of the multi-criteria $(Z_A, \prec_{z_i})$ with a permutation π and a lexicographic order is the optimal solution of the MDP.

6. Value iteration algorithms with social welfare ordering

The MDP described in Section 5 considering limited resources can be solved using a standard dynamic programming algorithms (value iteration algorithm). This algorithm is based on backward chain starting from the value of terminal states given by Eqs (9) and (10) and computing the values of the initial state and the intermediate ones using Eq. (8). To perform this processing, we need the initial values for starting computation Eqs (9) and (10) and the max operator for the computation of the values of the other states. To do that, we have to give for each social welfare ordering [13] what are the value of terminal states and which max operator we use.

- *Egalitarian Social Welfare*: in this approach, we have used u^{sw} measure given in Eq. (11) to initialize the values of terminal states given by Eqs (9) and (10). After that, we use the standard max operator for the value iteration. While for the lexicographic approach, max is given by $V^* = \max(V^*, V)$ iff $V \prec_{lex} V^*$.
- *Preference-based approaches:* in all those approaches, we have just to adapt the definition of the max operator to the definition of the preference introduced in [12,13] we use: (1) Pareto-optimal definition we use: $V^* = \max(V^*, V)$ iff $\forall z_{k\neq i} V.v(z_i) \prec_{z_i} V^*.v^*(z_i)$, (2) Ranking criteria, we use: $V^* = \max(V^*, V)$ iff $V \prec_{lex} V^*$ with a particular permutation π defined by: $\pi(z_i) < \pi(z_j)$ iff $z_i \prec_{\mathsf{Z}} z_j$. Finally, we use standard max operator for the elitist social welfare using u^{opt} by initializing the value of terminal states given by Eqs (9) and (10) using the measure u^{opt}.

7. A socially satisfying policy for 2VMDP

7.1. Basic idea

The overall objective of our approach is to allow the decision making of an artificial agent to evolve actions that have multi-dimensional outcomes. Our approach uses a multi-dimensional value function as a vector of value functions of criteria and then it can use one of welfare social ordering to prefer an action over another one. But, *Maximin ordering* and *Leximin ordering* are interesting when the values of criteria have the same scale or the same nature. In this paper, we consider an approach using *Leximin ordering* and measures similar to *Maximin regret* and *Competitive ratio* to make leximin independent from scale factor. Furthermore, these measures will give more global sense among all criteria to the values of states. Indeed, our approach needs to know the distance between the current value of a criterion and its optimal value that it is computed during an optimization step. To do that, the first step of our approach is to derive all policies $\{\ \delta_i\colon\ i \in \{1 \ldots n\}\ \}$ that optimize one criterion independently from the others and the second step constructs a new policy using the values of the policies δ_i, $\mathbf{V}^{\delta}\mathbf{i}(Z_0)$. Indeed, we compare the value of each outcome $\mathbf{V(Z)} = (v_1(z_1), v_2(z_2), \ldots, v_n(z_n))$ using vector $V^{\delta_1,\delta_2,\ldots,\delta_n} = (v_1^{*,\delta_1}, v_2^{*,\delta_2}, \ldots, v_n^{*,\delta_n})$ where v_i^{*,δ_i} is the expected value of the policy δ_i optimizing criterion i (Fig. 1). Note that vector $V^{\delta_1,\delta_2,\ldots,\delta_n}$ can be realistically not reachable but it allows us to use it as a value of an ideal state that a decision making agent tries to reach. In the example presented in Fig. 1, vector (17, 11, 9) is out of reach. The decision making agent will prefer an action a over an action b when the value of the outcome of action a is *closer to* $V^{\delta_1,\delta_2,\ldots,\delta_n}$ than the value of the outcome of action b. The relation *closer to*

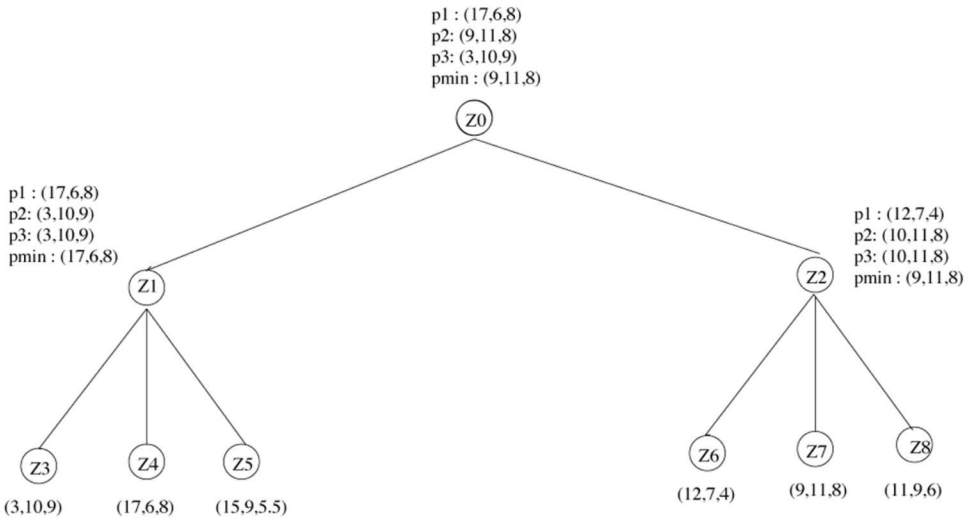

Fig. 1. The computation of local policies δ_i.

will be defined using qualitative decision criteria like *maximin, minmax regret, leximin order and competitive ratio*. We also discuss the Euclidean distance decision criterion and its unsuitability.

7.2. Decision making with vector-valued functions

7.2.1. Computation of policies δ_i

This step is necessary to compute vector $V^{\delta_1,\delta_2,\ldots,\delta_k}$ where the value of each feature i of this vector is a result of a policy δ_i. Let's define now how policy δ_i prefers an action over another.

Definition 7. Let **V(Z)** and **V(Z')** be respectively the values of the outcomes of actions a and b. The decision making agent prefers action a over b when $\mathbf{v_i}(z_i) > \mathbf{v_i}(z'_i)$. Let this ordering be $\max^i$.

The policy δ_i is computed by using the $\max^i$ *ordering* in the linear system of equations Eq. (2). It allows us to maximize the value of criterion i. This policy is given by solving the following linear system:

$$\mathbf{V}(\mathbf{Z}) = \mathbf{r}(\mathbf{Z}) + \max_{a\in\mathcal{A}}^{i} p(\mathbf{Z}, a, \mathbf{Z}').\mathbf{V}(\mathbf{Z}') \qquad (13)$$

$$\delta_i^* = arg\max_{a\in A}^{i} \mathbf{V}(\mathbf{Z})$$

Let v_i^{*,δ_i} be the value of the optimal policy optimizing criterion i, given in Eq. (13). The values of the other criteria $j \neq i$ are given as follows:

$$v_j(z_j) = r_j(z_j, \delta_i(\mathbf{Z})) + \sum_{\mathbf{Z}'} P(\mathbf{Z}', \delta_i(\mathbf{Z}), \mathbf{Z}).v_j(z'_j)$$

This processing is performed for all criteria $i \in \{1, \ldots, n\}$. Thus, $V^{\delta_1,\delta_2,\ldots,\delta_n} = (v_1^{*,\delta_1}, v_2^{*,\delta_2}, \ldots, v_n^{*,\delta_n})$.

7.2.2. Computation of a socially satisfying policy

In order to define the relation *close to*, we use qualitative decision criteria like *maximin, minmax regret, leximin order and competitive ratio* that have interesting properties such *closure under union*.

Definition 8. Let $\mathbf{V(Z)} = (v_{z_1}, v_{z_2}, \ldots, v_{z_n})$ and $\mathbf{V(Z')} = (v_{z'_1}, v_{z'_2}, \ldots, v_{z'_n})$ are respectively the values of the outcomes of actions a and b. The decision making agent prefers action a over b based upon ratio regret criterion adapted to our former if vector $(\frac{v_{z_1}}{v_1^{*,\delta_1}}, \frac{v_{z_2}}{v_2^{*,\delta_2}}, \ldots, \frac{v_{z_n}}{v_n^{*,\delta_n}})$ is lexicographically preferred over $(\frac{v_{z'_1}}{v_1^{*,\delta_1}}, \frac{v_{z'_2}}{v_2^{*,\delta_2}}, \ldots, \frac{v_{z'_n}}{v_n^{*,\delta_n}})$. We call this ordering minimax regret ordering.

These qualitative decision criteria should be analyzed in sense that our objective is to satisfy as high as possible all the criteria even if some criteria are contradictory. We have also to make our measure independent from the scale factors. Local values v_i can have different scales or different natures and the use of qualitative decision criteria like *maximin, minmax regret, leximin order* cannot be significant. That's why, we use another measure that combines lexicographic order with a normalized regret measure. This ordering is called competitive ratio ordering. This measure allows us to define a distance to the value of the ideal state $V^{\delta_1,\delta_2,\ldots,\delta_k}$. For each value of a state **V(Z)**, we define a new value $\mathbf{V^u(Z)}$ as follows: $\mathbf{V^u(Z)} = (v^u_{z_1}, v^u_{z_2}, \ldots, v^u_{z_n})$ where: $v^u_{z_i} = \frac{\left|v_i^{*,\delta_i} - v_{z_i}\right|}{v_i^{*,\delta_i}}$ and $v^u_{z_i} = 0$ for $v_i^{*,\delta_i}(z_i^0) = 0$.

$$v^u_{z_1} = \frac{\left|v_1^{*,\delta_1} - v_{z_1}\right|}{v_1^{*,\delta_1}}, v^u_{z_2} = \frac{\left|v_2^{*,\delta_2} - v_{z_2}\right|}{v_2^{*,\delta_2}}, \ldots,$$

$$v^u_{z_n} = \frac{\left|v_n^{*,\delta_n}(z_n^0) - v_{z_n}\right|}{v_n^{*,\delta_n}}$$

In the same way, we define a new reward function of all terminal states $\mathbf{Z}_T$ as follows: $\mathbf{r^u}(Z_T) = (r^u_{z_1}, r^u_{z_2}, \ldots, r^u_{z_n})$ where $r^u_{z_i} = \frac{\left|v_i^{*,\delta_i} - r_{z_i}\right|}{v_i^{*,\delta_i}}$

$$\mathbf{r^u}(Z_T) = \left(\frac{\left|v_1^{*,\delta_1} - r_{z_1}\right|}{v_1^{*,\delta_1}}, \frac{\left|v_2^{*,\delta_2} - r_{z_2}\right|}{v_2^{*,\delta_2}}, \frac{\left|v_n^{*,\delta_n} - r_{z_n}\right|}{v_n^{*,\delta_n}} \right)$$

This reward $r^u(Z_T)$ is necessary to compute the policy as shown bellow.

This new value vector $\mathbf{V^u}(\mathbf{Z})$ and reward $\mathbf{r^u}(Z_T)$ is used by the decision making agent to prefer an action over another one.

Definition 9. Let **V(Z)** and **V(Z')** be respectively the values of the outcomes of actions a and b. The decision making agent prefers action a over b when $\mathbf{V^u}(\mathbf{Z})$ is lexicographically preferred over $\mathbf{V^u}(\mathbf{Z}')$. We name this ordering competitive ratio and we denote it $\min^{lex,u}$.

The policy δ^* is computed using the $\min^{lex,u}_{a\in\mathcal{A}}$ *ordering* and allows us to satisfy as high as possible all the criteria by reducing the distance between the value of each criterion i and its best expected value. Consequently, the obtained policy is socially satisfying. By socially satisfying, we mean that this approach cannot prefer a vector over another when the distance of a criterion is too high even if the other criteria are very satisfying. This policy is given by solving the Bellman equation of our context:

$$\mathbf{V}(\mathbf{Z}) = \mathbf{r}(\mathbf{Z}) + \min^{lex,u}_{a\in\mathcal{A}} p(\mathbf{Z}, a, \mathbf{Z}').\mathbf{V}(\mathbf{Z}') \qquad (14)$$

$$\delta^* = arg \min^{lex,u}_{a\in\mathcal{A}} \mathbf{V}(\mathbf{Z})$$

Example 1. Figure 1 illustrates our algorithm in an example with 6 terminal states $\mathbf{Z}_3, \mathbf{Z}_4, \mathbf{Z}_5, \mathbf{Z}_6, \mathbf{Z}_7, \mathbf{Z}_8$, each of which a reward with three criteria is assigned. For simplicity of the presentation, we assume, in the example, that $\mathbf{r}(\mathbf{Z}) = (0, 0, 0)$ except for terminal states and that actions are deterministic as shown in Fig. 1.

We compute, first, policies δ_i noted in the example p_1, p_2, p_3 that optimize respectively criteria 1, 2 and 3. We also compute a policy $\delta_{minimax}$ in order to show the difference with our approach. From the policies p_1, p_2, p_3, we have $\mathbf{V^{p_i}}(\mathbf{Z_0})$ which are used to compute $\mathbf{V^{p_1,p_2,p_3}} = (17, 11, 9)$. This vector is used to compute vectors $\mathbf{V^u}(Z_3), \mathbf{V^u}(Z_4), \mathbf{V^u}(Z_5), \mathbf{V^u}(Z_6), \mathbf{V^u}(Z_7), \mathbf{V^u}(Z_8)$ that will be used with our technique as mentioned in Fig. 2 using Eq. (4) that leads to a more egalitarian approach. We can see that policy p_1 prefers $\mathbf{Z_4}$ over $\mathbf{Z_5}$ because the value of its first criterion is higher ($17 > 15$). Policy p_2 prefers vector $\mathbf{Z_3}$ over $\mathbf{Z_5}$ because the value of its second criterion is higher ($10 > 9$). Policy p_3 prefers $\mathbf{Z_3}$ over $\mathbf{Z_5}$ because the value of its third criterion is higher ($9 > 5.5$).

As we can see the expected value of the initial state following the policy p_{min} is (9,11,8) while the expected value of the initial state following the policy of our approach is (15,9,5.5). This result shows that p_{min} is a local optimization while our approach uses values with more global sense. Our approach can be seen as a *maximin* with more complete and global information among all criteria. Our approach considers that vector (15,9,5.5) is closer to (17,11,9) rather than (9,11,8) because of the value of criterion 1 that it is not satisfying and the distance to the best value 17 is high. Our approach is socially satisfying in the sense we give above that all the values of criteria should be as close as possible to the best possible values.

Theorem 1. The policy δ using $\min^{lex,u}$ is Pareto optimal.

Proof. Using Claims 1 and 2 and the fact that leximin ordering leads to a Pareto optimal solution and that $\min^{lex,u}$ uses this ordering then the policy obtained is Pareto optimal. □

7.2.3. *Value iteration algorithms with social welfare ordering*

From Eq. (5), we can derive a satisfying policy by using an optimization operator like $\min_{lex,u}$ that leads to Eq. (14). Solving this equation uses the following algorithm:

1. For all criteria z_i, compute the policy δ_i optimizing criterion z_i and get its optimal value v_i^{*,δ_i}.
2. For each terminal state $\mathbf{Z}_T$, compute $\mathbf{r}^u(\mathbf{Z}_T)$.
3. By backward chain from terminal states to initial state, compute for each state $\mathbf{Z}$ its regret ratio $\mathbf{V}^u(\mathbf{Z})$, using Eq. (14).
4. Derive the social satisfying policy $\delta^* = arg \min^{lex,u}_{a\in\mathcal{A}} \mathbf{V^u}(\mathbf{Z})$

This algorithm is illustrated in the example of Fig. 2. For example in this Figure we have to determine the preferred outcome from the different computed val-

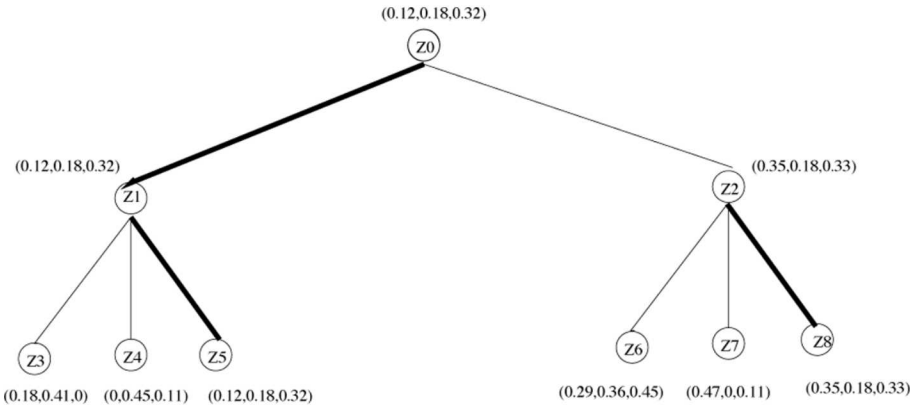

Fig. 2. The computed policy using $\mathbf{V}^u$.

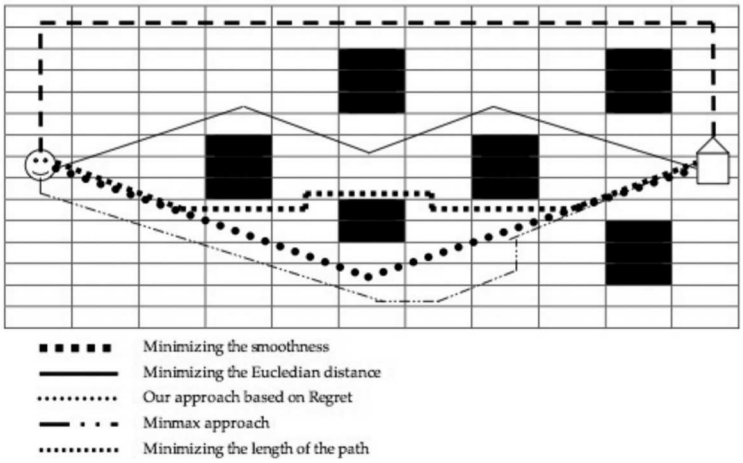

Fig. 3. Example: Multi-Criteria path planning.

ues V^u (0.18,0.41,0), (0, 0.45, 0.11) and (0.12, 0.18, 0.32). These vectors are ordered in a decreasing way that leads to vectors (0.41,0.18,0), (0.45,0.11,0), (0.32,0.18,0.12), then we apply a minmax criterion that allows us to select vector (0.32,0.18,0.12) because its $\max$ is the $\min$ of all the $\max$ of the other vectors. Then this vector becomes the value of the parent node. This algorithm has used the policies δ_i using Eq. (3) that compares two vectors by comparing only the values of criterion i. For example, in Fig. 1, policy $p1$ prefers vector (17,6,8) over all because the value of criterion 1 is dominating. This algorithm is similar to general-sum stochastic game but it differs from the fact that we use a regret measure.

Theorem 2. Our algorithm to solve 2VMDP with finite horizon with no loop is polynomial.

Proof. Let $|S|$ be the size of state space. The first step uses a classic backward chaining algorithm n times to compute all δ_i. Thus, this step is $O(n.|S|)$. The second step to compute the policy δ uses the backward chaining algorithm based on a leximin ordering. Thus, the algorithm is $O((n+1).|S|)$. The algorithm is then polynomial. □

8. Analysis and experimental results

We have tested our approach in the path planning in a grid problem depicted in Fig. 3. A robot is at a start location and should move toward the destination (charging room). There are many obstacles (rectangles in the figure). The actions of a robot are the 8 directions that we assume in this example as deterministic. However, we consider a multi-dimensional reward function using the length of the path (Manhattan distance) and the number of slopes of the path (simplified to the number of time the robot changes the direction). We con-

Table 1
Results of different policies

Policy	Expected smoothness	Expected length
δ_{length}	7	13.6
δ_{smooth}	2	24
$\delta_{minimax}$	6	16.6
$\delta_{lex,u}$	2	15.41
$\delta_{distance}$	5	14.1

sidered only two dimensions of the function but more dimensions doesn't affect the performance of the algorithm according to the statement of Theorem 2. The robot can create a path by any of the aforementioned multi-criteria planning technique. The paths presented in Fig. 3 are created by using *minimax* qualitative decision approaches, or a local optimization of a criterion. The obtained paths with these different approaches are summarized in Table 1. The measures reported in Table 1 are based on the fact that each cell has a dimension of one unit and crossing diagonally the cell measures $\sqrt{2}$. This table allows us to compare our approach with the single-criterion optimization approaches.

What we can interpret from these results is: the shortest path (length = 13.6) given by the policy δ_{length} has a poor value of the smoothness (7) while the smoothest path (2) is given by the policy δ_{smooth} has a poor value of the length (24). However, these policies allow us to have a value of an ideal path (the shortest and smoothest) that it is (2,13.6). Although the fact that there is no path having this value, the robot tries to find a path with a value closer to this ideal value. This is what we find with our approach which creates a path with value (2,15.41) that is close to (2,13.6). We can also see in Table 1 that the *minimax* qualitative decision approach is not enough accurate in general because the values of criteria are with local sense. In order to consider an approach based on weigthed-sum for comparison, we have been interested in an approach using an Euclidean distance as a decision criterion where an action a is preferred over b when the distance between the value of the outcomes of a and $\mathbf{V}^{\delta_1,\delta_2,\ldots,\delta_n}$ is smaller than the distance between the value of the outcomes of b and $\mathbf{V}^{\delta_1,\delta_2,\ldots,\delta_n}$. Formally speaking,

$$\delta_{distance} = arg\min_{\mathbf{Z}} \left(\sum_i (v_i(z_i) - v_i^u(z_i^0))^2 \right)^{\frac{1}{2}}$$

This criterion is a specific case of weigthed-sum approaches. The obtained policy creates a path with an expected value of (5,14.1). Here also, we can say that this decision criterion is not enough accurate. It means that this policy can prefer an action over another using

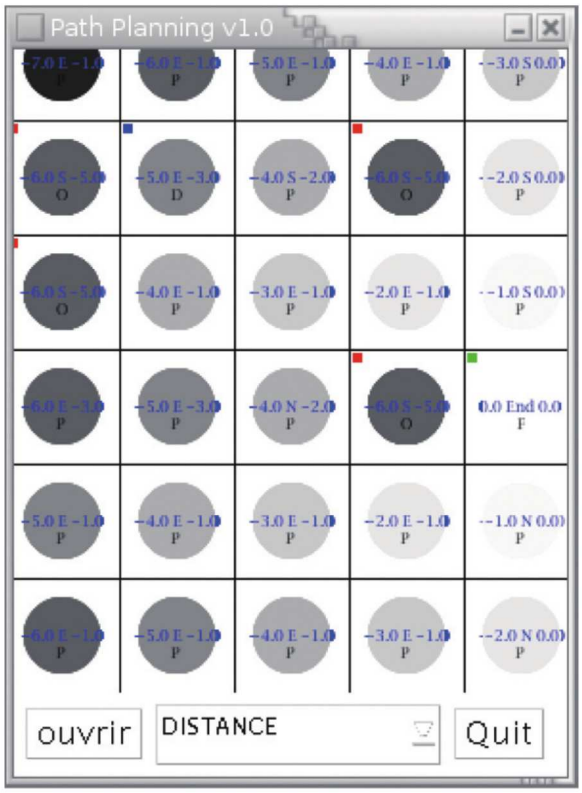

Fig. 4. Policy optimizing the length criterion.

the distance but it is insensitive to the fluctuation. Also it is known that an Euclidean distance is an approximation of the *closer to* relation because it does not take into account the fact that some locations are not realistically accessible. Our approach prefers (2,15.41) over (5,14.1) because criterion 1 is not satisfying and the distance to the best value 2 is high. Also, our approach prefers (2,15.41) over (6,16,6) because of the same reasons.

In addition to synthetic evaluation, we have implemented the algorithms and we assessed them on a grid 5×6. In Figs 4, 5 and 6, each cell in the grid contains the expected value of the length, the expected number of slopes and the action selected by the policy. Also, the cells are labeled D for the departure cell, F for the arrival cell, O for obstacles and P for the others. Also, the color of the cell represents the degree of preference of the policy to reach this cell. Indeed, more the cell color is dark less it is preferred. The results depicted in Figs 4, 5 and 6 representing respectively the policies optimizing of length criterion, number of slope criterion and a leximin order over both criteria. It is shown that our policy has a value $(-5, -2)$ that is a good compromise in comparison with the short length policy

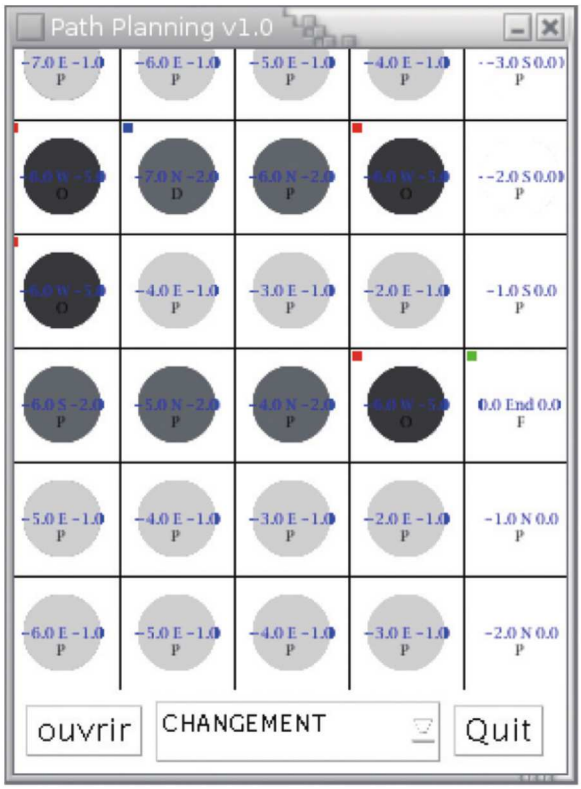

Fig. 5. Policy optimizing the number of slopes criterion.

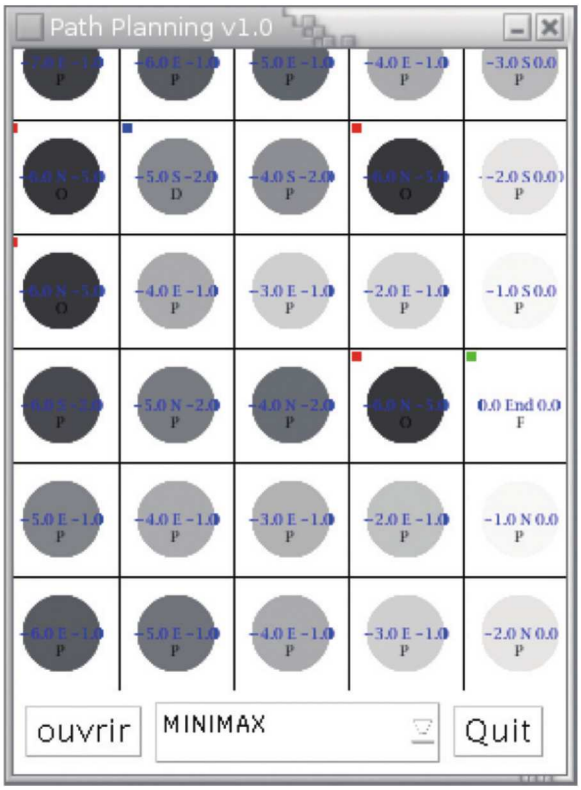

Fig. 6. Socially satisfying policy.

$(-5, -3)$ with a bad value of number of slopes and the number of slopes policy $(-7, -2)$ that has a bad value for the length criterion.

In this example we presented results with only two criteria but it can be extended to more than 2 criteria since the computational complexity is polynomial as stated in Theorem 2.

Other experiments have been conducted with stochastic actions using scenarios as depicted in Fig. 3. We considered a grid representing an environment with obstacles (black boxes) and risky areas ("red" triangles) where a robot has to evolve using one of the policies introduced in previous sections. Actions are stochastic in sense that each action reaches the target cell with a probability 0.8 but it can reach the cells on the left or on the right of the target cell with respectively a probability 0.1. In Fig. 7 (grid 7×7) we can see different paths proposed with different policies. We can see that our policy (black path) proposes a path with a good balance between the risk (to collide with obstacles) and the length of the path, while the other policies optimize only one criterion (short-dash line path) optimizes the risk of collision that's why the path is far from obstacles and risky areas and the long-dash path optimizes the path by minimizing the risk (minimize the risk of collision first and the length after). An interesting observation that we can do from these experiments is the fact that the intuitive best path is that the robot goes to the left and then up. This intuition comes from the fact that we reason with deterministic actions but when introducing the stochastic aspect, the robot cumulates both risks of the left and the right ("red" triangles and thus crossing between both risky areas is with low value while deciding to go to one of side of risky areas leads to consider only one risky area and thus minimizes the risk. This is what it was performed with our approach in the black path.

Table 2
Computation time of the policy according to the size of the grid

Grid	Computation time of the policy (ms)
5×5	0.32
7×7	0.33
10×10	0.37
20×200.48	
50×50	1.01

This experiments confirm the expected behavior as presented in Fig. 3. We have also developed experiments with different size of the grid from 5×5 to $50 \times$

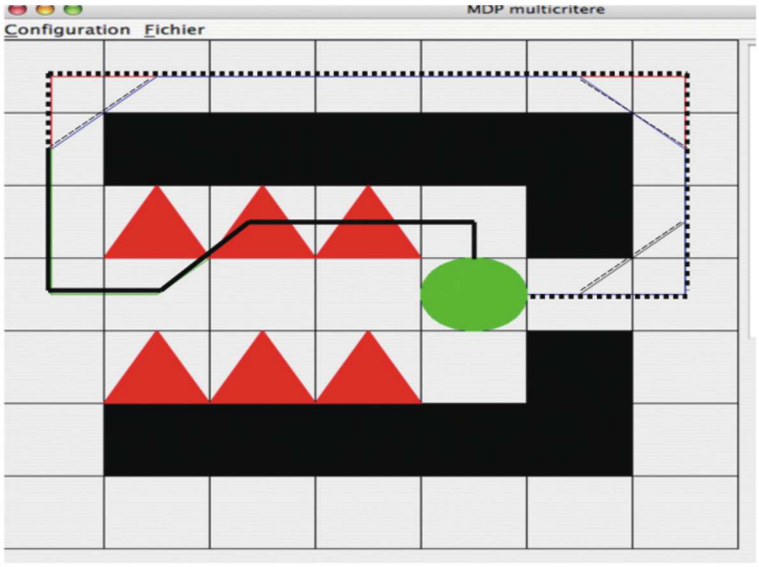

Fig. 7. Policies with stochastic actions.

50 and the time of computing the policy is summarized in Table 2. Experiments have been run on an Intel Core Duo 3.00 GHz CPU proces- sor with 1 GB of RAM. In this table is shown that for grids with sizes less than 50×50 we need less than 1 ms for computing the policy and around 1 ms for a grid with 50×50.

9. Related work

9.1. Decision-theoretic exploration algorithms using (PO)MDP

The exploration model is fully subsumed by the POMDP (Partially Observable Markov Decision Process) framework and applied in many domain such as planetary exploration [9,22,26], search and rescue [20], abandoned mine mapping [16,27] and sensor fusion [29]. However, exploring using POMDP meets the curve of dimensionality and limits its application to exploration domain because it considers uncertainty on observation, outcomes of actions an its state. Recent researches have been focused on algorithms that scale up. The most popular algorithms are QMDP which transforms a POMDP into an MDP of beliefs which has the same complexity of an MDP [15]. Beliefs represent the distribution probability on states and their set is huge. To reduce this set, a particle filter version of the POMDP algorithms approximate beliefs using particles. These class of techniques are named Monte-Carlo POMDP [32]. Other techniques based on specific structure or augmented MDP allow us to reduce this complexity. Recently, we have produced a new algorithm, named TOP [8], which is one of the most competitive algorithm based on topological organization of the space to better organize the resolution. This algorithm is very suitable to allow us to incrementally explore and map the environment. 2V-MDP is a new decision model to extend these techniques to multi-criteria issues.

9.2. Multi-Objective optimization

This work concerns Multi-Objective Linear Programming (MOLP) [1,7] which generalizes the standard Multi-criteria Optimization Problem (MOP) [10]. In the standard MOP, the problem remains the search of the Nadir point while the computation of the Ideal point is easy. Vector-Value Markov Decision Process (2V-MDP) contributes in the resolution of this problem by giving an approximate Nadir point while the Ideal point is computed from standard dynamic programming method. 2V-MDP is similar to the method presented in [7] which finds a solution without any additional computation. In [7], the authors use a MINMAX and MINSUM of percentage of deviations to find an optimal solution while in 2V-MDP we use a lexicographic order on percentage of deviations which lead to the same set of solutions when the vectors have no identical components. The theoretical solution on the weak pareto-optimality of the solution obtained by 2V-MDP is compatible with the claim of the authors in [7] considering the dominance order.

9.3. Non-classical and qualitative MDPs

This approach is in the spirit of many existing models of MDPs with vector value functions [24,33] and appropriate algorithms to solve them where most of them use backward induction, policy iteration and value iteration by substituting operations $(+, \times)$ by $(\max, \min)$ in computations. Other approaches have been interested in the use of a qualitative version of MDPs and algebraic MDPs [24,28]. Besides these positive results, we propose an alternative to standard MDPs combining regret measure similar to Tchebychev norm with an appropriate lexicographic order and a backward induction algorithm to derive a satisfying policy. Further comparisons with these non-classical MDPs model will be developed in the future work. Another contribution of our model is the use of these non-classical models of MDP for multi-agent planning coordination problem.

10. Conclusion and future works

In this paper, we have addressed the problem of stochastic path planning where more than one criterion is considered to prefer a path over another one. To do that, we propose an MDP with multi-attribute value function to design a multi-objective decision-theoretic planner. This approach meets the problem of deriving an optimal policy using a multi-dimensional value function. For this reason, we have presented different approaches using social welfare techniques and preferences to deal with an appropriate maximization operation of a multi-dimensional expected value. Consequently, we redefine max operator by different definitions of preferences in the Bellman equation for deriving an optimal policy. We present a new operator based on a regret ratio measure to derive a policy for the path planning offering a good trade-off between all the criteria of the path without degrading anyone.

Future work will concern the extension of this work to multi-agent stochastic path planning using multi-agent MDP [3,19] and the application of this approach in a real autonomous robot by considering its multiple resources such as power, storage and bandwidth communication [14]. Another direction consists of a theoretic study on formalizing qualitative MDPs such as algebraic MDP [24,34] using our approach and assess to at what scale this model is more general than the existing ones.

References

[1] R. Aras and A. Dutech, An investigation into mathematical programming for finite horizon decentralized pomdps, *J Artif Intell Res* (*JAIR*) **37** (2010), 329–396.

[2] R. Bellman, Markov Decision Process, *Journal of mathematical Mech* **6** (1957), 610–616.

[3] A. Beynier and A. Mouaddib, A polynomial algorithm to solve decentralized MDP with temporal constraints, in: *In the Fourth International joint Conference on Autonomous Agents and Multi Agent Systems* (*AAMAS*), 2005, pp. 963–969.

[4] C. Boutilier, Towards a logic for qualitative decision theory, in: *Knowledge Representation and Reseaning* (*KR*), 1994, pp. 75–86.

[5] C. Boutilier, F. Bacchus and R. Brafman, Ucp-networks: A directed graphical representation of conditional utilities, in: *International Conference on Uncertainty in Artificial Intelligence* (*UAI*), 2001, pp. 56–64.

[6] R. Brafman and M. Tennenholtz, An axiomatic treatment of three qualitative decision criteria, *Journal of ACM* **47**(3) (2000), 452–482.

[7] H. Daellenbach and C.D. Kluyver, Note on multiple objective dynamic programming, *International Journal of Operational Research Society* **31** (1980), 591–594.

[8] J. Dibangoye, B. Chaib-Draa, A. Mouaddib and G. Shani, Topological order planner for POMDPs, in: *International Joint Conference of Artificial Intelligence* (*IJCAI*), 2009, pp. 1684–1689.

[9] E. Gat, R. Desai, R. Ivlev, J. Loch and D.P. Miller, Behavior control for robotic exploration of planetary surfaces, *IEEE Transactionon Robotics and Automation* **10** (1994), 490–503.

[10] M. Ehrgott and D. Podehl, Computation of ideal and nadir values and implications of their use in mcdm methods, *European Journal of Operational Research Optimization* **151** (2003), 119–139.

[11] R. Howard and A. Matheson, Influence diagrams. Principles of Applications of Decision Analysis 2, 1981.

[12] U. Junker, Preference-based search and multi-criteria optimization. In: CPAIOR, 2000.

[13] R. Keeney and H. Raiffa, Decision with Multiple Objectives: Preferences and Value Tradeoffs. John Wiley and Sons, Inc, 1976.

[14] S. Le_Gloannc, A. Mouaddib and F. Charpillet, Adaptive multiple resources consumption control for an autonomous rover. In: European Robotic System Symposium, 2008, pp. 1–11.

[15] M. Littman, A. Cassandra and L. Kealbling, Learning policies for partially observable environments: Scaling up, in: *International Conference on Machine Learning* (*ICML*), 1995, pp. 362–370.

[16] S. Mahadevan and N. Khaleeli, Robust mobile robot navigation using Partially-Observable Semi-Markov Decision Process. 51] P. Maybeck. Stochastic Models, Estimation, and Control 1, 1999.

[17] M. Mataric, Behavior-based control: Example from navigation, learning and group behavior, *Journal of Experimental and Theoretic Artificial Intelligence* **9**(2–3) (1997), 323–336.

[18] A. Mouaddib, Multi-criteria path planning, in: *IEEE International Conference on Robotic and Automaton* (*ICRA*), 2004, pp. 2814–1819.

[19] A. Mouaddib, B. Boussard and M. Bouzid, Multi-objective multiagent planning, in: *Internation Joint Conference on Autonomous Agent and MultiAgent Sytems* (*AAMAS*), 2007, pp. 123–130.

[20] R. Murphy, Human-robot interaction in rescue robotics. IEEE Systems, Man and Cybernetics Part C: Applications and Reviews, 2004, 138–153.

[21] N.J. Nilsson, Principles of Artificial Intelligence. Springer, 1982.

[22] A. Pahliani, M.T.J. Spaan and P.U. Lima, Decision-theoretic robot guidance for active cooperative perception, in: *Proc. of International Conference on Intelligent Robots and Systems*, 2009, pp. 4837–4842.

[23] J. Pearl, Probabilistics reasoning in intelligent systems: Networks of plausible influence. Morgan-kaufmann, 1981.

[24] P. Perny, O. Spanjaard and P. Weng, Algebraic Markov Decision Processes, in: *International Joint Conference on Artificial Intelligence* (*IJCAI*), 2005, pp. 1372–1377.

[25] P. Pirjanian, Multiobjective action selection in behavior-based control, in: *Sixth Symposium for Intelligent Robotic Systems*, 1998, pp. 83–92.

[26] J.M. Porta, M.T.J. Spaan and N. Vlassis, Robot planning in partially observable continuous domains, in: *Proc. of the 17th Belgian-Dutch Conference on Artifical Intelligence*, Brussels, Belgium, Oct. 2005, pp. 375–376, extended abstract.

[27] S. Thrun, S. Thayer, W. Whittaker, C. Baker, W. Burgard, D. Furguson, D. Hahnel, M. Montemerlo, A. Morris, C. Reverte and W. Whittaker, Autonomous exploration and mapping of abandoned mines, *IEEE Robotics and Automation Magazine* **11**(4) (2005).

[28] R. Sabbadin, Possibilistic Markov Decision Process, in: *European Conference on Artificial Intelligence* (*ECAI*), 2000, pp. 586–590.

[29] A. Sanfeliu, J. Andrade-Cetto, M. Barbosa, R. Bowden, J. Capitán, A. Corominas, A. Gilbert, J. Illingworth, L. Merino, J.M. Mirats, P. Moreno, A. Ollero, J. Sequeira and M.T.J. Spaan, Decentralized sensor fusion for ubiquitous networking robotics in urban areas, *Sensors* **10**(3) (2010), 2274–2314.

[30] R. Shachter, Evaluating influence diagrams, *Operation Research* **34**(6) (1981), 871–882.

[31] S.-W. Tan and J. Pearl, Qualitative decision theory, in: *Internation Conference of American Association of Artificial Intelligence* (*AAAI*), 1994, pp. 928–933.

[32] S. Thrun, W. Burgard and D. Fox, Real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping, in: *International Conference on Robotic and Automation* (*ICRA*), 2000, pp. 321–328.

[33] K. Wakuta and K. Togawa, Solution procedures for multi-objective markov decison processes, *Optimization* **43** (1998), 29–46.

[34] P. Weng, Axiomatic foundations for a class of generalized expected utility: Algebraic expected utility, in: *International Conference on Uncertainty in Artificial Intelligence* (*UAI*), 2006, pp. 520–527.

[35] G. Zoltan, Z. Kalmar and C. Szepesvari, Multi-criteria reinforcement learning, in: *International Conference on machine Learning* (*ICML*), 1998, pp. 197–205.