Multi-Objective Influence Diagrams with Possibly Optimal Policies

Radu Marinescu
IBM Research – Ireland
radu.marinescu@ie.ibm.com

Abdul Razak and **Nic Wilson**
Insight Centre for Data Analytics
University College Cork, Ireland
{abdul.razak,nic.wilson}@insight-centre.org

Abstract

The formalism of multi-objective influence diagrams has recently been developed for modeling and solving sequential decision problems under uncertainty and multiple objectives. Since utility values representing the decision maker's preferences are only partially ordered (e.g., by the Pareto order) we no longer have a unique maximal value of expected utility, but a set of them. Computing the set of maximal values of expected utility and the corresponding policies can be computationally very challenging. In this paper, we consider alternative notions of optimality, one of the most important one being the notion of *possibly optimal*, namely optimal in at least one scenario compatible with the inter-objective tradeoffs. We develop a variable elimination algorithm for computing the set of possibly optimal expected utility values, prove formally its correctness, and compare variants of the algorithm experimentally.

Introduction

Multi-Objective Influence Diagrams (MOID) (Marinescu, Razak, and Wilson 2012) are a recent extension of influence diagrams (ID) (Howard and Matheson 1984) that provide a general framework for modeling and solving sequential decision making problems under uncertainty and multiple objectives. They involve both chance variables, where the outcome is determined randomly based on the values assigned to other variables, and decision variables, which the decision maker can choose the value of, based on observations of some other variables. Uncertainty is represented (like in Bayesian networks) by a collection of conditional probability distributions, one for each chance variable. To cope with multiple and non-commensurate utility scales on which the decision maker's preferences are expressed, it is natural to consider multi-objective or multi-attribute utility functions (White, Sage, and Dozono 1984; Keeney and Raiffa 1993; Roy 1996; Ehrgott 1999). Therefore, the utility values are vectors in $I\!R^p$, with p being the number of objectives, and the overall value of an outcome is given by the sum of a set of local multi-objective utility functions.

Since the utility values are only partially ordered (for instance by the Pareto ordering) we no longer have a unique maximal value of expected utility, but a set of them. In general, given a MOID, the Pareto ordering on multi-objective utility is a rather weak one; the effect of this is that the set of maximal values of expected utility and the corresponding set of decision policies can often become huge. Therefore, the main drawback of MOIDs is that in most practical situations the set of undominated decision policies is often too large to be meaningful to the decision maker. Sometimes, the decision maker may be happy to provide additional tradeoffs between objectives, and these can be used to reduce further the set of undominated expected utility values and their policies (but typically not to singleton sets) as was recently shown in (Marinescu, Razak, and Wilson 2012).

Contribution In this paper we take a different approach and focus on finding optimal values of expected utility with respect to alternative notions of optimality, such as the notion of *possibly optimal solutions* considered in (Wilson, Razak, and Marinescu 2015a) for multi-objective constraint optimization problems. More specifically, we extend this approach to multi-objective influence diagrams and look to compute the set of *possibly optimal* values of expected utility and the corresponding decision policies. A possibly optimal policy is a policy that is optimal in at least one scenario that is compatible with the inter-objective tradeoffs. We develop a variable elimination algorithm for computing this set, and give the key formal properties that ensure the correctness of the computation. Quite often, the decision maker is happy to provide some preferences for one multi-objective utility vector over another; these may arise, for example, from an iterative elicitation technique. We use such inputs to infer other preferences and use them to eliminate dominated utility vectors during the computation. Our numerical experiments demonstrate the efficiency of the proposed algorithm both in terms of solution quality (as the cardinality of the solution sets obtained) and the solution times.

Proofs and additional experimental results are contained in a longer version of the paper (Marinescu, Razak, and Wilson 2017).

Background

Multi-objective Utility Values

Let p be the number of objectives (or attributes). A *multi-objective utility value* is defined to be a vector $\vec{u} = (u_1, \ldots, u_p) \in I\!R^p$, where u_i represents the utility with respect to objective $i \in \{1, \ldots, p\}$. We assume the standard pointwise arithmetic operations, namely $\vec{u} + \vec{v} = (u_1 + v_1, \ldots, u_p + v_p)$ and $q \times \vec{u} = (q \times u_1, \ldots, q \times u_p)$, where $q \in I\!R$. For finite sets $A, B \subseteq I\!R^p$, we define $A + B = \{\vec{u} + \vec{v} : \vec{u} \in A, \vec{v} \in B\}$.

DEFINITION **1 (weak Pareto order)** *Let $\vec{u}, \vec{v} \in I\!R^p$ so that $\vec{u} = (u_1, \ldots, u_p)$ and $\vec{v} = (v_1, \ldots, v_p)$. We define the binary relation $\geq$ on $I\!R^p$ by $\vec{u} \geq \vec{v} \iff \forall i = 1, \ldots, p, u_i \geq v_i$.*

Let $\succcurlyeq$ be a partial order on $I\!R^p$ and let $\vec{u}, \vec{v} \in I\!R^p$. If $\vec{u} \succcurlyeq \vec{v}$ then we say that $\vec{u}$ *dominates* $\vec{v}$ (as usual, the symbol $\succ$ refers to the asymmetric part of $\succcurlyeq$). Given finite sets $U, V \subseteq I\!R^p$, we say that U *dominates* V, denoted $U \succcurlyeq V$, if $\forall \vec{v} \in V$ $\exists \vec{u} \in U$ such that $\vec{u} \succcurlyeq \vec{v}$. Furthermore, given a finite set $U \subseteq I\!R^p$, we define the *maximal set* $\max_{\succcurlyeq}(U) = \{\vec{v} \in U \mid \nexists \vec{v} \in U, \vec{v} \succ \vec{u}\}$. When $\succcurlyeq$ is the weak Pareto ordering $\geq$, we call $\max_{\geq}(U)$ the *Pareto set*.

Imprecise Tradeoffs

We assume that the decision maker (DM) provides additional preferences i.e., a set Θ of pairs of the form $(\vec{u}, \vec{v})$ meaning that they prefer multi-objective vector $\vec{u}$ to $\vec{v}$. These can be viewed as a general kind of tradeoff between the objectives. We will use this input information to deduce further preferences, based on the assumption that the user's model is a weighted sum of the objectives (thus being a simple form of a Multi-attribute Utility Theory (MAUT) model (Figueira, Greco, and Ehrgott 2005)). The treatment of imprecise tradeoffs is based on the formalism introduced recently in (Marinescu, Razak, and Wilson 2012; 2013; Wilson, Razak, and Marinescu 2015a).

Let $\mathcal{W}$ be the set of vectors in $I\!R^p$ with all components non-negative and summing to 1. Elements of $\mathcal{W}$ are known as *weights vectors*. Each weights vector $\vec{w} \in \mathcal{W}$ induces a total pre-order $\succcurlyeq_{\vec{w}}$ on $I\!R^p$ by, for $\vec{u}, \vec{v} \in I\!R^p$, $\vec{u} \succcurlyeq_{\vec{w}} \vec{v}$ if and only if $\vec{w} \cdot \vec{u} \geq \vec{w} \cdot \vec{v}$, where e.g., $\vec{w} \cdot \vec{u} = \sum_{i=1}^{p} \vec{w}_i \vec{u}_i$.

We assume that the user's preference ordering over multi-objective vectors $I\!R^p$ is equal to $\succcurlyeq_{\vec{w}}$ for some weights vector $\vec{w} \in \mathcal{W}$ (which is unknown to us). For $\vec{w} \in \mathcal{W}$ and preference pair $(\vec{u}, \vec{v})$, we say that $\vec{w}$ satisfies $(\vec{u}, \vec{v})$ if $\vec{u} \succcurlyeq_{\vec{w}} \vec{v}$, i.e., if $\vec{w} \cdot \vec{u} \geq \vec{w} \cdot \vec{v}$. We also say, for set of pairs Θ, that $\vec{w}$ satisfies Θ if $\vec{w}$ satisfies each pair in Θ; let $\mathcal{W}(\Theta)$, the set of *(consistent) scenarios*, be all such $\vec{w}$. If we knew the weights vector $\vec{w}$, the problem would reduce to a single-objective problem. However, all we know is that $\vec{w} \in \mathcal{W}(\Theta)$. This leads to the induced preference relation $\succcurlyeq_\Theta$ defined as follows: $\vec{u} \succcurlyeq_\Theta \vec{v}$ if and only if $\vec{u} \succcurlyeq_{\vec{w}} \vec{v}$ for all $\vec{w} \in \mathcal{W}(\Theta)$; hence, $\succcurlyeq_\Theta$ is equal to the intersection of relations $\succcurlyeq_{\vec{w}}$ over all $\vec{w} \in \mathcal{W}(\Theta)$. Therefore, $\vec{u}$ is preferred to $\vec{v}$ if and only if the preference holds for every consistent scenario. The associated strict relation $\succ_\Theta$ is given by $\vec{u} \succ_\Theta \vec{v} \iff \vec{u} \succcurlyeq_\Theta \vec{v}$ and $\vec{v} \not\succcurlyeq_\Theta \vec{u}$. When Θ is empty, $\succcurlyeq_\Theta$ is just the Pareto ordering. We make the weak assumption that $\succcurlyeq_\Theta$ is a partial order.

Operators PO**,** CSD **and** POCSD Let $\mathcal{D}$ be a finite set of utility vectors. There is more than one way of defining the optimal utility vectors in $\mathcal{D}$. Given input preferences Θ, define the set $\mathrm{CSD}_\Theta(\mathcal{D})$ to be the $\succcurlyeq_\Theta$-undominated vectors in $\mathcal{D}$. (CSD stands for *Can Strictly Dominate*, since $\vec{u} \in \mathrm{CSD}_\Theta(\mathcal{D})$ if and only if for all non-equivalent $\vec{t} \in \mathcal{D}$ there exists a consistent scenario $\vec{w}$ in which $\vec{u}$ strictly dominates $\vec{t}$, i.e., $\vec{u} \succ_{\vec{w}} \vec{t}$ (Wilson and O'Mahony 2011).) For $\vec{w} \in \mathcal{W}(\Theta)$, we define $\mathrm{O}_{\vec{w}}(\mathcal{D})$ to be the vectors $\vec{u}$ in $\mathcal{D}$ that are optimal in scenario $\vec{w}$, i.e., that maximize $\vec{w} \cdot \vec{u}$. We define $\mathrm{PO}_\Theta(\mathcal{D})$ to be the set of possibly optimal utility vectors, i.e., the vectors $\vec{u} \in \mathcal{D}$ that are optimal in some consistent scenario, i.e., such that there exists $\vec{w} \in \mathcal{W}(\Theta)$ with $\mathrm{O}_{\vec{w}}(\mathcal{D}) \ni \vec{u}$. Thus, $\mathrm{PO}_\Theta(\mathcal{D})$ is equal to the union of $\mathrm{O}_{\vec{w}}(\mathcal{D})$ over all $\vec{w} \in \mathcal{W}(\Theta)$. We also define operator POCSD_Θ by $\mathrm{POCSD}_\Theta(\mathcal{D}) = \mathrm{PO}_\Theta(\mathcal{D}) \cap \mathrm{CSD}_\Theta(\mathcal{D})$. We will sometimes abbreviate PO_Θ to PO, and similarly for CSD_Θ and POCSD_Θ.

One can also define the set of *necessarily optimal* utility vectors NO_Θ, i.e., vectors that are optimal in every scenario. This will usually be empty. However, if NO_Θ is non-empty, then it is equal to CSD_Θ (Wilson and O'Mahony 2011), so it suffices to compute the latter.

Multi-objective Influence Diagrams

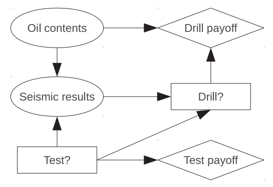
A *Multi-objective influence diagrams* (MOID) is defined by a tuple $\langle \mathbf{C}, \mathbf{D}, \mathbf{P}, \mathbf{U} \rangle$, where $\mathbf{C} = \{C_1, \ldots, C_n\}$ is a set of *chance variables* which specify the uncertain decision environment and $\mathbf{D} = \{D_1, \ldots, D_m\}$ is a set of *decision variables* which specify the possible decisions to be made. $\Omega(Y)$ denotes the domain of variable $Y \in \mathbf{C} \cup \mathbf{D}$, and for a set $\mathbf{S} \subseteq \mathbf{C} \cup \mathbf{D}$ such that $\mathbf{S} = \{Y_{i_1}, \ldots, Y_{i_k}\}$, we define $\Omega(\mathbf{S}) = \Omega(Y_{i_1}) \times \cdots \times \Omega(Y_{i_k})$.

As in Bayesian networks (Pearl 1988), each chance variable $C_i \in \mathbf{C}$ is associated with a *conditional probability table* (CPT) $P_i = P(C_i|pa(C_i))$, where $P_i \in \mathbf{P}$ and $pa(C_i) \subseteq \mathbf{C} \cup \mathbf{D} \setminus \{C_i\}$. Each decision variable $D_k \in \mathbf{D}$ has a parent set $pa(D_k) \subseteq \mathbf{C} \cup \mathbf{D} \setminus \{D_k\}$, denoting the variables whose values will be known at the time of the decision and may affect directly the decision.

The *multi-objective utility functions* $\mathbf{U} = \{U_1, \ldots, U_r\}$ are defined over subsets of variables $\mathbf{Q} = \{\mathbf{Q}_1, \ldots, \mathbf{Q}_r\}$, where $\mathbf{Q}_i \subseteq \mathbf{C} \cup \mathbf{D}$ is called the *scope* of U_i, and represent the preferences of the decision maker, namely $U_i : \Omega(\mathbf{Q}_i) \rightarrow I\!R^p$. Utility function U_i can be identified in the obvious way with a function to $2^{I\!R^p}$, by mapping each element of $\Omega(\mathbf{Q}_i)$ to a singleton subset of $I\!R^p$; we abuse notation by using U_i to refer to both functions. The local utility functions define a global multi-objective utility function that is additively decomposable, as follows: $U(\mathbf{C} \cup \mathbf{D}) = \sum_{j=1}^{r} U_j(\mathbf{Q}_j)$.

A *policy* for a MOID is a set of decision rules $\Delta = (\delta_1, \ldots, \delta_m)$ consisting of one rule for each decision variable. A *decision rule* for the decision $D_k \in \mathbf{D}$ is a mapping $\delta_k : \Omega(pa(D_k)) \rightarrow \Omega(D_k)$. Therefore, a policy Δ determines a value for each decision variable D_k (which depends on the parents set $pa(D_k)$). We assume acyclicity of the dependency relation on decision variables. Given a utility function U, a policy Δ yields a utility function $[U]_\Delta$ that involves no decision variables, by assigning their values using Δ. The expected utility of policy Δ is $EU_\Delta = \sum_{\mathbf{C}} [(\prod_{i=1}^{n} P_i \times \sum_{j=1}^{r} U_j)]_\Delta$. We therefore have $EU_\Delta \in I\!R^p$.

Solving a MOID means finding the set of policies that



Oil contents	P(O)		
	dry	wet	soak
	0.5	0.3	0.2

Test payoff	U_1(T)
Test?	
yes	{(-10,10)}
no	{(0,0)}

Seismic results		P(S | O,T)			
Oil cnt.	Test?	closed	open	diffuse	notest
dry	yes	0.1	0.3	0.6	0
dry	no	0	0	0	1
wet	yes	0.3	0.4	0.3	0
wet	no	0	0	0	1
soak	yes	0.5	0.4	0.1	0
soak	no	0	0	0	1

Drill payoff		U_2(O,D)
Oil cnt.	Drill?	
dry	yes	{(-70,18)}
dry	no	{(0,0)}
wet	yes	{(50,12)}
wet	no	{(0,0)}
soak	yes	{(200,8)}
soak	no	{(0,0)}

Figure 1: A bi-objective influence diagram.

generate maximal values of expected utility, i.e., values of utility in the set $\max_{\succcurlyeq}\{EU_\Delta \mid \text{policies } \Delta\}$. We say that a policy Δ is *undominated* if the corresponding expected utility vector EU_Δ is undominated.

Example 1 *In Figure 1 we show the influence diagram of the oil wildcatter problem (adapted from (Raiffa 1968)). An oil wildcatter must decide either to* drill *or* not to drill *for oil at a specific site. Before drilling, a* seismic test *could help determine the geological structure of the site. The test results can show a* closed *reflection pattern (indication of significant oil), an* open *pattern (indication of some oil), or a* diffuse *pattern (almost no hope of oil). The special value* notest *is used if no seismic test is performed. There are therefore two decision variables, T (Test) and D (Drill), and two chance variables S (Seismic results) and O (Oil contents). The probabilistic knowledge consists of the conditional probability tables $P(O)$ and $P(S|O,T)$.*

We consider a utility function with two attributes representing the testing/drilling payoff *and* damage *to the environment, respectively. The utility of testing (represented by the component $U_1(T)$) is $(-10, 10)$, whereas the utility of drilling (represented by the component $U_2(O,D)$) is $(-70, 18)$, $(50, 12)$, $(200, 8)$ for a dry, wet and soaking hole, respectively. Therefore, the utility function is the sum of $U_1(T)$ and $U_2(O,D)$. The aim is to find optimal policies that maximize the payoff and minimize the damage to environment. The dominance relation is defined in this case by $\vec{u} \geq \vec{v} \Leftrightarrow u_1 \geq v_1$ and $u_2 \leq v_2$ (e.g., $(10, 2) \geq (8, 4)$ and $(10, 2) \not\geq (8, 1)$). The Pareto set $\max_{\geq}\{EU_\Delta \mid \text{policies } \Delta\}$ contains 4 elements, i.e., $\{(22.5, 17.56), (20, 14.2), (11, 12.78), (0, 0)\}$, corresponding to the four optimal (undominated) policies shown below:*

	Δ_1	Δ_2	Δ_3	Δ_4
δ_T	*yes*	*no*	*yes*	*no*
δ_D	*yes (S = closed)* *yes (S = open)* *no (S = diffuse)*	*yes (S = notest)*	*yes (S = closed)* *no (S = open)* *no (S = diffuse)*	*no (S = notest)*
EU_{Δ_i}	$\{(22.5, 17.56)\}$	$\{(20, 14.2)\}$	$\{(11, 12.78)\}$	$\{(0, 0)\}$

Optimality Operators

We want to show that MOID computations can be correctly performed with respect to optimality definitions PO and POCSD, as well as CSD. In this section, building on work by (Wilson, Razak, and Marinescu 2015a), we consider a more general and abstract notion of optimality, defined axiomatically. We show that the operators satisfy properties that ensure correctness of a variable elimination computation.

Axioms and Properties

For set of alternatives or outcomes A, $\mathrm{OPT}(A)$ is intended to be the set of optimal ones, with respect to some particular notion of optimality. Thus, operator OPT is a function that has input some set of outcomes A (which is a subset of some universal set $\mathcal{D}$ of outcomes), and returns a subset of A.

DEFINITION **2 (optimality operator)** *Let $\mathcal{D}$ be a finite set. We say that* OPT *is an* optimality operator *over $\mathcal{D}$ if* OPT *is a function from $2^{\mathcal{D}}$ to $2^{\mathcal{D}}$ satisfying the following three conditions, for arbitrary $A, B \subseteq \mathcal{D}$.*

(1) $\mathrm{OPT}(A) \subseteq A$;

(2) If $A \subseteq B$ *then* $\mathrm{OPT}(B) \cap A \subseteq \mathrm{OPT}(A)$;

(3) If $\mathrm{OPT}(B) \subseteq A \subseteq B$ *then* $\mathrm{OPT}(A) = \mathrm{OPT}(B)$.

The properties entail that OPT is idempotent (Lemma 1 of (Wilson, Razak, and Marinescu 2015b)). The following property, known as *Path Independence* in the social choice function literature (Plott 1973; Aizerman and Malishevski 1981; Moulin 1985), is another important consequence of the axioms, see Proposition 1 of (Wilson, Razak, and Marinescu 2015b):

PROPOSITION **1** *Let* OPT *be an optimality operator over $\mathcal{D}$. Then* OPT *satisfies the Union Decomposition property, i.e., for all $A, B \subseteq \mathcal{D}$,* $\mathrm{OPT}(A \cup B) = \mathrm{OPT}(\mathrm{OPT}(A) \cup \mathrm{OPT}(B))$.

Additive Decomposition Suppose that $\mathcal{D}$ is a subset of $\mathbb{R}^p$ for some $p = 1, 2, \ldots$. We say that $\mathrm{OPT} : 2^{\mathcal{D}} \to 2^{\mathcal{D}}$ satisfies the *Additive Decomposition property* if for any $A, B \subseteq \mathcal{D}$ such that $A + B \subseteq \mathcal{D}$, $\mathrm{OPT}(A + B) = \mathrm{OPT}(\mathrm{OPT}(A) + \mathrm{OPT}(B))$. This property is necessary for the correctness of our variable elimination approach.

PO, CSD and POCSD as Optimality Operators

Proposition 4 of (Wilson, Razak, and Marinescu 2015a) shows that PO_Θ, CSD_Θ and POCSD_Θ are optimality operators and satisfy Union Decomposition and Additive Decomposition.

Influence diagram computation involves multiplication of a set of utility vectors by a positive real (probability value). The Scaling property relates to this. We say that OPT satisfies *Scaling* if for any $A \subseteq \mathcal{D}$ and $\lambda \in \mathbb{R}$ with $\lambda > 0$, we have $\mathrm{OPT}(\lambda A) = \mathrm{OPT}(\lambda \mathrm{OPT}(A))$, whenever λA and $\lambda \mathrm{OPT}(A)$ are in $\mathcal{D}$. The fact that $\vec{t} \succcurlyeq_{\vec{w}} \vec{u}$ if and only if $\lambda\vec{t} \succcurlyeq_{\vec{w}} \lambda\vec{u}$, leads to $\mathrm{OPT}(\lambda A) = \lambda \mathrm{OPT}(A)$, for $\mathrm{OPT} \in \{\mathrm{PO}, \mathrm{CSD}, \mathrm{POCSD}\}$, which, applying OPT to both sides and using idempotence, implies Scaling.

PROPOSITION **2** *Operators* PO_Θ, CSD_Θ *and* POCSD_Θ *satisfy Scaling.*

Example 2 *Continuing Example 1, suppose that we have the input tradeoff* $\Theta = \{((3,0),(6,-4))\}$*, where the* -4 *represents 4 units of environmental damage. Then,* $\mathrm{CSD} = \{(11,-12.78),(20,-14.2),(0,0)\}$ *(where the minus sign indicates that the corresponding objective value is to be minimized) because* $(20,-14.2) \succcurlyeq_{\Theta} (22.5,-17.56)$*. We have that* $\mathrm{PO} = \{(20,-14.2),(0,0)\}$ *because the utility value* $(11,-12.78) \notin \mathrm{PO}$ *since* $\nexists \vec{w} \in \mathcal{W}(\Theta)$ *satisfying the linear inequalities* $-1.5w_1 + 2w_2 \geq 0$*,* $-9w_1 + 1.42w_2 \geq 0$*,* $11w_1 - 12.78w_2 \geq 0$*,* $w_1 + w_2 = 1$ *and* $w_1 \geq 0, w_2 \geq 0$*. Similarly, we can check that the utility values* $(20,-14.2)$ *and* $(0,0)$ *are possibly optimal with respect to the scenarios* $(0.5,0.5)$ *and* $(0.4,0.6)$*, respectively. Consequently, we also have that* $\mathrm{POCSD} = \{(20,-14.2),(0,0)\}$*.*

Equivalences using Convex Closure

For sets of utility vectors we consider a form of equivalence based on convex closure, i.e., two sets are equivalent if and only if they have same convex closure: $A \equiv_{\mathcal{C}} B$ if and only if $\mathcal{C}(A) = \mathcal{C}(B)$, where $\mathcal{C}(A)$ is the convex hull of A.

We also make use of another form of equivalence $\equiv$, given partial order $\succcurlyeq = \succcurlyeq_{\Theta}$ defined above. For $\mathcal{U}, \mathcal{V} \subseteq I\!R^p$, we say that $\mathcal{U} \succcurlyeq \mathcal{V}$ if every element of $\mathcal{V}$ is (weakly) dominated by some element of $\mathcal{U}$, and define $\mathcal{U} \approx \mathcal{V}$ if and only if $\mathcal{U} \succcurlyeq \mathcal{V}$ and $\mathcal{V} \succcurlyeq \mathcal{U}$. Given $\mathcal{U}, \mathcal{V} \subseteq I\!R^p$, we define the equivalence relation $\equiv$ by $\mathcal{U} \equiv \mathcal{V}$ if and only if $\mathcal{C}(\mathcal{U}) \approx \mathcal{C}(\mathcal{V})$. Therefore, two sets of utility vectors are considered equivalent if, for every convex combination of elements of one, there is a convex combination of elements of the other which is at least as good (with respect to the partial order $\succcurlyeq$ on $I\!R^p$). Note that if $A \equiv_{\mathcal{C}} B$ then $A \equiv B$. We have that:

PROPOSITION **3** *Suppose that* $A, B \subseteq \mathcal{D}$ *are such that* $A \equiv_{\mathcal{C}} B$*. Then* $\mathrm{PO}(A) \equiv_{\mathcal{C}} \mathrm{PO}(B)$*.*

Algorithm 1: MOVE(OPT)

```
Data: A MOID $\mathcal{M} = \langle \mathbf{C}, \mathbf{D}, \mathbf{P}, \mathbf{U} \rangle$ with $p > 1$ objectives, a legal
      elimination ordering $\tau = (Y_1, \dots, Y_t)$ of the variables $\mathbf{C} \cup \mathbf{D}$,
      optimality operator $\mathrm{OPT}$
Result: Set of $\mathrm{OPT}$-optimal expected utility values and corresponding
        decision policies
1  Let $\Phi = \{\phi | \phi \in \mathbf{P}\}, \Psi = \{\psi | \psi \in \mathbf{U}\}$;
2  foreach variable $Y$ in $\tau = (Y_1, \dots, Y_t)$ do
3      Create bucket $\mathcal{B}_Y$ and its associated sets $\Phi_Y$ and $\Psi_Y$;
4      Let $\Phi_Y = \{\phi | \phi \in \Phi, Y \in sc(\phi)\}$ and $\Phi \leftarrow \Phi \setminus \Phi_Y$;
5      Let $\Psi_Y = \{\psi | \psi \in \Psi, Y \in sc(\psi)\}$ and $\Psi \leftarrow \Psi \setminus \Psi_Y$;
6  foreach variable $Y$ in ordering $\tau = (Y_1, \dots, Y_t)$ do
7      if $Y \in \mathbf{C}$ is a chance variable then
8          $\phi_Y \leftarrow \sum_Y \prod_{\phi \in \Phi_Y} \phi$;
9          foreach $\psi \in \Psi_Y$ do
10             $\psi_Y \leftarrow \frac{1}{\phi_Y} \sum_Y (\prod_{\phi \in \Phi_Y} \phi) \times \psi$;
11     else if $Y \in \mathbf{D}$ is a decision variable then
12         foreach $\phi \in \Phi_Y$ do
13             Let $\mathbf{S} = sc(\phi) \setminus \{Y\}$;
14             Compute $\phi_Y$ as follows: $\forall \mathbf{x} \in \Omega(\mathbf{S}), \phi_Y(\mathbf{x}) = \phi(\mathbf{x}y)$ for
                 any value $y \in \Omega(Y)$ ;
15         $\psi_Y \leftarrow \bigvee_Y \sum_{\psi \in \Psi_Y} \psi$ ;
16     Place each of the $\phi_Y$ (resp. $\psi_Y$) to the set $\Phi_Z$ (resp. $\Psi_Z$) of the highest
         bucket $\mathcal{B}_Z$ corresponding to a variable $Z$ in $sc(\phi_Y)$ (resp. $sc(\psi_Y)$);
         If $Y$ is the last variable then add $\phi_Y$ to $\Phi_0$ and $\psi_Y$ to $\Psi_0$;
17 Let $\mathcal{E} = \mathrm{OPT}((\prod_{\phi \in \Phi_0} \phi) \times (\sum_{\psi \in \Psi_0} \psi))$;
18 foreach variable $Y$ in reversed ordering $\tilde{\tau} = (Y_t, \dots, Y_1)$ do
19     if $Y \in \mathbf{D}$ is a decision variable then
20         $\psi \leftarrow (\prod_{\phi \in \Phi_Y} \phi) \times (\sum_{\psi \in \Psi_Y} \psi)$;
21         Let $\mathbf{S} = sc(\psi) \setminus \{Y\}, \delta' : \Omega(\mathbf{S}) \rightarrow 2^{\Omega(Y)}$;
22         foreach $\mathbf{x} \in \Omega(\mathbf{S})$ do
23             $\mathbf{y}' = \{y | \psi(\mathbf{x}y) \in \mathrm{OPT}(\bigcup_{y \in \Omega(Y)} \psi(\mathbf{x}y))\}$;
24             $\delta'(\mathbf{x}) = \mathbf{y}'$;
25         $\Delta_Y = \{\delta_Y | \delta_Y(\mathbf{x}) = y, \forall \mathbf{x} \in \Omega(\mathbf{S}), y \in \delta'(\mathbf{x})\}$
26 Let $\Delta^* = \{(\delta_1, \dots, \delta_m) | \forall \delta_Y \in \Delta_Y, Y \in \mathbf{D}\}$;
27 return $(\mathcal{E}, \Delta^*)$
```

Variable Elimination

We now describe the main contribution of this paper, that is a variable elimination algorithm that eliminates chance variables and decision variables, respectively, using marginalization operators $\sum_X$ and $\bigvee_X$, to compute the set of OPT-optimal values of expected utility and their corresponding decision policies, where OPT is an optimality operator.

The decision variables are assumed to be temporally ordered. This property which is known as *regularity* induces a *strict partial order* $\prec$ over all the variables $\mathbf{C} \cup \mathbf{D}$, given by $\mathbf{I}_0 \prec D_1 \prec \mathbf{I}_1 \prec \cdots \prec D_m \prec \mathbf{I}_m$, where $\mathbf{I}_0, \dots, \mathbf{I}_m$ are disjoint sets of chance variables, and for each k, $0 < k < m$, $\mathbf{I}_k$ are the chance variables observed between D_k and D_{k+1}, $\mathbf{I}_0$ are the initial evidence variables, while $\mathbf{I}_m$ are the unobserved variables. Consequently, the variables must be processed along a *legal elimination ordering* that respects the partial order $\prec$ induced by the decision variables, namely the reverse of the elimination ordering is some extension of $\prec$ to a total order (Jensen, Jensen, and Dittmer 1994; Dechter 2000; Marinescu, Razak, and Wilson 2012). The set of OPT-optimal expected utility values can be found by computing:

$$\sum_{\mathbf{I}_0} \bigvee_{D_1} \cdots \sum_{\mathbf{I}_{m-1}} \bigvee_{D_m} \sum_{\mathbf{I}_m} \left(\prod_{i=1}^{n} P_i \times \sum_{j=1}^{r} U_j \right) \tag{1}$$

For a real valued function f defined over a set of variables $\mathbf{S}$, $f : \Omega(\mathbf{S}) \rightarrow I\!R$, and a variable $X \in \mathbf{S}$, the function $(\sum_X f)$ is defined over $\mathbf{U} = \mathbf{S} \setminus \{X\}$ as follows. For every $\mathbf{u} \in \Omega(\mathbf{U})$, $(\sum_X f)(\mathbf{u}) = \sum_{x \in \Omega(X)} f(\mathbf{u}, x)$. Similarly, for $g : \Omega(\mathbf{S}) \rightarrow 2^{I\!R^p}$, the functions $(\bigvee_X g)$ and $(\sum_X g)$ are defined as: for every $\mathbf{u} \in \Omega(\mathbf{U})$, $(\bigvee_X g)(\mathbf{u}) = \mathrm{OPT}(\bigcup_{x \in \Omega(X)} g(\mathbf{u}, x))$, and $(\sum_X g)(\mathbf{u}) = \mathrm{OPT}(\sum_{x \in \Omega(X)} g(\mathbf{u}, x))$ (here $\sum_x$ denotes the repeated application of the $+$ operator defined for sets of utility vectors).

Algorithm MOVE

Algorithm 1 describes our variable elimination procedure, called MOVE, for solving MOIDs. It takes as input an optimality operator OPT (e.g., PO), a legal elimination ordering $\tau = Y_1, \ldots, Y_t$ of the variables, and partitions the input functions into a set of *buckets*, such that each bucket $\mathcal{B}_Y$ is labeled by a single variable Y and is associated with sets Φ_Y and Ψ_Y containing all input probability and utility functions whose highest variable in their scope is Y.

MoVE processes each bucket, top-down from the first to the last, by a variable elimination procedure that computes new probability (denoted by ϕ) and utility (denoted by ψ) components which are then placed in corresponding buckets (lines 6-16). For a chance variable Y, the ϕ_Y component is generated by multiplying all probability functions in that bucket and eliminating Y by summation (line 8). The ψ_Y component is computed as the average utility in that bucket, normalized by the bucket's compiled ϕ (line 10). Most importantly, the operator $\sum_Y$ uses OPT during elimination to ensure that only OPT-optimal utility values are kept in ψ_Y.

For a decision variable Y, we compute the ψ_Y component by summing all the utility functions in that bucket and eliminating Y by operator $\bigvee_Y$ which prunes away non OPT-optimal utility values (line 15). Each of the probability functions in this bucket is a constant when viewed as a function of the bucket's decision variable and therefore the compiled ϕ_Y is a constant as well (lines 12-14) (Jensen, Jensen, and Dittmer 1994; Wilson and Marinescu 2012).

The set $\mathcal{E}$ of OPT-optimal expected utility values is obtained after eliminating the last variable, by combining all remaining constant utility and probability functions.

In a second, bottom-up step, MoVE generates all policies that produce expected utility values in set $\mathcal{E}$ (i.e., OPT-optimal policies). The decision buckets are processed in reversed order, from the last to the first. Let Y be the current decision variable. The algorithm computes the set Δ_Y of OPT-optimal decision rules associated with Y as follows. It first combines all utility and probability functions residing in bucket $\mathcal{B}_Y$ into a temporary function ψ which is defined on Y and its parent-set $pa(Y)$. Then, for each configuration $\mathbf{x}$ of the parent-set $pa(Y)$, it records the subset of domain values $\mathbf{y}' \subseteq \Omega(Y)$ that correspond to OPT-optimal values of ψ (lines 21-24). The sets $\mathbf{y}'$ are used to generate the decision rules $\delta_Y \in \Delta_Y$, by $\delta_Y(\mathbf{x}) = y$, for each configuration $\mathbf{x} \in \Omega(pa(Y))$ and $y \in \mathbf{y}'$ (line 25). Finally, the algorithm enumerates all OPT-optimal policies by selecting iteratively for each decision variable Y an OPT-optimal decision rule δ_Y from the corresponding set Δ_Y (line 26).

Complexity Based on previous work (Dechter 2000; Marinescu, Razak, and Wilson 2012), the complexity of MoVE can be bounded (time and space) by $O(\mathcal{K} \cdot N \cdot k^{w^*})$, where N is the total number of variables, k bounds the domain size, and w^* is the induced width of the legal elimination ordering. Since the utility values are vectors in $\mathbb{R}^p$, it is not easy to bound $\mathcal{K}$, the size of the set of OPT-optimal expected utility values.

Correctness of the MoVE Computation

We will prove the correctness of the variable elimination algorithm, showing that the set of expected utility values computed is *equivalent* to the set of OPT-optimal utility values, where OPT is an alternative optimality operator.

Algorithm MoVE(I): MoVE without Pruning

We first consider the version MoVE(I) of the algorithm which uses the identity optimality operator I given by $\mathrm{I}(A) = A$, for all A. This is equivalent to removing the OPT operator from the marginalization operators $\sum_X$ and $\bigvee_X$, and lines 17 and 23 of Algorithm 1.

Given an MOID $\mathcal{M}$, let $H_\mathcal{M}$ be the set of all values of (expected) utility generated by all possible policies, so that $\vec{u}$ is in $H_\mathcal{M}$ if and only if there exists some policy with expected utility of $\vec{u}$. The set of possibly optimal values of utility is equal to $\mathrm{PO}_\Theta(H_\mathcal{M})$. The set of undominated values of utility is equal to $\mathrm{CSD}_\Theta(H_\mathcal{M})$, which equals $\max_{\succeq_\Theta}(H_\mathcal{M})$. The algorithm in (Marinescu, Razak, and Wilson 2012) computes $\mathrm{CSD}_\Theta(H_\mathcal{M})$ up to $\equiv$-equivalence.

The following result follows directly from Theorem 4 and Proposition 5 of (Wilson and Marinescu 2012) (using $\succcurlyeq$ equalling $=$, since then $\equiv$ in Theorem 4 equals convex closure equivalence).

THEOREM 1 *Let G be the output of* MoVE(I). *Then, $G \equiv_\mathcal{C} H_\mathcal{M}$.*

Using Proposition 3, this immediately implies the following corollary, stating that if we apply the PO operator to the result of the no-pruning algorithm, then we obtain, up to convex-closure-based equivalence, the set of possibly optimal values of utility.

COROLLARY 1 *Let G be the output of* MoVE(I). *Then* $\mathrm{PO}(G) \equiv_\mathcal{C} \mathrm{PO}(H_\mathcal{M})$.

However, using the no-pruning algorithm to generate G first will tend to be very computationally expensive. Instead one would like to be able to apply the PO operator at intermediate stages of the algorithm. The algorithm involves applying the operations of union, addition and scalar multiplication to sets of utility vectors. The fact that the PO operator satisfies Union Decomposition, Additive Decomposition and Scaling implies that, if we apply PO to sets of utility vectors generated during the algorithm, the result is equal to $\mathrm{PO}(G)$. This leads to the theorem below, which shows that MoVE(PO) computes $\mathrm{PO}(H_\mathcal{M})$, the set of possibly optimal utility vectors, up to convex closure equivalence. A related result holds for MoVE(POCSD) (see (Marinescu, Razak, and Wilson 2017) for details).

THEOREM 2 *Let F be the set of utility vectors generated by* MoVE(PO). *Then $F \equiv_\mathcal{C}$ $\mathrm{PO}(H_\mathcal{M})$, i.e., F is convex-closure-equivalent to the set of possibly optimal utility vectors.*

Experiments

We evaluate empirically the performance of the proposed variable elimination algorithms on random multi-objective influence diagrams with tradeoffs. The problem instances were generated using the random problem generator from (Marinescu, Razak, and Wilson 2012) using 2, 3 and 5 objectives, respectively. We consider algorithms MoVE(PO) and MoVE(POCSD), which compute sets of PO and

size	w^*	MoVe(Pareto)					MoVe(CSD)					MoVe(PO)				
(C,D,O)		#	time	avg	stdev	med	#	time	avg	stdev	med	#	time	avg	stdev	med
$K = 1; T = 0$																
(15,5,2)	9	9	139.58	2,714	2,864	1,673	100	0.1	22	93	1	100	0.21	3	6	1
(25,5,2)	11	7	18.25	2,344	2,614	269	97	35.87	75	296	1	100	0.29	4	7	1
(35,5,2)	14	2	283.29	9,115	8,934	9,024	96	52.34	105	537	1	100	15.09	29	122	1
(45,5,2)	16	2	397.72	7,596	7,536	7,566	90	133.68	159	561	1	100	9.40	13	31	1
(55,5,2)	18	4	717.68	10,422	5,851	14,896	94	98.68	136	537	1	97	45.19	8	21	1
$K = 2; T = 1$																
(15,5,3)	9	6	10.69	4,889	4,069	5,830	98	25.85	224	824	1	98	30.37	52	232	1
(25,5,3)	11	2	4.42	4,000	3,938	3,969	84	216.88	582	2170	1	85	197.31	149	699	1
(35,5,3)	14	0					95	73.81	203	623	1	99	17.47	15	33	2
(45,5,3)	16	2	242.68	15,431	1,729	8,580	78	270.45	266	989	4	86	181.96	23	55	4
(55,5,3)	18	0					77	243.52	73	334	2	85	196.51	13	35	2
$K = 6; T = 3$																
(15,5,5)	9	3	65.73	15,062	12,229	14,741	98	24.13	36	143	2	97	36.56	8	25	2
(25,5,5)	11	1	50.35	21,074	0	21,074	95	63.40	155	659	4	97	42.05	14	34	4
(35,5,5)	14	0					88	152.95	299	1499	2	92	118.76	29	92	2
(45,5,5)	16	0					79	217.52	230	996	2	88	150.42	23	47	3
(55,5,5)	18	0					82	245.84	191	569	2	93	107.75	22	44	3
$K = 1; T = 0$																
(15,10,2)	12	5	91.82	6,856	8,280	3,175	88	154.35	304	975	1	89	182.40	207	848	1
(25,10,2)	17	1	808.94	4,964	0	4,964	85	220.18	15	49	1	91	138.48	26	149	1
(35,10,2)	20	0					61	510.36	154	721	1	82	338.56	24	65	1
(45,10,2)	22	0					20	816.32	287	653	6	22	783.43	10	17	3
(55,10,2)	26	0					7	1004.2	45	87	1	6	1028.83	16	18	7
$K = 2; T = 1$																
(15,10,3)	12	0					99	12.52	17	113	2	99	15.09	3	5	2
(25,10,3)	17	0					49	627.47	146	936	1	54	576.97	10	27	1
(35,10,3)	20	0					30	823.64	521	1597	2	39	747.22	46	138	2
(45,10,3)	22	0					20	962.45	46	165	3	23	636.289	60	131	4
(55,10,3)	26	0					0					9	841.95	34	52	5
$K = 6; T = 3$																
(15,10,5)	12	0					99	17.37	1340	868	3	97	42.83	19	107	2
(25,10,5)	17	0					65	415.73	222	1362	1	70	412.24	26	94	1
(35,10,5)	20	0					33	732.06	553	1137	16	42	728.14	40	62	17
(45,10,5)	22	0					19	468.08	7	14	1	22	705.79	3	3	1
(55,10,5)	26	0					0					1	1195.74	97	0	97

Table 1: Results comparing algorithms MoVe(Pareto), MoVe(CSD), and MoVe(PO). Time limit 20 minutes.

POCSD expected utility values. We also ran MoVe(CSD) for computing sets of CSD expected utility values.

We generated consistent random tradeoffs between the objectives of a given problem instance using the approach from (Marinescu, Razak, and Wilson 2012). The random tradeoffs are mainly characterized by the parameters K and T, where K is the number of pairwise or binary tradeoffs and T is the number of 3-way tradeoffs, respectively.

For reference, we also ran MoVe(Pareto) for computing Pareto optimal sets without any tradeoffs (Marinescu, Razak, and Wilson 2012). All algorithms were written in C++ and the experiments were run on a 2.6GHz processor with 4GB of RAM.

Comparison between the Pareto, CSDs and POs Table 1 compares cardinalities of the sets of Pareto, CSD and PO expected utility values for random influence diagrams with 2, 3 and 5 objectives, respectively. The tradeoff instances were generated with $(K = 1, T = 0)$ for 2 objectives, $(K = 2, T = 1)$ for 3 objectives, and $(K = 6, T = 3)$ for 5 objectives, respectively. For each problem size (characterized by the number of chance variables C, number of decision variables D, number of objectives O), we generated 100 random instances. We report the average cardinality of these expected utility sets (together with the standard deviation and median), the CPU time in seconds, as well as the number of problem instances solved. All algorithms were allotted a 20 minute time limit per problem instance. An empty cell in the table denotes that the respective algorithm could not solve any of the problem instances. For computing CSDs we take the approach from (Marinescu, Razak, and Wilson 2012), where we replaced the faster dominance checks between the multi-objective utility values using the matrix multiplication presented in (Marinescu, Razak, and Wilson 2013).

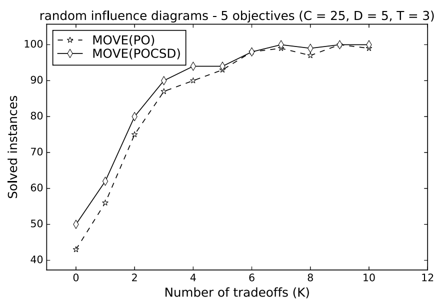


Figure 2: Number of solved instances as a function of pairwise tradeoffs. Time limit 20 minutes.

We can see that the POs become much smaller than the corresponding CSDs, especially when the difficulty of the problems increases. For example, for problems with 5 objectives, 25 chance variables and 5 decision, POs are more than ten times smaller than CSDs. In addition, MoVe(PO) is able to solve slightly more problem instances than MoVe(CSD). Although applying the PO operator is relatively expensive, because of the repeated checking of the optimality condition using a linear programming solver, the algorithm for computing POs is faster than the algorithm for computing CSDs, which implies that the former prunes more effectively than the latter. We also observed that MoVe(PO) and MoVe(POCSD) performed very similarly, in many cases solving the same number of instances with exactly the same cardinality of the corresponding sets, and therefore, the computed set PO being a subset of the computed set CSD. It may well be the case that the randomly generated tradeoffs lead to the set PO almost always being a subset of CSD.

Impact of Increasing the Number of Tradeoffs Figure 2 shows the impact of the number of pairwise tradeoffs K for random influence diagram instances with 5 objectives, 25 chance variables and 5 decision variables. We fixed the number of 3-way tradeoffs to $T = 3$. Results were taken over 100 problem instances with a 20 minute time limit per instance. We can see that, as K increases, the PO and POCSD algorithms solve more problems and consequently the running time decreases substantially.

Related Work

Our notion of possible optimality is related to the convex coverage sets introduced by (Roijers, Whiteson, and Oliehoek 2013; 2014) in the context of multi-objective coordination graphs. The algorithms proposed therein are also based on variable elimination and therefore time and space exponential in the induced widths of the underlying graphs.

More recently, (Benabbou and Perny 2015) proposed a multi-objective A* search approach for computing possibly optimal solutions in the context of multi-objective combinatorial optimization. An incremental elicitation mechanism is also embedded into the search so as to reduce the set of admissible scenarios until a nearly-optimal solution is found. For soft constraint problems with missing preferences, (Gelain et al. 2010) describes a branch and bound search algorithm augmented with a preference elicitation scheme for computing possibly optimal solutions. Similarly, in computational social choice (Xia and Conitzer 2011) presents methods to determine possible and necessary winners when incomplete preferences are available.

The work that is closest in spirit with ours is that by (Roijers, Whiteson, and Oliehoek 2015) who introduce the notion of optimistic linear support for multi-objective POMDPs. The basic idea is to solve a series of scalarized single-objective POMDPs, each corresponding to a different weighting of the objectives. The output is a set of policies that are optimal for some weighting (or scenario), which is very similar to our approach. However, their solving method is an iterative one which looks at different weightings one at a time (reusing results from previous iterations), whereas ours is based on dynamic programming and uses a linear programming formulation for checking possibly optimal solutions.

Conclusion

We introduced a variable elimination algorithm for computing the set of OPT-optimal values of expected utility (and the corresponding decision policies) for multi-objective influence diagrams with tradeoffs, where OPT is an optimality operator. We also gave the key formal properties and proved the correctness of the computation. Our experimental results indicate that computing PO and POCSD sets of expected utility values is generally faster and can scale up to more difficult problems involving more decisions and objectives. Moreover, the PO/POCSD sets are often considerably smaller than the corresponding CSD ones, thus yielding fewer policies. This is important because the decision maker can be presented with the PO/POCSD policies first before making any decision or eliciting more tradeoffs.

Acknowledgements

This work was supported in part by the Science Foundation Ireland grant SFI/12/RC/2289. We are grateful to the reviewers for their helpful comments.

References

Aizerman, M., and Malishevski, A. 1981. General theory of best variants choice: Some aspects. *IEEE Transactions on Automatic Control* 26:1030–1040.

Benabbou, N., and Perny, P. 2015. Incremental weight elicitation for multiobjective state space search. In *Proc. AAAI-2015*, 1093–1099.

Dechter, R. 2000. A new perspective on algorithms for optimizing policies under uncertainty. In *Artificial Intelligence Planning Systems (AIPS)*, 72–81.

Ehrgott, M. 1999. *Multicriteria optimization*. Springer.

Figueira, J.; Greco, S.; and Ehrgott, M. 2005. *Multiple Criteria Decision Analysis—State of the Art Surveys*. Springer International Series in Operations Research and Management Science Volume 76.

Gelain, M.; Pini, M. S.; Rossi, F.; Venable, K. B.; and Walsh, T. 2010. Elicitation strategies for soft constraint problems with missing preferences: Properties, algorithms and experimental studies. *Artif. Intell.* 174(3-4):270–294.

Howard, R., and Matheson, J. 1984. Influence diagrams. In *Readings on the Principles and Applications of Decision Analyis*, 721–762.

Jensen, F.; Jensen, V.; and Dittmer, S. 1994. From influence diagrams to junction trees. In *Proc. UAI-94*, 367–363.

Keeney, R., and Raiffa, H. 1993. *Decisions with multiple objectives: preferences and value tradeoffs*. Cambridge University Press.

Marinescu, R.; Razak, A.; and Wilson, N. 2012. Multi-objective influence diagrams. In *Proc. UAI-2012*, 574–583.

Marinescu, R.; Razak, A.; and Wilson, N. 2013. Multi-objective constraint optimization with tradeoffs. In *Proc. CP-2013*, 497–512.

Marinescu, R.; Razak, A.; and Wilson, N. 2017. *Multi-objective Influence Diagrams with Possibly Optimal Policies (extended version available online)*. http://ucc.insight-centre.org/nwilson/POIDAAAI17longer.pdf.

Moulin, H. 1985. Choice functions over a finite set: a summary. *Social Choice and Welfare* 2(2):147–160.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann.

Plott, C. R. 1973. Path independence, rationality, and social choice. *Econometrica* 41:1075–1091.

Raiffa, H. 1968. *Decision analysis*. Addison-Wesley.

Roijers, D.; Whiteson, S.; and Oliehoek, F. 2013. Computing convex coverage sets for multi-objective coordination graphs. In *Proc. ADT-2013*, 309–323.

Roijers, D.; Whiteson, S.; and Oliehoek, F. 2014. Linear support for multi-objective coordination graphs. In *Proc. AAMAS-2014*, 1297–1304.

Roijers, D.; Whiteson, S.; and Oliehoek, F. 2015. Point-based planning for multi-objective POMDPs. In *Proc. IJCAI-2015*, 1666–1672.

Roy, B. 1996. *Multicriteria methodology for decision analysis*. Kluwer Academic Press.

White, C.; Sage, A. P.; and Dozono, S. 1984. A model of multiattribute decision-making and trade-off weight determination under uncertainty. *IEEE Transactions on Systems, Man, and Cybernetics* 14(2):223–229.

Wilson, N., and Marinescu, R. 2012. An axiomatic framework for influence diagram computation with partially ordered utilities. In *Proc. KR-2012*, 210–220.

Wilson, N., and O'Mahony, C. 2011. The relationships between qualitative notions of optimality for decision making under logical uncertainty. In *Proc. AICS-2011*.

Wilson, N.; Razak, A.; and Marinescu, R. 2015a. Computing possibly optimal solutions for multi-objective constraint optimisation with tradeoffs. In *Proc. IJCAI-2015*.

Wilson, N.; Razak, A.; and Marinescu, R. 2015b. *Computing Possibly Optimal Solutions for Multi-Objective Constraint Optimisation with Tradeoffs (extended version of IJCAI-15 paper available online)*. http://ucc.insight-centre.org/nwilson/POIJCAI2015longer.pdf.

Xia, L., and Conitzer, V. 2011. Determining possible and necessary winners under common voting rules given partial orders. *Journal of Artificial Intelligence Research* 41(1):25–67.