Algorithm 2: \((\epsilon, \lambda)-\text{COVERING}(U)\)

1. \(\Gamma \leftarrow \emptyset; \mathcal{V} \leftarrow \emptyset;\)
2. foreach \(\vec{u} \in U\) do
3. \(\text{if } \phi_\lambda(\vec{u}) \notin \Gamma \) then
4. remove from \(\Gamma\) all \(\phi_\lambda(\vec{v})\) such that \(\phi_\lambda(\vec{u}) \geq \phi_\lambda(\vec{v})\);
5. \(\Gamma \leftarrow \Gamma \cup \{\phi_\lambda(\vec{u})\};\)
6. foreach \(\phi_\lambda(\vec{u}) \in \Gamma\) do \(\mathcal{V} \leftarrow \mathcal{V} \cup \{\vec{u}\};\)
7. return \(\mathcal{V}\);

The set \(U\) is not unique. However, it is possible to compute an \(\epsilon\)-covering of a finite set \(U \subseteq \mathbb{R}_+^p\) by mapping each vector \(\vec{u} \in U\) onto a hyper-grid using the log transformation \(\phi : \mathbb{R}_+^p \rightarrow \mathbb{Z}_+^p\), defined by \(\phi(\vec{u}) = (\phi(u_1), \ldots, \phi(u_p))\) where \(\forall i, \phi(u_i) = \lceil \log u_i / \log(1+\epsilon) \rceil\) [8]. By definition, we have that:

**Proposition 2** \(\forall \vec{u}, \vec{v} \in \mathbb{R}_+^p, \phi(\vec{u}) \geq \phi(\vec{v}) \Rightarrow \vec{u} \geq_\epsilon \vec{v}\).

It is easy to see that any cell of the grid represents a different class of vectors having the same image through \(\phi\). Based on Proposition 2, any vector belonging to a given cell \(\epsilon\)-dominates any other vector of that cell. Therefore, for any finite \(\epsilon\), we can obtain a valid \(\epsilon\)-covering of \(U\) by choosing one representative element in each cell and by keeping only dominated cells occupied. For example, if we restrict the vectors in \(U\) to be bounded by: \(1 \leq u_i \leq B\) for all \(i \in \{1, \ldots, p\}\), then the size of \(U\), is polynomial in \(\log B\) and \(1/\epsilon\) (see also [8] for more details).

**Example 3** Let \(U = \{\vec{u}, \vec{v}\}\) such that \(\vec{u} = (3.1, 2.9)\) and \(\vec{v} = (3, 3.05)\). Clearly, \(\vec{u} \npreceq \vec{v}\) and \(\vec{v} \npreceq \vec{u}\). Set \(\epsilon = 0.1\). We have that \(\phi(\vec{u}) = \phi(\vec{v}) = (12, 12)\), and it is easy to verify that \(\vec{u} \succeq_\epsilon \vec{v}\) and \(\vec{v} \succeq_\epsilon \vec{u}\). Therefore, \(U = \{\vec{u}\}\) is a valid \(\epsilon\)-covering of \(U\), as is \(\{\vec{v}\}\).

We next extend algorithm ELIM-MOID to compute an \(\epsilon\)-covering of the expected utility set \(\{EU_\lambda : \text{policies } \Delta\}\). However, it is not possible to just replace Pareto dominance with \(\epsilon\)-dominance at each variable elimination step and still guarantee a valid \(\epsilon\)-covering because \(\epsilon\)-dominance is not a transitive relation (e.g., if \(\vec{u} \geq_\epsilon \vec{v}\) and \(\vec{v} \geq_\epsilon \vec{w}\), we only have that \(\vec{u} \geq (1+\epsilon)^2 \cdot \vec{w}\)). To overcome this difficulty, we use a finer dominance relation, defined as follows [9].

**Definition 5** \(((\epsilon, \lambda)\)-dominance) For any finite \(\epsilon > 0\) and \(\lambda \in (0,1)\), the \((\epsilon, \lambda)\)-dominance relation is defined on positive vectors of \(\mathbb{R}_+^p\) by \(\vec{u} \succeq^\epsilon\lambda\vec{v} \iff (1+\epsilon)^\lambda \cdot \vec{u} \geq_\epsilon \vec{v}\).

Given a set \(U \subseteq \mathbb{R}_+^p\), a subset \(U_{\epsilon, \lambda} \subseteq U\) is called an \((\epsilon, \lambda)\)-covering, if \(\forall \vec{v} \in U \exists \vec{u} \in U_{\epsilon, \lambda}\) such that \(\vec{u} \succeq^\epsilon\lambda\vec{v}\).

Algorithm 2 computes an \((\epsilon, \lambda)\)-covering of a finite set \(U \subseteq \mathbb{R}_+^p\) by using the log grid mapping \(\phi_\lambda : \mathbb{R}_+^p \rightarrow \mathbb{Z}_+^p\) defined by \(\phi_\lambda(\vec{u}) = (\phi_\lambda(u_1), \ldots, \phi_\lambda(u_p))\) where \(\forall i, \phi_\lambda(u_i) = \lceil \log u_i / \log(1+\epsilon) \lambda \rceil\). It is easy to see that:

**Proposition 3** \(\forall \vec{u}, \vec{v} \in \mathbb{R}_+^p, \phi_\lambda(\vec{u}) \geq \phi_\lambda(\vec{v}) \Rightarrow \vec{u} \geq^\epsilon\lambda\vec{v}\).

**Proposition 4** Let \(\vec{u}, \vec{v}, \vec{w} \in \mathbb{R}_+^p\) and \(\lambda, \lambda' \in (0,1)\). The following properties hold: (i) if \(\vec{u} \geq^\epsilon\lambda\vec{v}\) then \(\vec{u} + \vec{w} \geq_{\epsilon}^\lambda\vec{v} + \vec{w}\), and if \(\vec{u} \geq^\epsilon\lambda\vec{v}\) and \(q \geq 0\) then \(q \cdot \vec{u} \geq^\epsilon\lambda q \cdot \vec{v}\); (ii) if \(\vec{u} \geq^\epsilon\lambda\vec{v}\) and \(\vec{v} \geq^\epsilon\lambda\vec{w}\) then \(\vec{u} \geq_{\epsilon}^{\lambda + \lambda'}\vec{w}\).

We define operations \(\max^\epsilon\) and \(\max^+\) on finite sets of utility values by \(\max^\epsilon(U, V) = \max_{\epsilon}(U \cup V)\) and \(U + \epsilon V = \max_{\epsilon}(U + V)\), where \(\max_{\epsilon}(U)\) is an \((\epsilon, \lambda)\)-covering of the finite set \(U \subseteq \mathbb{R}_+^p\) (computed using Algorithm 2).

The algorithm called ELIM-MOID\(_\epsilon\), which computes an \(\epsilon\)-covering of the expected utility set, is obtained from Algorithm 1 by replacing \(\max\) and \(\sum\) by \(\max^\epsilon\) and \(\sum^+\), respectively. Since the top-down phase of the algorithm consists of \(t\) elimination steps, one for each variable, and therefore requires computing \(t\) \((\epsilon, \lambda)\)-coverings via \(\max_{\epsilon}\), \(\lambda = 1, \ldots, t\), a sufficient condition to obtain a valid \(\epsilon\)-covering is to choose the \(\lambda\) values summing to 1, specifically \(\lambda_i = 1/t\), where \(t\) is the number of variables. Thus:

**Theorem 2** Given a MOID instance \((X, D, P, U)\) with \(t\) variables, \(p > 1\) objectives and any finite \(\epsilon > 0\), algorithm ELIM-MOID\(_\epsilon\) computes an \(\epsilon\)-covering.

The time and space complexity of ELIM-MOID\(_\epsilon\) is also bounded exponentially by the induced width of the legal elimination ordering. However, the size of \(\epsilon\)-covering generated can be in many cases significantly smaller than the corresponding Pareto set, as we will see in Section 6.

## 5 HANDLING IMPRECISE TRADEOFFS

The Pareto ordering is rather weak (often leading to very large Pareto optimal sets, as mentioned above in Section 4, and illustrated by the experiments in Section 6). Very often the decision maker will be happy to allow some trade-offs between objectives. For example, in a two-objective situation, they may tell us that they are happy to gain 3 units of the first objective at the cost of losing one unit of the second, and hence prefer \((3, -1)\) to \((0, 0)\). Such tradeoffs may be elicited using some structured method, or in a more \textit{ad hoc} way. For instance, in Example 1, the decision maker may be asked to imagine a scenario where it was known that the oil content was "wet", and whether they would then have a preference for drilling. To do so would imply a preference of \((50, 12)\) over \((0, 0)\).

We thus consider some set \(\Theta\) of vector pairs of the form \((\vec{u}, \vec{v})\), where \(\vec{u}, \vec{v} \in \mathbb{R}_+^p\). The idea is that this consists of elicited preferences of the decision maker. We say that binary relation \(\succ\) on \(\mathbb{R}_+^p\) extends \(\Theta\) if \(\vec{u} \succ \vec{v}\) for all \((\vec{u}, \vec{v}) \in \Theta\). Similarly, we say that \(\succ\) extends Pareto if \(\vec{u} \succeq \vec{v} \Rightarrow \vec{u} \not\succ \vec{v}\).

We consider that the decision maker has a partial order \(\succ\) over \(\mathbb{R}_+^p\), and that they specify a set of preferences \(\Theta\). We assume the Scale-Invariance and Independence properties hold (see Section 2), and, naturally, assume that \(\succ\) extends