Pareto. The input preferences $\Theta$ could contradict the other assumptions we make. We say that $\Theta$ is consistent if there exists some partial order $\succ$ that extends $\Theta$, extends Pareto, and satisfies Scale-Invariance and Independence.

The input preferences $\Theta$ (if consistent) give rise to a relation $\succeq_\Theta$ which specifies the deduced preferences. We say that $(\vec{u}, \vec{v})$ can be deduced from $\Theta$ if $\vec{u} \succeq \vec{v}$ holds for all partial orders $\succ$ that extend $\Theta$, extend Pareto, and satisfy Scale-Invariance and Independence. In this case we write $\vec{u} \succeq_\Theta \vec{v}$. The definition easily implies the following.

**Proposition 5** If $\Theta$ is consistent then $\succeq_\Theta$ is a partial order extending $\Theta$ and Pareto, and satisfying Scale-Invariance and Independence.

Proposition 5 shows that this dominance relation $\succeq_\Theta$ satisfies Scale-Invariance and Independence, giving the properties (Theorem 1) we need for the variable elimination algorithm to be correct (up to equivalence).

In Example 1, suppose now we have the additional user preference of $(50, 12)$ over $(0, 0)$, and hence include the pair $((50, 12), (0, 0))$ in $\Theta$. This would then imply that $(11, 12, 78)$ is dominated w.r.t. $\succeq_\Theta$ by $(20, 14.2)$.

Theorem 3 below gives a characterization of the partial order $\succeq_\Theta$, which we use as the basis of our implemented algorithm for testing this kind of dominance. Let $W$ be some subset of $\mathbb{R}^P$. Define $C(W)$, the convex cone generated by $W$, to be the set of all vectors $\vec{u}$ such that there exists $k \geq 0$ and non-negative real scalars $q_1, \ldots, q_k$ and $\vec{w}_i \in W$ with $\vec{u} \geq \sum_{i=1}^k q_i \vec{w}_i$, where $\geq$ is the weak Pareto relation (and an empty summation is taken to be equal to 0). $C(W)$ is the set of vectors that weakly-Pareto dominate some (finite) positive linear combination of elements of $W$.

**Theorem 3** Let $\Theta$ be a consistent set of pairs of vectors in $\mathbb{R}^P$. Then $\vec{u} \succeq_\Theta \vec{v}$ if and only if $\vec{u} - \vec{v} \in C(\vec{u}_i - \vec{v}_i : (\vec{u}_i, \vec{v}_i) \in \Theta)$.

Write finite set of input preferences $\Theta$ as $\{(\vec{u}_i, \vec{v}_i) : i = 1, \ldots, k\}$. Theorem 3 shows that, to perform the dominance test $\vec{u} \succeq_\Theta \vec{v}$, it is sufficient to check if there exist for $i = 1, \ldots, k$, non-negative real scalars $q_i$ such that $\vec{u} - \vec{v} \geq \sum_{i=1}^k q_i (\vec{u}_i - \vec{v}_i)$. This can be determined using a linear programming solver, since it amounts to testing if a finite set of linear inequalities is satisfiable.

**Example 4** Consider $\Theta = \{(-1, 2, -1), (4, -3, 0)\}$ and vectors $\vec{u} = (1, -1, 0)$ and $\vec{v} = (0, -2, 1)$. Then $\vec{u} \succeq_\Theta \vec{v}$ iff $\vec{u} - \vec{v}$ weak Pareto-dominates a non-negative combination of elements of $\Theta$, i.e., $3q_1 \geq 0$, $q_2 \geq 0$ such that $\vec{u} - \vec{v} \geq q_1 (-1, 2, -1) + q_2 (4, -3, 0)$, which is iff there exists a solution for the linear system defined by:

$$
1 \geq -q_1 + 4q_2 \\
1 \geq 2q_1 - 3q_2 \\
-1 \geq -q_1
$$

Since this is the case (e.g., $q_1 = 1$; $q_2 = 0.5$) we have $\vec{u} \succeq_\Theta \vec{v}$.

Alternatively, we can use the fact that the dominance test is checking whether $\vec{u} - \vec{v}$ is in the convex cone generated by $\{\vec{u}_i - \vec{v}_i : i = 1, \ldots, k\}$ plus the $p$ unit vectors in $\mathbb{R}^P$. We made use of an (incomplete) algorithm [27] for this purpose (which computes the distance of a vector from a cone).

Therefore, the algorithm called ELIM-MOID-ToF that exploits tradeoffs is obtained from Algorithm 1, by replacing the $+$ and max operators with $+$ and max, respectively, where max$^\Theta(U, V) = \max_{\succeq_\Theta} (U \cup V)$, $U +^\Theta V = \max_{\succeq_\Theta} (U + V)$, and max$^\Theta(U)$ is the set of undominated elements of finite set $U \subseteq \mathbb{R}^P$ with respect to $\succeq_\Theta$.

Instead of eliminating $\succeq_\Theta$-dominated utility values during the computation, one could generate the Pareto optimal set of expected utility values, and only then eliminate $\succeq_\Theta$-dominated values. However, the experimental results in Section 6 (Table 2) indicate that this will typically be much less computationally efficient.

**6 EXPERIMENTS**

In this section, we evaluate empirically the performance of the proposed variable elimination algorithms on random multi-objective influence diagrams. All experiments were run on a 2.6GHz quad-core processor with 4GB of RAM.

The algorithms considered were implemented in C++ (32-bit) and are denoted by ELIM-MOID (Section 3), ELIM-MOID$_0$ (Section 4) and ELIM-MOID-ToF (Section 5), respectively. We implemented both methods for performing the $\succeq_\Theta$-dominance, namely the linear programming and the distance from a cone based one, and report only on the former because their performance was comparable overall.

We experimented with a class of random influence diagrams described by the parameters $(C, D, k, p, r, a, O)$, where $C$ is the number of chance variables, $D$ is the number of decision variables, $k$ is the maximum domain size, $p$ is the number of parents in the graph for each variable, $r$ is the number of root nodes, $a$ is the arity of the utility functions and $O$ is the number of objectives. The structure of the influence diagram is created by randomly picking $C + D - r$ variables out of $C + D$ and, for each, selecting $p$ parents from their preceding variables, relative to some ordering, whilst ensuring that the decision variables are connected by a directed path. We then added to the graph $D$ utility nodes, each one having $a$ parents picked randomly from the chance and decision variables.

We generated random problems with parameters $k = 2$, $p = 2$, $r = 5$, $a = 3$ and varied $C \in \{15, 25, 35, 45, 55\}$, $D \in \{5, 10\}$ and $O \in \{2, 3, 5\}$, respectively. In each case, 25% of the chance nodes were assigned deterministic CPTs (containing 0 and 1 entries). The remaining CPTs were randomly filled using a uniform distribution. The utility vectors were generated randomly, each objective value being