ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230572

Enforcing Natural Properties of Choice Functions, with Application for Combination

Nic Wilson *

Insight Centre, School CS and IT, University College Cork, Ireland
ORCiD ID: Nic Wilson https://orcid.org/0000-0003-1874-8255

Abstract. One important and natural representation of preferences is a choice function, which returns the preferred options amongst any given subset of the alternatives. There are some very intuitive coherence conditions that might be assumed for an agent's choice function, in particular path independence, and a consistency condition stating that there is always at least one preferred alternative among any non-empty set. However, an elicited choice function may not satisfy path independence, because of the elicitation being incomplete, or because of there being some incoherence in the agent's reported choice function (despite the agent assenting to the general coherence conditions). Furthermore, if we wish to combine the choice functions of more than one agent, simple natural combination operations can lose path independence. This paper develops methods for enforcing path independence and restoring consistency, thus, making the user preferences coherent; this method also leads to approaches for combining two choice functions, in order to suggest the most promising alternatives for a pair of agents.

1 Introduction

Reasoning with agent preferences is important for many AI decision support systems, in order to help find choices that are favoured by a decision maker. One important and natural representation of preferences is a choice function, which returns the preferred options amongst any given subset of the alternatives.

There are some very intuitive coherence conditions that might be assumed for an agent's choice function, in particular path independence [19, 11], and a consistency condition stating that there is always at least one preferred alternative among any non-empty set. Path independence means that if A' is the set of optimal elements of set of alternatives A then $A' \cup B$ has the same optimal elements as $A \cup B$. As well as being a very intuitive property for a choice function, it can be very helpful computationally since it allows incremental computational of the set of preferred alternatives. However, an elicited choice function may not satisfy path independence, because of the elicitation being incomplete, or because of there being some incoherence in the agent's reported choice function (despite the agent assenting to the general coherence conditions). Furthermore, if we wish to combine the choice functions of more than one agent, simple natural combination operations can lose path independence and the consistency condition.

This paper develops methods for enforcing path independence and restoring consistency, thus, making the user preferences coherent; this method also leads to approaches for combining two choice functions, in order to suggest the most promising alternatives for a pair of agents.

Path independence is equivalent to a pair of well-known conditions, which have been called *Heritage* and *Outcast* [1, 11], and have been considered in many classic works on choice functions. We show that path independence can be enforced by first enforcing Heritage and then enforcing Outcast; furthermore, if the input choice function already satisfies Heritage, one can efficiently compute the best alternatives in a given set, according to the enforced choice function.

We show further that there is a simple way of restoring consistency that maintains path independence, and that pre-processing the choice function by restoring consistency can lead to a stronger choice function.

Section 2 gives the basic definitions and considers important properties that one might expect of a choice function; we also discuss how path independent choice functions naturally arise in a preference elicitation context. The basic combination operations of union, intersection and composition are considered in Section 3, and we analyse which of the properties of choice functions are maintained by the basic combination operations. Section 4 shows how path independence can be enforced, and Section 5 discusses different approaches for restoring consistency. Section 6 concludes.

2 Desirable Properties of Choice Functions

We start with definitions relating to choice functions, and then consider classic desirable properties. We discuss how path independent choice functions arise naturally in a context when there is only partial knowledge about a decision maker's preferences.

2.1 Basic Definitions of Choice Functions

Let Ω be a finite set, which is intended to represent a set of alternatives, i.e., alternative choices in a decision making problem. We define a *Choice Function (CF)* Op *over* Ω to be a function from 2^Ω to 2^Ω satisfying the following contraction property:

(Sub): for all $A \subseteq \Omega$, $\mathrm{Op}(A) \subseteq A$.

In this paper, the main intended interpretation of a choice function Op will be that it represents (what we know about) the preferred alternatives (e.g., of some agent) in a particular decision making problem. For set of alternatives $A \subseteq \Omega$, the set $\mathrm{Op}(A)$ represents the set of optimal (i.e., best) alternatives among A (so Op is short for *Optimal*).

* Email: nic.wilson@insight-centre.org.

Given the choice function Op over Ω, we will also consider the complementary function $\overline{\mathrm{Op}}$ over Ω, given by $\overline{\mathrm{Op}}(A) = A \setminus \mathrm{Op}(A)$ for $A \subseteq \Omega$. If $\mathrm{Op}(A)$ represents the optimal elements in A, then $\overline{\mathrm{Op}}(A)$ is the set of sub-optimal elements.

We define the identity function Id on 2^Ω by $\mathrm{Id}(A) = A$ for all $A \subseteq \Omega$. This choice function can be considered as the vacuous choice function over Ω, since, it eliminates no alternative of any A: no alternative is sub-optimal. For mathematical reasons, it is also helpful to consider the null choice function Emp on 2^Ω defined by $\mathrm{Emp}(A) = \emptyset$ for all $A \subseteq \Omega$ (where Emp is short for *empty set*). This is a choice function that always eliminates all the alternatives.

We define the fixed points *Fix*(Op) of choice function Op to be $\{A \subseteq \Omega : \mathrm{Op}(A) = A\}$. These are the sets for which Op is equal to the vacuous CF. If $A \in \mathit{Fix}(\mathrm{Op})$ then Op is, in a sense, uninformative about A: none of elements of A are preferred to the others.

2.2 Properties and their Relationships

Not all choice functions represent sensible decision making attitudes, and there are natural properties that one might assume on a choice function. The main properties of choice functions (over Ω), that we focus on, are the following important and well known rationality requirements (where $A, B \subseteq \Omega$):

(**NE**) *(Non-emptiness)*: For non-empty A, $\mathrm{Op}(A) \neq \emptyset$.
(**H**) *(Heritage)*: If $B \subseteq A$ then $\mathrm{Op}(A) \cap B \subseteq \mathrm{Op}(B)$.
(**O**) *(Outcast)*: If $\mathrm{Op}(A) \subseteq B \subseteq A$ then $\mathrm{Op}(A) = \mathrm{Op}(B)$.

Property (**NE**) (being non-empty) might be viewed as a kind of consistency requirement: that at least one alternative is optimal (i.e., not excluded). Mostly in the social choice literature, a (social) choice function is defined to be a function 2^Ω to 2^Ω satisfying *(Sub)* and property (**NE**). We consider functions not satisfying (**NE**) for a number of reasons; in particular, for mathematical convenience, because this looser definition makes the set of choice functions closed under intersection, and because certain natural notions of optimality, such as being necessarily optimal, and being possibly strictly optimal [26] do not always satisfy (**NE**).

The Heritage property (**H**) implies that optimal alternatives from a larger set are inherited by a smaller set. In terms of the corresponding sub-optimality function, it has an even simpler form:

If $B \subseteq A$ then $\overline{\mathrm{Op}}(B) \subseteq \overline{\mathrm{Op}}(A)$.

That is, an element is sub-optimal in A if it is sub-optimal in a subset of A.

In the Outcast property (**O**), letting C be $A \setminus B$ we obtain the following equivalent form:

If $C \subseteq A$ and $\mathrm{Op}(A) \cap C = \emptyset$ then $\mathrm{Op}(A) = \mathrm{Op}(A \setminus C)$.

That is, if every element of C is sub-optimal in A, then deleting C from A does not change the optimal elements. Thus, (**O**) is a form of *independence of irrelevant alternatives* [15, 9].

A choice function Op over Ω is said to be *path independent* [19], and a *Plott function* [11], if it satisfies $\mathrm{Op}(A \cup B) = \mathrm{Op}(\mathrm{Op}(A) \cup B)$ for any $A, B \subseteq \Omega$. This property, which is very helpful computationally [27], holds if and only if Op satisfies (**H**) and (**O**) [1, 11], so is also desirable from a semantics perspective. We say that Op is a *consistent Plott function* if it also satisfies the non-empty property (**NE**).

These properties have been explored a good deal in the literature, see Moulin's survey article [17], and e.g., [1, 7]. The names Heritage (or Heredity) (**H**) and Outcast (**O**) come from Aizerman [1] via [11]. Both properties appear as postulates in Chernoff [10]. Property (**H**) also corresponds with Sen's Condition α [20], and Moulin's Chernoff Condition [17]. Property (**O**) was used in Nash [18] and relates with Moulin's Aizerman property [17].

Both (**H**) and (**O**) imply idempotence: $\mathrm{Op}(\mathrm{Op}(A)) = \mathrm{Op}(A)$. We also consider a stronger version of idempotence, that is still weaker than (**H**):

(**SIdem**): Strong Idempotence: If $A \subseteq \mathrm{Op}(B)$ then $\mathrm{Op}(A) = A$.

2.3 Choice Functions from Elicitation

A choice function over Ω will be a very large object to represent explicitly, unless Ω is very small. Fortunately, there are a variety of ways of compactly representing choice functions, in particular, if the choice function is derived from elicited preferences of a decision maker (with the elicited preferences being, by their very nature, limited in size). In this section we consider two different kinds of compact representation.

Preference elicitation will often lead to a partial representation of the user preferences, an upper bound on the agent's choice function. Various natural preference statements can be expressed in the form $\mathrm{Op}(A) \subseteq B$, i.e., that all optimal elements of A are in B. If we elicit a set of such statements, this gives rise to a function Op' defined on a set $\mathcal{L}$ of subsets of Ω, where, for $A \in \mathcal{L}$, $\mathrm{Op}'(A)$ is assumed to contain the user's preferred/optimal subset of A. The function Op' can be considered as trivially extended to the whole of 2^Ω by $\mathrm{Op}'(A) = A$ for all $A \notin \mathcal{L}$; this defines a choice function over Ω, which is an upper bound on the user's true choice function, and which is compactly represented by Op' on $\mathcal{L}$ (since, at least for moderately sized Ω, 2^Ω will be huge, and $\mathcal{L}$, being based on elicited preferences, will be typically very much smaller).

This upper bound function Op' will usually not satisfy intuitive properties, such as Heritage and Outcast. It can therefore be desirable to enforce such properties; we discuss how one can do this in Section 4 below.

Plott functions generated from partial information

As well as enforcing path independence, there are also other ways of generating Plott functions from elicited user preferences. Let us suppose that an agent's true preferences are a total pre-order over Ω. Very often our information about someone's preferences is incomplete. So suppose that we have, through elicitation, some reliable information restricting this set of relations, and thus a non-empty set $\mathcal{X}$ of total pre-orders on Ω. Given $\succcurlyeq$ in $\mathcal{X}$, and $A \subseteq \Omega$, we say that α *is optimal in* A, written $\alpha \in \mathrm{O}_{\succcurlyeq}(A)$, if $\alpha \in A$ and for all $\beta \in A$, $\alpha \succcurlyeq \beta$. We say that α is *possibly optimal in* A, written $\alpha \in \mathrm{PO}_{\mathcal{X}}(A)$, if there exists $\succcurlyeq$ in $\mathcal{X}$ such that $\alpha \in \mathrm{O}_{\succcurlyeq}(A)$. Thus, $\mathrm{PO}_{\mathcal{X}}(A) = \bigcup_{\succcurlyeq \in \mathcal{X}} \mathrm{O}_{\succcurlyeq}(A)$; this gives the set of alternatives that could still be optimal in A, given our current information.

Define the pre-order $\succcurlyeq_{\mathcal{X}}$ to be the intersection of all relations in $\mathcal{X}$, with $\alpha \succcurlyeq_{\mathcal{X}} \beta$ if and only if $\alpha \succcurlyeq \beta$ for all relations $\succcurlyeq$ in $\mathcal{X}$. Let $\succ_{\mathcal{X}}$ be the strict part of $\succcurlyeq_{\mathcal{X}}$, so that $\alpha \succ_{\mathcal{X}} \beta$ if and only if $\alpha \succcurlyeq_{\mathcal{X}} \beta$ and it is not the case that $\beta \succcurlyeq_{\mathcal{X}} \alpha$. We define $\mathrm{UD}_{\mathcal{X}}(A)$ to be the set of elements of A that are undominated (with respect to elements in A). Thus $\alpha \in \mathrm{UD}_{\mathcal{X}}(A)$ if and only if $\alpha \in A$ and there does not exist $\beta \in A$ with $\beta \succ_{\mathcal{X}} \alpha$.

It is well known that $\mathrm{PO}_{\mathcal{X}}$ and $\mathrm{UD}_{\mathcal{X}}$ are consistent Plott functions; indeed, any consistent Plott function is pseudo-rationalizable, i.e., expressible as $\mathrm{PO}_{\mathcal{X}}$ for some set of total orders $\mathcal{X}$ [1, 11]. Very often, $\mathrm{PO}_{\mathcal{X}}(A)$ is a smaller set than $\mathrm{UD}_{\mathcal{X}}$, and so is in a sense more informative. (We can also consider the intersection $\mathrm{POUD}_{\mathcal{X}}(A) =$

$\mathrm{PO}_{\mathcal{X}}(A) \cap \mathrm{UD}_{\mathcal{X}}(A)$, which is also always a consistent Plott function [27]).

In the choice function literature, there is a good deal of focus on rationalizable choice functions, that satisfy the property: for all $A, B \subseteq \Omega$, $\mathrm{Op}(A) \cap \mathrm{Op}(B) \subseteq \mathrm{Op}(A \cup B)$ (Concordance [1]; Expansion [17]). $\mathrm{UD}_{\mathcal{X}}$ is rationalizable, but $\mathrm{PO}_{\mathcal{X}}$ typically is not (because the Expansion property is not maintained by union). Therefore, non-rationalizable consistent Plott functions ($\mathrm{PO}_{\mathcal{X}}$) are important in preference elicitation methods, e.g., [12, 3, 4, 21].

Note that even though $\mathcal{X}$ may be a very large set, it will be compactly represented by the preference inputs. One common form, see e.g., [8, 22, 16, 27, 3, 23], is where it is assumed that the user preference model is linear, so that each alternative α has an associated vector $\overline{\alpha}$ in a multiple-objective space $I\!R^p$, and it is assumed that the user utility of α is of the form $w \cdot \overline{\alpha} = \sum_{i=1}^{p} w(i)\overline{\alpha}(i)$, for some unknown vector of weights $w \in \mathcal{X}$. Hence, vector w has an associated total pre-order $\geq_w$ over alternatives given by $\alpha \geq_w \beta$ if and only if $w \cdot \overline{\alpha} \geq w \cdot \overline{\beta}$. A convex polytope $\mathcal{Y}$ is defined by linear inequalities arising from preference inputs; for example a preference of alternative γ over alternative δ, expressed by the user, corresponds to the constraint $w \cdot (\overline{\gamma} - \overline{\delta}) \geq 0$. Let $\mathcal{X}$ be the set of total pre-orders $\{\geq_w : w \in \mathcal{Y}\}$. The preference inputs thus generate the Plott function $\mathrm{PO}_{\mathcal{X}}$. Then, for $\alpha \in A \subseteq \Omega$, one can efficiently determine if $\alpha \in \mathrm{PO}_{\mathcal{X}}(A)$ by checking the consistency of the set of linear inequalities [$w \in \mathcal{Y}$ and $w \cdot (\overline{\alpha} - \overline{\beta}) \geq 0$ for all $\beta \in A$].

3 Combinations

The most straightforward ways of combining choice functions are based on: (i) taking the intersections of the sets; (ii) taking the union of the sets; and (iii) applying one choice function and then the other. In this section we explore some basic properties of these simple combination operations.

Union and intersection of choice functions: We can define the union and intersection of choice functions Op_1 and Op_2 (over Ω) in the obvious way ('pointwise'). We define $\mathrm{Op}_1 \cup \mathrm{Op}_2$ and $\mathrm{Op}_1 \cap \mathrm{Op}_2$ by $(\mathrm{Op}_1 \cup \mathrm{Op}_2)(A) = \mathrm{Op}_1(A) \cup \mathrm{Op}_2(A)$ and $(\mathrm{Op}_1 \cap \mathrm{Op}_2)(A) = \mathrm{Op}_1(A) \cap \mathrm{Op}_2(A)$ for each $A \subseteq \Omega$.

Intersection seems a rather intuitive way of combining choice functions: especially when the individual choice functions are relatively indecisive; it involves choosing options that are preferred by both parties. Union, on the hand, seems rather cautious, with the returned choices being the alternatives that are preferred by either agent.

We can also extend the subset relation to choice functions, by applying it for each subset of Ω. Thus, $\mathrm{Op}_1 \subseteq \mathrm{Op}_2$ means for all $A \subseteq \Omega$, $\mathrm{Op}_1(A) \subseteq \mathrm{Op}_2(A)$. We say that Op_1 then *strengthens* Op_2, because Op_1 gives a stronger (or at least as strong) result than Op_2, i.e., for each set A, Op_1 finds as least as many elements as Op_2 to be suboptimal. We have $\mathrm{Op}_1 \subseteq \mathrm{Op}_2 \iff \mathrm{Op}_1 \cup \mathrm{Op}_2 = \mathrm{Op}_2 \iff \mathrm{Op}_1 \cap \mathrm{Op}_2 = \mathrm{Op}_1$.

Composition of choice functions: For choice functions Op_1 and Op_2 over Ω we define the composition choice function $\mathrm{Op}_2 \circ \mathrm{Op}_1$ (meaning Op_1 followed by Op_2) by $(\mathrm{Op}_2 \circ \mathrm{Op}_1)(A) = \mathrm{Op}_2(\mathrm{Op}_1(A))$ for each $A \subseteq \Omega$.

Work on combining preference information using priority includes, for instance, a general framework for combining preference relations using priority [2], voting rules based on sequential elimination of alternatives [6]; and a computational technique for preference inference based on composition of lexicographic orders [25].

3.1 *Relationships Between Intersection and Composition*

The intersection and the two compositions are often very different from each other. However, they do have the same set of fixed points, which is equal to the intersection of the fixed points of the choice functions. (This doesn't require any additional assumption on the choice functions.)

Proposition 1 *Consider choice functions* Op_1 *and* Op_2 *over* Ω. *Then Fix*$(\mathrm{Op}_1 \circ \mathrm{Op}_2)$ = *Fix*$(\mathrm{Op}_2 \circ \mathrm{Op}_1)$ = *Fix*$(\mathrm{Op}_1 \cap \mathrm{Op}_2)$ = *Fix*$(\mathrm{Op}_1) \cap$ *Fix*(Op_2). [1]

In the composition $\mathrm{Op}_2 \circ \mathrm{Op}_1$, choice function Op_1 is given priority over Op_2, since the first step is to eliminate alternatives with Op_1. To restore symmetry between the two choice functions, one can consider the union between the compositions $(\mathrm{Op}_1 \circ \mathrm{Op}_2) \cup (\mathrm{Op}_2 \circ \mathrm{Op}_1)$, and the intersection $(\mathrm{Op}_1 \circ \mathrm{Op}_2) \cap (\mathrm{Op}_2 \circ \mathrm{Op}_1)$; however, the latter is not so interesting, since it equals the intersection $\mathrm{Op}_1 \cap \mathrm{Op}_2$ if Op_1 and Op_2 satisfy (**H**).

Proposition 2 *When choice functions* Op_1 *and* Op_2 *over* Ω *satisfy property (**H**) then* $\mathrm{Op}_1 \cap \mathrm{Op}_2 = (\mathrm{Op}_1 \circ \mathrm{Op}_2) \cap (\mathrm{Op}_2 \circ \mathrm{Op}_1)$, *i.e., for all* $A \subseteq \Omega$,

$$\mathrm{Op}_1(A) \cap \mathrm{Op}_2(A) = \mathrm{Op}_1(\mathrm{Op}_2(A)) \cap \mathrm{Op}_2(\mathrm{Op}_1(A)).$$

For choice functions satisfying property (**H**), Proposition 2 implies that Op_1 and Op_2 commute (i.e., the two compositions are equal) if and only if both compositions are equal to the intersection. We now explore some situations when the two choice functions commute. Proposition 3 shows that this happens when the intersection satisfies property (**O**). Proposition 4 gives another sufficient condition for the choice functions to commute.

Proposition 3 *Let* Op_1 *and* Op_2 *be choice functions over* Ω, *and suppose that their intersection satisfies property (**O**) and that* Op_1 *is idempotent. Then* $\mathrm{Op}_2 \circ \mathrm{Op}_1 = \mathrm{Op}_1 \cap \mathrm{Op}_2$. *Thus, if* Op_2 *is also idempotent then* Op_1 *and* Op_2 *commute:* $\mathrm{Op}_1 \circ \mathrm{Op}_2 = \mathrm{Op}_2 \circ \mathrm{Op}_1$.

An example of this is when, for some $\mathcal{X}$, $\mathrm{Op}_1 = \mathrm{PO}_{\mathcal{X}}$ and $\mathrm{Op}_2 = \mathrm{UD}_{\mathcal{X}}$ (see Section 2.3) The intersection $\mathrm{POUD}_{\mathcal{X}}$ satisfies property (**O**) which implies, by this proposition, that choice functions $\mathrm{PO}_{\mathcal{X}}$ and $\mathrm{UD}_{\mathcal{X}}$ commute.

Proposition 4 *Let* Op_1 *and* Op_2 *be Plott functions over* Ω, *and assume that for all* $A \subseteq \Omega$, *either* $\mathrm{Op}_1(A) \subseteq \mathrm{Op}_2(A)$ *or* $\mathrm{Op}_1(A) \supseteq \mathrm{Op}_2(A)$. *Then* $\mathrm{Op}_1 \circ \mathrm{Op}_2 = \mathrm{Op}_2 \circ \mathrm{Op}_1 = \mathrm{Op}_1 \cap \mathrm{Op}_2$.

3.2 *Combinations maintaining properties*

We will consider which of the properties described in Section 2.2 are maintained by these different combinations. Formally, a property P on choice functions is *maintained by a combination operation* $\otimes$ (on choice functions) if for every finite Ω and every choice functions Op_1 and Op_2 over Ω that satisfy property P, we have that $\mathrm{Op}_1 \otimes \mathrm{Op}_2$ satisfies P. The parts of (1) and (2) not involving strong idempotence are well-known (see e.g., [5, 11]).

[1] Proofs of all the results can be found in the longer version of the paper, available online [24].

Theorem 1 *1. Union maintains properties (**NE**), (**H**) and (**O**) but not strong idempotence.*

*2. Intersection maintains (**H**) and strong idempotence but not (**NE**) or (**O**).*

*3. Composition maintains (**NE**) and strong idempotence, but not (**H**) or (**O**).*

*4. Union of composition maintains (**NE**), but not strong idempotence or (**H**) or (**O**).*

Examples showing properties not being maintained

In the examples below, we use only total orderings on $\Omega = \{a, b, c\}$, and abbreviate a total ordering such as (b, a, c) to just bac. For instance, we can consider a set of orderings $\mathcal{X} = \{abc, cab\}$, and the associated choice function based on possibly optimal alternatives, $\mathrm{PO}_{\{abc,cab\}}$ (see Section 2.2). For non-empty subset A of $\{a, b, c\}$, $\mathrm{PO}_{\{abc,cab\}}(A)$ is the set containing the best element in A according to abc and the best element according to cab. Recall that for any $\mathcal{X}$, the choice function $\mathrm{PO}_{\mathcal{X}}$ satisfies the properties in Section 2.2, in particular, properties (**NE**), (**H**) and (**O**), i.e., is a consistent Plott function.

Now, a is the top element in the total order abc, and c is the top element in cab, so the set of possibly optimal elements in $\Omega = \{a, b, c\}$, i.e., $\mathrm{PO}_{\{abc,cab\}}(\{a, b, c\})$, is equal to $\{a, c\}$. If we are interested in $\{a, b\}$ then we consider the restrictions of the two orderings to this set, giving the set of orderings $\{ab, ab\} = \{ab\}$, so a is better than b in both orderings, and hence, $\mathrm{PO}_{\{abc,cab\}}(\{a, b\})$ equals $\{a\}$.

We will first construct an example that shows that composition does not maintain either (**H**) or (**O**), and that intersection does not maintain (**O**). Let Op_1 be $\mathrm{PO}_{\{abc,bca\}}$, and let Op_2 be $\mathrm{PO}_{\{abc,cab\}}$. For $i = 1, 2$, $\mathrm{Op}_i(A) = A$ for a singleton or empty set A. Otherwise we have values as follows:

$$
\begin{array}{ll}
\mathrm{Op}_1(\{a,b,c\}) = \{a,b\}; & \mathrm{Op}_2(\{a,b,c\}) = \{a,c\}; \\
\mathrm{Op}_1(\{a,b\}) = \{a,b\}; & \mathrm{Op}_2(\{a,b\}) = \{a\}; \\
\mathrm{Op}_1(\{a,c\}) = \{a,c\}; & \mathrm{Op}_2(\{a,c\}) = \{a,c\}; \\
\mathrm{Op}_1(\{b,c\}) = \{b\}; & \mathrm{Op}_2(\{b,c\}) = \{b,c\}.
\end{array}
$$

We write $\mathrm{Op}_{1\cap 2}$ for $\mathrm{Op}_1 \cap \mathrm{Op}_2$, and $\mathrm{Op}_{1\circ 2}$ for $\mathrm{Op}_1 \circ \mathrm{Op}_2$, and $\mathrm{Op}_{2\circ 1}$ for $\mathrm{Op}_2 \circ \mathrm{Op}_1$. From the table of values above we have $\mathrm{Op}_{2\circ 1}(\{a, b, c\}) = \mathrm{Op}_2(\{a, b\}) = \{a\}$. If $\mathrm{Op}_{2\circ 1}$ satisfied (**O**) then we would have $\mathrm{Op}_{2\circ 1}(\{a, c\}) = \{a\}$, because $\{a\} \subseteq \{a, c\} \subseteq \{a, b, c\}$. However, $\mathrm{Op}_{2\circ 1}(\{a, c\}) = \mathrm{Op}_2(\{a, c\}) = \{a, c\}$, showing that $\mathrm{Op}_{2\circ 1}$ does not satisfy (**O**). Since, Op_1 and Op_2 satisfy (**O**) this shows that composition does not maintain (**O**).

Similarly, $\mathrm{Op}_{1\cap 2}(\{a, b, c\}) = \{a, b\} \cap \{a, c\} = \{a\}$. And we have $\mathrm{Op}_{1\cap 2}(\{a, c\}) = \{a, c\} \cap \{a, c\} = \{a, c\}$, showing that $\mathrm{Op}_{1\cap 2}$ does not satisfy (**O**), and so intersection does not maintain (**O**). Also, we have that $\mathrm{Op}_{1\circ 2}(\{b, c\}) = \mathrm{Op}_1(\{b, c\}) = \{b\}$, and, $\mathrm{Op}_{1\circ 2}(\{a, b, c\}) = \mathrm{Op}_1(\{a, c\})$ which equals $\{a, c\}$, and thus, $\mathrm{Op}_{1\circ 2}(\{a, b, c\}) \cap \{b, c\} = \{c\} \not\subseteq \mathrm{Op}_{1\circ 2}(\{b, c\})$, which implies that $\mathrm{Op}_{1\circ 2}$ does not satisfy property (**H**). Hence, composition does not maintain (**H**).

In the longer version [24] we give an example that shows that intersection does not maintain (**NE**), and that union does not maintain strong idempotence, and that the union of compositions maintains neither strong idempotence nor (**H**).

4 Enforcing Properties

Properties (**H**), (**O**) and (**NE**) are very intuitive and important properties of a choice function. However, applying intersection or composition operations can mean that some of these properties can be lost, as shown in the last section. In particular, even if Op_1 and Op_2 are consistent Plott functions, we may have that their intersection $\mathrm{Op}_1 \cap \mathrm{Op}_2$ fails to satisfy either (**O**), or (**NE**), or both. It is thus desirable to be able to recover the properties, to obtain a coherent form of choice function. In addition, a partial specification of an agent's choice function, based on elicitation, can lack some of these properties, so again it is desirable to restore coherence.

4.1 Enforcing Through Maximal Strengthenings

Suppose that a choice function Op does not satisfy a desirable property P. One can attempt to enforce this property by changing Op to Op' that does satisfy P. Our focus here is on strengthening Op to make it satisfy P; in this way, if α is viewed as sub-optimal in a set A w.r.t. Op (i.e., $\alpha \notin \mathrm{Op}(A)$), then it will be suboptimal w.r.t. Op' ($\alpha \notin \mathrm{Op}'(A)$).

Definition 1 (P-enforcement for union-closed properties.) *Let P be a property on choice functions over Ω. We say that P is* union-closed *if it satisfies the following two properties:*

(a) union maintains P (i.e., if Op_1 and Op_2 satisfy P then so does $\mathrm{Op}_1 \cup \mathrm{Op}_2$); and

(b) the null choice function Emp satisfies P.

Consider any union-closed property P, and any choice function Op *over Ω. Define $\mathrm{Op}^{(P)}$, called the P-*enforcement of Op*, to be the union of all choice functions Op' such that (I) Op' satisfies P and (II) $\mathrm{Op}' \subseteq \mathrm{Op}$. (Since Emp satisfies P and Emp $\subseteq \mathrm{Op}$, there exists at least one such choice function Op'.)*

For union-closed P, the definition implies that $\mathrm{Op}^{(P)}$ is the unique maximal choice function Op' satisfying P and such that $\mathrm{Op}' \subseteq \mathrm{Op}$. In other words, $\mathrm{Op}^{(P)}$ is the (setwise) maximal (i.e., weakest) strengthening of Op that satisfies P. The result below gives some basic properties of P-enforcement, including that the operation $(\mathrm{Op})^{(P)}$ is idempotent, and is monotone with respect to both Op and P.

Proposition 5 *Suppose that properties P and P' on choice functions over Ω are both union-closed, and that P implies P'. Let* Op *and Op_1 be choice functions over Ω such that $\mathrm{Op} \subseteq \mathrm{Op}_1$. Then (i) $\mathrm{Op}^{(P)} \subseteq \mathrm{Op}$; (ii) $\mathrm{Op}^{(P)}$ satisfies P; (iii) if* Op *satisfies P then $\mathrm{Op}^{(P)} = \mathrm{Op}$, so P-enforcement is idempotent; in particular, $(\mathrm{Op}^{(P)})^{(P)} = \mathrm{Op}^{(P)}$; (iv) P-enforcement is monotone: $\mathrm{Op}^{(P)} \subseteq (\mathrm{Op}_1)^{(P)}$; (v) $\mathrm{Op}^{(P)} \subseteq \mathrm{Op}^{(P')}$ and (vi) $(\mathrm{Op}^{(P')})^{(P)} = (\mathrm{Op}^{(P)})^{(P')} = \mathrm{Op}^{(P)}$.*

Both properties (**H**) and (**O**) are union-closed (see Theorem 1) and so, their conjunction, path independence, is too; we can thus consider respective enforcements.

PI-enforcement of Op**:** For choice function Op over Ω, we write $\mathrm{Op}^{(\pi)}$ for the path independent enforcement (PI-enforcement) of Op. Thus, $\mathrm{Op}^{(\pi)}$ is the enforcement of Op under the conjunction of properties (**H**) and (**O**). PI-enforcement has been studied under the name *Plottization*: see Danilov and Koshevoy [11].

*4.2 Enforcing Property (**H**)*

There is a simple explicit formula for $\mathrm{Op}^{(\mathrm{H})}$, the result of enforcing property (**H**) on choice function Op. To understand $\mathrm{Op}^{(\mathrm{H})}$, it is helpful to consider property (**H**) in terms of the associated sub-optimality function $\overline{\mathrm{Op}}$: if $B \subseteq A$ then $\overline{\mathrm{Op}}(B) \subseteq \overline{\mathrm{Op}}(A)$. Based on this, one can see that $\overline{\mathrm{Op}}^{(\mathrm{H})}$, the associated sub-optimality function for $\mathrm{Op}^{(\mathrm{H})}$, is given by $\overline{\mathrm{Op}}^{(\mathrm{H})}(A) = \bigcup_{B\subseteq A} \overline{\mathrm{Op}}(B)$, which leads to the following result:

Proposition 6 *For choice function* Op *over* Ω*, and any* $A \subseteq \Omega$ *we have* $A \setminus \mathrm{Op}^{(H)}(A) = \overline{\mathrm{Op}}^{(H)}(A) = \bigcup_{B\subseteq A} \overline{\mathrm{Op}}(B)$*, and thus,* $\mathrm{Op}^{(H)}(A) = \bigcap_{B\subseteq A}(\mathrm{Op}(B) \cup (A \setminus B))$.

4.3 Enforcing Path Independence Given (*H*)

We say that C *encloses* A *[with respect to* Op*]* if $\mathrm{Op}(C) \subseteq A \subseteq C$, i.e., A is a subset of C that contains all the optimal elements of C. We also write this as $C \rhd_{\mathrm{Op}} A$. The Outcast axiom (**O**) states that if C encloses A then A and C have the same optimal elements.

Assume that choice function Op over Ω satisfies property (**H**). Consider any $A \subseteq \Omega$, and define C_A^{Op} to be the union of all sets $B \subseteq \Omega$ that enclose A. We call C_A^{Op} the *max enclosure* of A. The following result shows that the max enclosure is monotonic in A, and encloses A (and is thus the unique maximal set enclosing A).

Proposition 7 *Assume that choice function* Op *over* Ω *satisfies property (**H**). Consider any* $A \subseteq \Omega$*. Then* $C_A^{\mathrm{Op}} \rhd_{\mathrm{Op}} A$*, and* $\mathrm{Op}(C_A^{\mathrm{Op}}) \subseteq \mathrm{Op}(A) \subseteq A \subseteq C_A^{\mathrm{Op}}$*. If* $B \subseteq A$ *then* $C_B^{\mathrm{Op}} \subseteq C_A^{\mathrm{Op}}$.

Using Proposition 7, it can be shown that, given Heritage, the Outcast enforcement is equal to the optimal elements of the max enclosure:

Proposition 8 *Given* Op *over* Ω *satisfying property (**H**), define choice function* Op^* *over* Ω *by* $\mathrm{Op}^*(A) = \mathrm{Op}(C_A^{\mathrm{Op}})$ *for all* $A \subseteq \Omega$*. Consider any* $A \subseteq \Omega$.

1. Op^* *satisfies properties (**H**) and (**O**).*
2. $\mathrm{Op}^* \subseteq \mathrm{Op}$*. If* $\mathrm{Op}' \subseteq \mathrm{Op}$ *and* Op' *satisfies property (**O**) then* $\mathrm{Op}' \subseteq \mathrm{Op}^*$*. Thus,* $\mathrm{Op}^* = \mathrm{Op}^{(O)}$.

Theorem 2, which follows immediately from Proposition 8, states that the Outcast-enforcement satisfies Heritage and thus path independence, and corresponds with the optimal elements of the max enclosure operation.

Theorem 2 *Assume that choice function* Op *over* Ω *satisfies property (**H**). Consider any* $A \subseteq \Omega$*. Then* $\mathrm{Op}^{(O)}(A) = \mathrm{Op}(C_A^{\mathrm{Op}})$*. In addition,* $\mathrm{Op}^{(O)}$ *satisfies Property (**H**) and thus is a Plott function.*

Example. Let $\Omega = \{a, b, c, d\}$, and suppose that Op is given by $\mathrm{Op}(B) = \{a, b\}$ if $B = \Omega$, and otherwise, $\mathrm{Op}(B) = B$. It is easily seen that Op satisfies Heritage, but does not satisfy Outcast (because $\mathrm{Op}(\Omega) = \{a, b\} \neq \mathrm{Op}(\{a, b, c\})$). We have $\mathrm{Op}(\Omega) \subseteq \{a, b, c\} \subseteq \Omega$, and so Ω encloses $\{a, b, c\}$, and hence is the max enclosure of $\{a, b, c\}$. Thus, $\mathrm{Op}^{(\mathrm{O})}(\{a, b, c\}) = \{a, b\}$. In fact, $\mathrm{Op}^{(\mathrm{O})}$, which equals the PI-enforcement $\mathrm{Op}^{(\pi)}$ of Op, is given by $\mathrm{Op}^{(\mathrm{O})}(B) = \{a, b\}$ if B equals either Ω, $\{a, b, c\}$ or $\{a, b, d\}$, and otherwise, $\mathrm{Op}^{(\mathrm{O})}(B) = B$. □

Proposition 5(vi) implies that $(\mathrm{Op}^{(\mathrm{H})})^{(\pi)} = \mathrm{Op}^{(\pi)}$, and Theorem 2 implies that if Op satisfies (**H**) then enforcing (**O**) preserves (**H**), and also preserves (**NE**). This leads to the following corollary, which means that to enforce path independence (i.e., (**H**) $\wedge$ (**O**)), we can first enforce (**H**) and then enforce (**O**).

Corollary 1 *For any choice function* Op,

$$(\mathrm{Op}^{(H)})^{(O)} = (\mathrm{Op}^{(H)})^{(\pi)} = \mathrm{Op}^{(\pi)}.$$

If Op *satisfies (**H**) then* $\mathrm{Op}^{(\pi)} = \mathrm{Op}^{(O)}$*, and if* Op *also satisfies (**NE**) then so does* $\mathrm{Op}^{(\pi)}$.

Corollary 1 and Proposition 5(vi) imply that $((\mathrm{Op}^{(\mathrm{O})})^{(\mathrm{H})})^{(\mathrm{O})} = (\mathrm{Op}^{(\mathrm{O})})^{(\pi)} = \mathrm{Op}^{(\pi)}$. One might wonder if $(\mathrm{Op}^{(\mathrm{O})})^{(\mathrm{H})} = \mathrm{Op}^{(\pi)}$ also necessarily holds. This is, however, not the case, because applying (**H**)-enforcement can lose property (**O**). For example, suppose that $\Omega = \{a, b, c\}$, and that $\mathrm{Op}(\{a, b\}) = \{a\}$, $\mathrm{Op}(\{b, c\}) = \{b\}$, $\mathrm{Op}(\{a, c\}) = \{c\}$, and $\mathrm{Op}(A) = A$ for all other sets $A \subseteq \Omega$. Op satisfies (**O**), and $\mathrm{Op}^{(\mathrm{H})}$ is the same as Op except that $\mathrm{Op}^{(\mathrm{H})}(\{a, b, c\}) = \emptyset$, which implies that $\mathrm{Op}^{(\mathrm{H})}$ does not satisfy (**O**), and that $\mathrm{Op}^{(\pi)} = (\mathrm{Op}^{(\mathrm{H})})^{(\mathrm{O})} = \mathrm{Emp}$. Thus, $(\mathrm{Op}^{(\mathrm{O})})^{(\mathrm{H})} = \mathrm{Op}^{(\mathrm{H})} \neq \mathrm{Op}^{(\pi)}$ for this particular choice function Op.

If Op does not satisfy (**H**) then applying (**O**)-enforcement can lose consistency (**NE**). In fact, we can even have $\mathrm{Op}^{(\mathrm{O})} = \mathrm{Emp}$ for consistent Op. For example, suppose that $\Omega = \{a, b, c\}$, and $\mathrm{Op}(\{a, b, c\}) = \{a\}$, $\mathrm{Op}(\{a, b\}) = \{b\}$, and $\mathrm{Op}(A) = A$ for other sets A. Then, using the facts that $\mathrm{Op}^{(\mathrm{O})} \subseteq \mathrm{Op}$ and $\mathrm{Op}^{(\mathrm{O})}$ satisfies (**O**), we have $\mathrm{Op}^{(\mathrm{O})}(\{a, b, c\}) \subseteq \{a, b\} \subseteq \{a, b, c\}$, and hence, $\{a\} \supseteq \mathrm{Op}^{(\mathrm{O})}(\{a, b, c\}) = \mathrm{Op}^{(\mathrm{O})}(\{a, b\}) \subseteq \{b\}$, so $\mathrm{Op}^{(\mathrm{O})}(\{a, b, c\}) = \emptyset$, and thus, $\mathrm{Op}^{(\mathrm{O})} = \mathrm{Emp}$, by (**O**).

To summarise, we can enforce path independence by enforcing Heritage and then enforcing Outcast. So, if a choice function Op satisfies Heritage then enforcing path independence is the same as enforcing Outcast. If, in addition, Op is consistent then enforcing Outcast maintains consistency. In contrast, if we take a choice function Op satisfying Outcast, then enforcing Heritage does not necessarily enforce path consistency, since the Outcast property may be lost. In addition, enforcing Outcast can lose consistency, if the choice function does not satisfy Heritage.

4.4 Computation of Enforcements

With an explicit representation of a choice function Op, Proposition 6 and Theorem 2 enable efficient computation of $\mathrm{Op}^{(\mathrm{H})}$ and $\mathrm{Op}^{(\pi)}$, in linear time in the size of the representation. Regarding enforcing (**H**), we can update Op by starting with sets of cardinality 1, and then sets of cardinality 2 etc, using the update rule $\overline{\mathrm{Op}}(A) := \overline{\mathrm{Op}}(A) \cup \bigcup \{\overline{\mathrm{Op}}(B) : B \subseteq A,\ |B| = |A| - 1\}$.

If Ω is not very small then the explicit representation of a choice function will be infeasible, and we might, as discussed in Section 2.3, represent a function Op on a set $\mathcal{L}$ of subsets of Ω, arising from elicited preferences. Since $\overline{\mathrm{Op}}(B)$ is empty for B not in $\mathcal{L}$, we have, by Proposition 6, for $A \in \mathcal{L}$, $\overline{\mathrm{Op}}^{(\mathrm{H})}(A) = \bigcup_{B\in\mathcal{L}:B\subseteq A} \overline{\mathrm{Op}}(B)$. The computation of $\overline{\mathrm{Op}}^{(\mathrm{H})}$ (and thus $\mathrm{Op}^{(\mathrm{H})}$) over $\mathcal{L}$ can be done incrementally, starting from the minimal subsets in $\mathcal{L}$ and working upwards. After computing $\mathrm{Op}^{(\mathrm{H})}$ on $\mathcal{L}$, one can compute $\mathrm{Op}^{(\mathrm{H})}(A)$ for any particular set $A \in 2^\Omega$ that is not in $\mathcal{L}$, using $\overline{\mathrm{Op}}^{(\mathrm{H})}(A) = \bigcup_B \overline{\mathrm{Op}}(B)$ where the union is taken over all maximal subsets B of A within $\mathcal{L}$.

For the (**O**)-enforcement of Op that satisfies (**H**), one can set $\mathrm{Op}^{(\mathrm{O})}(A) = \mathrm{Op}(\Omega)$ for all A such that $\mathrm{Op}(\Omega) \subseteq A \subseteq \Omega$. Then, we iterate the process, by choosing any maximal set C whose value $\mathrm{Op}^{(\mathrm{O})}(C)$ has not yet been defined, and set $\mathrm{Op}^{(\mathrm{O})}(A) = \mathrm{Op}(C)$ for all A such that $\mathrm{Op}(C) \subseteq A \subseteq C$.

Of course, an explicit representation of Op is exponential in $|\Omega|$; but, in many situations we will have an implicit representation, and we will be interested in computing the path independent enforcement $\mathrm{Op}^{(\pi)}(A)$ for an individual set A corresponding to the set of currently available alternatives. Theorem 3 below shows that this can be done very efficiently for Op satisfying (**H**), requiring less than $|\Omega|$ calls of $\mathrm{Op}(\cdot)$ to compute $\mathrm{Op}^{(\pi)}(A)$. We generate, in a particular way, a nested sequence of sets $\Omega = B_0 \supsetneq B_1 \supsetneq \cdots \supsetneq B_k$, where B_k is the first element in the sequence such that $\mathrm{Op}(B_k) \subseteq A$; we obtain $\mathrm{Op}^{(\pi)}(A) = \mathrm{Op}(B_k)$.

The operation that generates B_i from B_{i-1} can be described as follows. Given a superset B of A, we delete all the optimal elements that are not in A; this generates a new set which is a subset of B but still contains A. We iterate the operation until we reach a fixed point C, i.e., when all optimal elements are in A, and so C encloses A.

If we start with any superset of the max enclosure C_A of A then the fixed point is the max enclosure, since it encloses A and contains C_A by Proposition 9(2) below. In particular, starting with Ω, the set of all alternatives, the fixed point will be the max enclosure of A, and thus equal to the PI-enforcement of A, by Theorem 2 and Corollary 1.

Theorem 3 *Let* Op *be a choice function satisfying (**H**), and let A be a subset of Ω. Define the sequence of sets $B_0, B_1, \ldots$ by $B_0 = \Omega$ and for $i \geq 1$, $B_i = A \cup (B_{i-1} \setminus \mathrm{Op}(B_{i-1}))$. There exists $k \leq |\Omega| - |A|$ such that $B_{k+1} = B_k$, and $\mathrm{Op}(B_k) \subseteq A$, and for all i with $1 \leq i < k$, $B_i \subsetneq B_{i-1}$; and $B_{k+1} = B_k$. We have $\mathrm{Op}^{(\pi)}(A) = \mathrm{Op}(B_k)$.*

Theorem 3 follows easily using Proposition 9 below.

Proposition 9 *Let* Op *be a choice function satisfying (**H**), and let $A \subseteq \Omega$, and define, for $A \subseteq \Omega$, the function f^A_{Op} by, for $E \subseteq \Omega$, $f^A_{\mathrm{Op}}(E) = A \cup (E \setminus \mathrm{Op}(E))$.*

1. *If $A \subseteq E$ then $A \subseteq f^A_{\mathrm{Op}}(E) \subseteq E$, and $f^A_{\mathrm{Op}}(E) = E \iff \mathrm{Op}(E) \subseteq A$.*
2. *Suppose that $C \subseteq \Omega$ is such that $\mathrm{Op}(C) \subseteq A$. If $E \supseteq C$ then $f^A_{\mathrm{Op}}(E) \supseteq C$.*
3. *E is a fixed point of f^A_{Op} if and only if $\mathrm{Op}(E) \subseteq A \subseteq E$, i.e., $E \rhd_{\mathrm{Op}} A$ (E encloses A).*

As discussed in Section 2.3, a Plott function arises naturally as a Possibly Optimal function PO in preference elicitation; with some common preference models, given an explicit set A of alternatives, $\mathrm{PO}(A)$ is computable in polynomial time using linear programming (see Section 2.3). When combining (using intersection) two such functions PO_1 and PO_2 (one for each agent), their intersection $\mathrm{Op} = \mathrm{PO}_1 \cap \mathrm{PO}_2$ satisfies (**H**), so Theorem 3 allows a value of the PI-closure $\mathrm{Op}^{(\pi)}(A)$ to be computed in polynomial time.

We illustrate the algorithm for computing $\mathrm{Op}^{(\pi)}(A)$, for the case in which Op is the intersection of two consistent Plott functions. Let $\Omega = \{a, b, c, d, e\}$, let $\mathrm{Op} = \mathrm{Op}_1 \cap \mathrm{Op}_2$, where Op_1 equals $\mathrm{PO}_{\{abcde, adecb, bcaed\}}$, i.e., the union $O_{abcde} \cup O_{adecb} \cup O_{bcaed}$, where O_{abcde}, for instance, is the choice function generated by the total order $a > b > c > d > e$, so e.g., $O_{abcde}(\{c, d, e\}) = \{c\}$. Let $\mathrm{Op}_2 = \mathrm{PO}_{\{baecd, cbaed\}}$.

We have $\mathrm{Op}_1(\{a, b, c, d, e\}) = \{a, b\}$ because a is the best element for total orders O_{abcde} and O_{adecb}, and b is the best element for total order O_{bcaed}. Similarly, $\mathrm{Op}_2(\{a, b, c, d, e\}) = \{b, c\}$.

Let $A = \{c, e\}$. Then, $B_0 = \Omega = \{a, b, c, d, e\}$; so, $\mathrm{Op}(B_0) = \mathrm{Op}_1(\{a, b, c, d, e\}) \cap \mathrm{Op}_2(\{a, b, c, d, e\}) = \{a, b\} \cap \{b, c\} = \{b\}$. Then, $B_0 \setminus \mathrm{Op}(B_0) = \{a, c, d, e\}$ and so $B_1 = \{c, e\} \cup \{a, c, d, e\} = \{a, c, d, e\}$. Similarly, $\mathrm{Op}(B_1) = \{a, c\} \cap \{a, c\} = \{a, c\}$. Then, $B_1 \setminus \mathrm{Op}(B_1) = \{d, e\}$ and so $B_2 = \{c, e\} \cup \{d, e\} = \{c, d, e\}$. $\mathrm{Op}(B_2) = \{c, d\} \cap \{c, e\} = \{c\}$. Since $\mathrm{Op}(B_2) \subseteq A$ we have, by Theorem 3, $\mathrm{Op}^{(\pi)}(A) = \mathrm{Op}(B_2) = \{c\}$. Note that $\mathrm{Op}(A) = \{c, e\} \neq \mathrm{Op}^{(\pi)}(A)$, which illustrates the fact that Op is not a Plott function.

5 Restoring consistency (NE)

Suppose that we have a choice function Op that does not satisfy property (**NE**), so that $\mathrm{Op}(A) = \emptyset$ for some non-empty $A \subseteq \Omega$; (in particular, Op may arise as the intersection of two Plott functions). We cannot restore (**NE**) in the same way that we enforced the properties in Section 4.1, since if Op fails to satisfy (**NE**) and $\mathrm{Op}' \subseteq \mathrm{Op}$ then Op' fails to satisfy (**NE**) (since $\mathrm{Op}(A) = \emptyset$ implies $\mathrm{Op}'(A) = \emptyset$). Instead we need to weaken Op rather than strengthen it.

The basic idea of our approach here is to replace an empty value by a backup value; that is, if $\mathrm{Op}(A) = \emptyset$, we reset $\mathrm{Op}(A)$ to be some non-empty set $\sigma(A)$ leading to a new version Op^σ of Op. For example, we could choose $\sigma(A)$ to be some default value; in particular, we might set $\sigma(A) = A$ for all subsets A, so that σ is the vacuous (identity) choice function Id.

Definition 2 Op^σ*: Let* Op *and σ be choice functions over Ω. Define Op^σ by $\mathrm{Op}^\sigma(A) = \mathrm{Op}(A)$ if $\mathrm{Op}(A)$ is non-empty, and otherwise, define $\mathrm{Op}^\sigma(A) = \sigma(A)$.*

Clearly, $\mathrm{Op} \subseteq \mathrm{Op}^\sigma$. If Op is consistent (i.e., satisfies (**NE**)) then $\mathrm{Op}^\sigma = \mathrm{Op}$. If σ is consistent then so is Op^σ. Also, if $\mathrm{Op} \subseteq \sigma$ then $\mathrm{Op}^\sigma \subseteq \sigma$.

The following result shows that we can restore consistency to a Plott function by just replacing the empty values by the values of a consistent Plott function, and still maintain path independence.

Proposition 10 *Let* Op *and σ be Plott functions over Ω. Then Op^σ is a Plott function, and if σ satisfies (**NE**) then so does Op^σ.*

If we are intersecting choice functions Op_1 and Op_2, and setting $\mathrm{Op} = (\mathrm{Op}_1 \cap \mathrm{Op}_2)^{(\pi)}$, another simple option is to use $\sigma = \mathrm{Op}_1 \cup \mathrm{Op}_2$, which ensures that the final combined choice function is always weaker than (or equal to) the union.

The result below shows that, in Plott functions, the sets that are mapped to the empty set are all the subsets of a particular set Z_{Op}. This means that Op^σ equals σ on subsets of Z_{Op}, and equals Op otherwise.

Proposition 11 *If* Op *is a Plott function over Ω then there exists some subset $Z_{\mathrm{Op}} \subseteq \Omega$ such that $\mathrm{Op}(A) = \emptyset \iff A \subseteq Z_{\mathrm{Op}}$. For all $A \subseteq \Omega$, $\mathrm{Op}(A) = \mathrm{Op}(A \setminus Z_{\mathrm{Op}}) \subseteq A \setminus Z_{\mathrm{Op}}$. If $A \subseteq Z_{\mathrm{Op}}$ then $\mathrm{Op}^\sigma(A) = \sigma(A)$; otherwise, $\mathrm{Op}^\sigma(A) = \mathrm{Op}(A)$.*

Iterative consistency restoration

Suppose we have a nested sequence of Plott functions $\mathrm{Op} \subseteq \sigma_1 \subseteq \cdots \subseteq \sigma_k = \mathrm{Id}$, representing a progressive weakening of Op. Then we can iteratively apply $(\cdot)^{\sigma_j}$, giving $\mathrm{Op}' = \mathrm{Op}^{\sigma_1 \sigma_2 \cdots \sigma_k}$ (i.e., $(\cdots(\mathrm{Op}^{\sigma_1})^{\sigma_2}\cdots)^{\sigma_k}$) which is a consistent Plott function by Proposition 10. If $\mathrm{Op}(A) \neq \emptyset$ then $\mathrm{Op}'(A) = \mathrm{Op}(A)$, and otherwise, $\mathrm{Op}'(A) = \sigma_j(A)$, where j is minimal such that $\sigma_j(A)$ is non-empty.

In particular, suppose we want to combine two agents' Plott functions Op_1 and Op_2, based on intersection, and achieve a consistent Plott function; we assume we have, for some $k \geq 1$, progressive weakenings of the two Plott functions: $\mathrm{Op}_z \subseteq \mathrm{Op}^1_z \subseteq \cdots \subseteq \mathrm{Op}^k_z = \mathrm{Id}$, for $z = 1, 2$. Let $\mathrm{Op} = (\mathrm{Op}_1 \cap \mathrm{Op}_2)^{(\pi)}$ be the PI-enforcement of the intersection, and let $\sigma_j = (\mathrm{Op}^j_1 \cap \mathrm{Op}^j_2)^{(\pi)}$, for $j = 1, \ldots, k$, so $\mathrm{Op} \subseteq \sigma_1 \subseteq \cdots \subseteq \sigma_k = \mathrm{Id}$, since PI-enforcement is monotone (see Proposition 5). Then we can restore consistency progressively with $\sigma_1, \sigma_2, \ldots, \sigma_k$, in order to try to keep as much information as possible from Op, giving the consistent Plott function $\mathrm{Op}^{\sigma_1\sigma_2\cdots\sigma_k}$.

Pre-processing with another consistency restoration

Proposition 10 implies that for any choice function Op, and for a consistent Plott function σ, $\mathrm{Op}^{(\pi)\sigma}$ (i.e., $(\mathrm{Op}^{(\pi)})^{\sigma}$) is a consistent Plott function and that $\mathrm{Op}^{(\pi)\sigma}(A)$ equals the PI-enforcement $\mathrm{Op}^{(\pi)}(A)$ when the latter is non-empty. However, there are some situations when the consistency restoration can seem too drastic, in particular if $\mathrm{Op}(\Omega) = \emptyset$; in that case, $\mathrm{Op}^{(\pi)}(A) = \emptyset$ for all $A \subseteq \Omega$ (i.e., $\mathrm{Op}^{(\pi)} = \mathrm{Emp}$) and so $\mathrm{Op}^{(\pi)\sigma}$ just equals σ, and no further (consistent) information from Op is used. (This is a problem especially if we don't have gradual progressive weakenings of Op, discussed in the last paragraph.)

To help avoid this issue, we can apply an additional pre-processing step: for Op satisfying (**H**), we first restore consistency, thus, considering the operator $(\cdot)^{\sigma(\pi)\sigma}$.

It seems desirable to only make changes for sets for which the PI-enforcement is empty. This is assured for operator $(\cdot)^{\sigma(\pi)\sigma}$ by the following result.

Proposition 12 *Assume that* Op *is a choice function that satisfies (**H**), and that σ is an arbitrary choice function. If* $\mathrm{Op}^{(\pi)}(A)$ *is non-empty then* $\mathrm{Op}^{\sigma(\pi)\sigma}(A) = \mathrm{Op}^{(\pi)\sigma}(A) = \mathrm{Op}^{\sigma(\pi)}(A) = \mathrm{Op}^{(\pi)}(A)$.

Proposition 13 below implies that $\mathrm{Op}^{\sigma(\pi)\sigma}$ is a consistent Plott function that is closer to the PI-enforcement than $\mathrm{Op}^{(\pi)\sigma}$.

Proposition 13 *Suppose that* Op *is choice function over Ω that satisfies (**H**), and that σ is a consistent Plott function over Ω such that* $\mathrm{Op} \subseteq \sigma$. *Then* $\mathrm{Op}^{\sigma(\pi)\sigma}$ *and* $\mathrm{Op}^{(\pi)\sigma}$ *are consistent Plott functions and* $\mathrm{Op}^{(\pi)} \subseteq \mathrm{Op}^{\sigma(\pi)\sigma} \subseteq \mathrm{Op}^{(\pi)\sigma}$. *If* Op *is a Plott function then* $\mathrm{Op}^{\sigma(\pi)\sigma} = \mathrm{Op}^{(\pi)\sigma} = \mathrm{Op}^{\sigma}$.

Example. Let $\Omega = \{a, b, c, d\}$, let $\mathrm{Op} = \mathrm{Op}_1 \cap \mathrm{Op}_2$, where $\mathrm{Op}_1 = \mathrm{PO}_{\{abcd, acbd, adbc\}}$, and $\mathrm{Op}_2 = \mathrm{PO}_{\{bacd, cabd\}}$. Then $\mathrm{Op}(\Omega) = \{a\} \cap \{b, c\} = \emptyset$, so (**NE**) fails for the intersection Op. Let $\sigma = \mathrm{Op}_1 \cup \mathrm{Op}_2 = \mathrm{PO}_{\{abcd, acbd, adbc, bacd, cabd\}}$. Since $\mathrm{Op}(\Omega) = \emptyset$, we have that Ω is the max enclosure of every set and so the PI-enforcement $\mathrm{Op}^{(\pi)}$ equals Emp (this also follows directly from the Outcast property, putting $A = \Omega$), and thus, $\mathrm{Op}^{(\pi)\sigma} = \sigma$, and e.g., $\mathrm{Op}^{(\pi)\sigma}(\Omega) = \{a, b, c\}$. In contrast, $\mathrm{Op}^{\sigma(\pi)\sigma}$ is stronger, equalling $\mathrm{Op}^{\sigma(\pi)}$, and with $\mathrm{Op}^{\sigma(\pi)\sigma}(\Omega) = \{a\}$. □

Restoring consistency and enforcing PI for the intersection of two total orders

Although our focus and motivation is based more on imprecise than precise choice functions, it is revealing to consider what happens in a precise case, in which we wish to combine two choice functions corresponding with total orders. Interestingly, it turns out that restoring consistency first with the vacuous choice function leads to a natural result, with the enforcement of path independence transferring useful information from non-conflicting sets to the conflicting ones.

Let $\geq_1$ and $\geq_2$ be two total orders on Ω, and let $\geq_\cap$ be their intersection, so that $\alpha \geq_\cap \beta$ if and only if $\alpha \geq_1 \beta$ and $\alpha \geq_2 \beta$. For $i = 1, 2$, let Op_i be the choice function associated with $\geq_i$, so that $\mathrm{Op}_i(A)$ is equal to the singleton set $\{max_{\geq_i}(A)\}$ consisting of the best element of A with respect to $\geq_i$. Let $\mathrm{Op} = \mathrm{Op}_1 \cap \mathrm{Op}_2$. Now, $\mathrm{Op}(A)$ is equal to $\{\alpha\}$ for a case in which α is the best element of A with respect to both orderings; otherwise, $\mathrm{Op}(A)$ is empty. In particular, if α and β are different alternatives then $\mathrm{Op}(\{\alpha, \beta\})$ is empty unless $\geq_1$ and $\geq_2$ agree on which of α and β is better. This implies that the choice function Op is not consistent unless the two total orders $\geq_1$ and $\geq_2$ are identical.

We consider restoring consistency and PI-enforcement, where we restore with respect to either $\sigma_1 = \mathrm{Id}$ (the vacuous choice function) or $\sigma_2 = \mathrm{Op}_1 \cup \mathrm{Op}_2$.

Restoring consistency with union then PI-enforcement: we have that $\mathrm{Op}^{\sigma_2(\pi)}(A) = \{max_{\geq_1}(A), max_{\geq_2}(A)\}$, i.e., we return the best element with respect to each total order. This is because if $\mathrm{Op}(A) \neq \emptyset$ then $\mathrm{Op}(A) = \mathrm{Op}_1(A) = \mathrm{Op}_2(A) = \sigma_2(A)$, and thus, $\mathrm{Op}^{\sigma_2} = \sigma_2$, which implies that $\mathrm{Op}^{\sigma_2(\pi)} = \sigma_2$ as the latter is a Plott function.

Restoring consistency with identity and then PI-enforcement: $\mathrm{Op}^{\sigma_1(\pi)}(A)$ is equal to the set of Pareto-undominated elements of A (call this $\mathrm{Op}'(A)$) i.e., all elements of A except those β with $\alpha >_\cap \beta$ for some $\alpha \in A$.

The set $\mathrm{Op}^{\sigma_1(\pi)}(A)$ contains both optimal elements $max_{\geq_1}(A)$ and $max_{\geq_2}(A)$ but may well contain many others as well; it is a little like a qualitative convex closure of Op_1 and Op_2. In a fully qualitative setting, it might be argued that this is the most natural result; in particular, it corresponds to the set of all alternatives α that could have maximum sum of utility $U_1(\alpha) + U_2(\alpha)$, where for $i = 1, 2$, U_i is some utility function on Ω that is compatible with $\geq_i$.

We can also consider PI-enforcement followed by restoring consistency. If the best element of Ω with respect to $\geq_1$ is not the best element of Ω with respect to $\geq_2$ then $\mathrm{Op}(\Omega) = \emptyset$, which implies, from the Outcast property, that $\mathrm{Op}^{(\pi)} = \mathrm{Emp}$, the trivial null choice function. Then $\mathrm{Op}^{(\pi)\sigma}$ is just equal to σ. This is thus another kind of example illustrating that restoring consistency first can lead to a stronger choice function.

6 Discussion

We have considered methods for the combination of a pair of choice functions, and analysed their properties. Natural combination methods can lose key properties, and we consider how path independence can be enforced, and consistency restored, which leads to combinations operations which maintain these properties.

Path independence can be enforced for a choice function by enforcing first the Heritage property and then the Outcast property. If the choice function already satisfies Heritage, then the preferred set among a given set of alternatives can be efficiently computed for the new path independent choice function. We have also shown how the non-emptiness property can be restored, to generate a consistent Plott function, with two different approaches. These methods can be used to generate a consistent Plott function when combining the preferences of more than one agent, in particular, by applying the methods to the intersection of their choice functions.

In future work, it would be valuable to consider more complex methods for combining multiple agents' choice functions, and consider what further properties of the combination one would like. For example, it could be possible to apply belief merging approaches e.g., [14, 13], for each given subset of alternatives A, and then enforce/restore desirable properties to the resulting choice function. It would be interesting to also explore the application of our enforcing/restoring methods to social choice functions generated by various forms of voting rule under uncertainty.

Acknowledgements

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 12/RC/2289-P2 at Insight the SFI Research Centre for Data Analytics at UCC, which is co-funded under the European Regional Development Fund; it has also been supported by the EU H2020 ICT48 project "TAILOR", under contract #952215. I also thank the reviewers for their constructive comments.

References

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

[2] H. Andréka, M. Ryan, and P.-Y. Schobbens, 'Operators and laws for combining preference relations', *J. Log. Comput.*, **12**(1), 13–53, (2002).

[3] N. Benabbou and P. Perny, 'Incremental weight elicitation for multiobjective state space search', in *Proc. AAAI 2015*, pp. 1093–1099, (2015).

[4] N. Benabbou and P. Perny, 'Adaptive elicitation of preferences under uncertainty in sequential decision making problems', in *Proc. IJCAI 2017*, pp. 4566–4572, (2017).

[5] D.H. Blair, 'Path independent social choice functions: a further result', *Econometrica*, **43**, 173–174, (1975).

[6] S. Bouveret, Y. Chevaleyre, F. Durand, and J. Lang, 'Voting by sequential elimination with few voters', in *Proc. IJCAI 2017*, pp. 128–134, (2017).

[7] F. Brandt and P. Harrenstein, 'Set-rationalizable choice and self-stability', *J. Economic Theory*, **146**(4), 1721–1731, (2011).

[8] D. Braziunas and C. Boutilier, 'Minimax regret based elicitation of generalized additive utilities.', in *UAI*, pp. 25–32, (2007).

[9] Susumu Cato, 'Choice functions and weak Nash axioms', *Review of Economic Design*, **22**, 159–176, (2018).

[10] H. Chernoff, 'Rational selection of decision functions', *Econometrica*, **22**, 422–443, (1954).

[11] V. Danilov and G. Koshevoy, 'Mathematics of Plott choice functions', *Mathematical Social Sciences*, **49**, 245—-272, (2005).

[12] M. Gelain, M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh, 'Elicitation strategies for soft constraint problems with missing preferences: Properties, algorithms and experimental studies', *Artif. Intell.*, **174**(3-4), 270–294, (2010).

[13] S. Konieczny, J. Lang, and P. Marquis, 'DA2 merging operators', *Artificial Intelligence*, **157**(1-2), 49–79, (2004).

[14] S. Konieczny and R. Pino Pérez, 'Merging information under constraints: a logical framework', *Journal of Logic and computation*, **12**(5), 773–808, (2002).

[15] R. D. Luce and H. Raiffa, *Games and decisions: introduction and critical survey*, Wiley, New York, 1957.

[16] R. Marinescu, A. Razak, and N. Wilson, 'Multi-objective constraint optimization with tradeoffs', in *Proc. CP-2013*, pp. 497–512, (2013).

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

[18] J.F. Nash, 'The bargaining problem', *Econometrica*, **18**(2), 155–162, (1950).

[19] C. R. Plott, 'Path independence, rationality, and social choice', *Econometrica*, **41**, 1075–1091, (1973).

[20] A. K. Sen, 'Choice functions and revealed preference', *Rev. Econ. Stud.*, **38**(3), 307–317, (1971).

[21] F. Toffano and N. Wilson, 'Minimality and comparison of sets of multi-attribute vectors', in *Proc. ECAI-2020*, (2020).

[22] P. Viappiani and C. Boutilier, 'Regret-based optimal recommendation sets in conversational recommender systems', in *Proc. RecSys-2009*, pp. 101–108. ACM, (2009).

[23] P. Viappiani and C. Boutilier, 'On the equivalence of optimal recommendation sets and myopically optimal query sets', *Artif. Intell.*, **286**, 103328, (2020).

[24] N. Wilson, *Enforcing Natural Properties of Choice Functions, with Application for Combination (Longer Version)*, http://ucc.insight-centre.org/nwilson/EnforcingPropertiesCFsLonger.pdf, 2023.

[25] N. Wilson and A.-M. George, 'Efficient inference and computation of optimal alternatives for preference languages based on lexicographic models', in *Proc. IJCAI 2017*, pp. 1311–1317, (2017).

[26] N. Wilson and C. O'Mahony, 'The relationships between qualitative notions of optimality for decision making under logical uncertainty', in *Proc. AICS-2011*, (2011).

[27] N. Wilson, A. Razak, and R. Marinescu, 'Computing possibly optimal solutions for multi-objective constraint optimisation with tradeoffs', in *Proc. IJCAI-2015*, (2015).