Strong Equivalence of Qualitative Optimization Problems

Wolfgang Faber
University of Calabria
faber@mat.unical.it

Mirosław Truszczyński
University of Kentucky
mirek@cs.uky.edu

Stefan Woltran
Vienna University of Technology
woltran@dbai.tuwien.ac.at

Abstract

We introduce the framework of *qualitative optimization problems* (or, simply, optimization problems) to represent preference theories. The formalism uses separate modules to describe the space of outcomes to be compared (the *generator*) and the preferences on outcomes (the *selector*). We consider two types of optimization problems. They differ in the way the generator, which we model by a propositional theory, is interpreted: by the standard propositional logic semantics, and by the equilibrium-model (answer-set) semantics. Under the latter interpretation of generators, optimization problems directly generalize answer-set optimization programs proposed previously. We study *strong equivalence* of optimization problems, which guarantees their interchangeability within any larger context. We characterize several versions of strong equivalence obtained by restricting the class of optimization problems that can be used as extensions and establish the complexity of associated reasoning tasks. Understanding strong equivalence is essential for modular representation of optimization problems and rewriting techniques to simplify them without changing their inherent properties.

Introduction

We introduce the framework of *qualitative optimization problems* in which, following the design of answer-set optimization (ASO) programs (Brewka, Niemelä, and Truszczyński 2003), we use separate modules to describe the space of outcomes to be compared (the *generator*) and the preferences on the outcomes (the *selector*). In all optimization problems we consider, the selector module follows the syntax and the semantics of preference modules in ASO programs, and the generator is given by a propositional theory. If this propositional theory is interpreted according to the standard propositional logic semantics, that is, outcomes to be compared are classical models of the generator, we speak about *classical optimization problems* (CO problems, for short). If the generator theory is interpreted by the semantics of *equilibrium* models (Pearce 1997), also known as *answer sets* (Ferraris 2005), that is, it is the answer sets of the generator that are being compared, we speak about *answer-set optimization problems* (ASO problems, for short).

Representing and reasoning about preferences in qualitative settings is an important research area for knowledge representation and qualitative decision theory. The main objectives are to design expressive yet intuitive languages to model preferences, and to develop automated methods to reason about formal representations of preferences in these languages. The literature on the subject of preferences is vast. We refer the reader to the special issue of Artificial Intelligence Magazine (Goldsmith and Junker 2008) for a collection of overview articles and references.

Understanding when optimization problems are equivalent, in particular, when one can be interchanged with another within any larger context, is fundamental to any preference formalism. Speaking informally, optimization problems P and Q are *interchangeable* or *strongly equivalent* when for every optimization problem R (*context*), $P \cup R$ and $Q \cup R$ define the same optimal models. Understanding when one optimization problem is equivalent to another in this sense is essential for preference analysis, modular preference representation, and rewriting techniques to simplify optimization problems into forms more amenable to processing, without changing any of their inherent properties. Let us consider a multi-agent setting, in which agents combine their preferences on some set of alternatives with the goal of identifying optimal ones. Can one agent in the ensemble be replaced with another so that the set of optimal alternatives is unaffected not only now, but also under any extension of the ensemble in the future? Strong equivalence of agents' optimization problems is precisely what is needed to guarantee this full interchangeability property!

The notion of strong equivalence is of general interest, by no means restricted to preference formalisms. In some cases, most notably for classical logic, it coincides with *equivalence*, the property of having the same models. However, if the semantics is not monotone, that is, extending the theory may introduce new models, not only eliminate some, strong equivalence becomes a strictly stronger concept, and the one to adopt if theories being analyzed are to be placed within a larger context. The nonmonotonicity of the semantics is the salient feature of nonmonotonic logics (Marek and Truszczyński 1993) and strong equivalence of theories in nonmonotonic logics, especially logic programming with the answer-set semantics (Gelfond and Lifschitz 1991), was extensively studied in that setting (Lifschitz, Pearce, and Valverde 2001; Turner 2003; Eiter, Fink, and Woltran 2007). Preference formalisms also often behave nonmonotonically as adding a new preference may cause a non-optimal outcome (model) to become an optimal one. Thus, in preference formalisms, equivalence and strong equivalence are typically different notions. Accordingly, strong equivalence was studied for logic programs with rule preferences (Faber and Konczak 2006), programs with ordered disjunction (Faber, Tompits, and Woltran 2008) and programs with weak constraints (Eiter et al. 2007).

We extend the study of strong equivalence to the formalism of *qualitative optimization problems*. The formalism is motivated by the design of answer-set optimization (ASO) programs of Brewka, Niemelä, and Truszczyński (2003). It borrows two key features from ASO programs that make it an attractive alternative to the preference modeling approaches based on logic programming that we mentioned above. First, following ASO programs, optimization problems provide a clear separation of hard constraints, which specify the space of feasible outcomes, and preferences (soft constraints) that impose a preference ordering on feasible outcomes. Second, optimization problems adopt the syntax and the semantics of preference rules of ASO programs that correspond closely to linguistic patterns of simple conditional preferences used by humans.

The separation of preference modules from hard constraints facilitates eliciting and representing preferences. It is also important for characterizing strong equivalence. When a clear separation is not present, like in logic programs with ordered disjunctions, strong equivalence characterizations are cumbersome as they have to account for complex and mostly implicit interactions between hard constraints and preferences. For optimization problems, which impose the separation, we have "one-dimensional" forms of strong equivalence, in which only hard constraints or only preferences are added. These "one-dimensional" concepts are easier to study yet provide enough information to construct characterizations for the general case.

Main Contributions. Our main contribution can be summarized as follows.

- We propose a general framework of qualitative optimization problems, extending in several ways the formalism of ASO programs. We focus on two important instantiations of the framework, the classes of classical optimization (CO) problems and answer-set optimization (ASO) problems. The latter one directly generalizes ASO programs.
- We characterize the concept of strong equivalence of optimization problems relative to changing selector modules. The characterization is independent of the semantics of generators and so, applies both to CO and ASP problems. We also characterize strong equivalence relative to changing generators (with preferences fixed). In this case, not surprisingly, the characterization depends on the semantics of generators. However, we show that the dependence is quite uniform, and involves a characterization of strong equivalence of generators relative to their underlying semantics, when they are considered on their own as propositional theories. Finally, we combine the characterizations of the "one-dimensional" concepts of strong equivalence into a characterization of the general "combined" notion.
- We develop our results for the case when preferences are ranked. In practice, preferences are commonly ranked due to the hierarchical structure of preference providers. The general case we study allows for additions of preferences of ranks from a specified interval $[i, j]$. This covers the case when only some segment in the hierarchy of preference providers is allowed to add preferences (top decision makers, middle management, low-level designers), as well as the case when there is no distinction between the importance of preferences (the non-ranked case).
- We establish the complexity of deciding whether two optimization problems are strongly equivalent relative to changing selectors, generators, or both.

Due to the space restriction, we present only proof sketches and some of the simpler and not overly technical proofs here. All proofs can be found in the full version of the paper at http://arxiv.org/abs/1112.0791.

Optimization Problems

A *qualitative optimization problem* (an *optimization problem*, from now on) is an ordered pair $P = (T, S)$, where T is called the *generator* and S the *selector*. The role of the generator is to specify the family of *outcomes* to be compared. The role of the selector S is to define a relation $\geq$ on the set of outcomes and, consequently, define the notion of an optimal outcome. The relation $\geq$ induces relations $>$ and $\approx$: we define $I > J$ if $I \geq J$ and $J \not\geq I$, and $I \approx J$ if $I \geq J$ and $J \geq I$. For an optimization problem P, we write P^g and P^s to refer to its generator and selector, respectively.

Generators. As generators we use propositional theories in the language determined by a fixed countable universe of propositional variables (or alphabet) $\mathcal{U}$ that form atomic propositions, a boolean constant $\bot$, and boolean connectives $\wedge$, $\vee$ and $\rightarrow$, and where we define the constant $\top$, and the connectives $\neg$ and $\leftrightarrow$ in the usual way as $\top := \neg\bot$, $\neg\phi := \phi \rightarrow \bot$, and $\phi \leftrightarrow \psi := (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$, respectively.[1] Models of the generator, as defined by the semantics used, represent *outcomes* of the corresponding optimization problem. We consider two quite different semantics for generators: the classical propositional logic semantics and the semantics of *equilibrium* models (Pearce 1997). Thus, outcomes are either models or equilibrium models, depending on the semantics chosen. The first semantics is of interest due to the fundamental role and widespread use of classical propositional logic, in particular, as a means to describe constraints. Equilibrium models generalize answer sets of logic programs to the case of arbitrary propositional theories (Pearce 1997; Ferraris 2005) and are often referred to as answer sets. The semantics of equilibrium models is important due to the demonstrated effectiveness of logic programming

[1] While the choice of primitive connectives is not common for the language of classical propositional logic, it is standard for the of logic here-and-there which underlies the answer-set semantics.

with the semantics of answer sets for knowledge representation applications. We use the terms equilibrium models and answer sets interchangeably.

Throughout the paper, we represent interpretations as subsets of $\mathcal{U}$, which contain exactly those atomic propositions that are interpreted as true. We write $I \models \phi$ to state that an interpretation $I \subseteq \mathcal{U}$ is a (classical propositional) model of a formula ϕ. Furthermore, we denote the set of classical models of a formula or theory T by $Mod(T)$.

Equilibrium models arise in the context of the propositional logic of here-and-there, or the logic HT for short (Heyting 1930). We briefly recall here definitions of concepts, as well as properties of the logic HT that are directly relevant to our work. We refer to the papers by Pearce (1997) and Ferraris (2005) for further details.

The logic HT is a logic located between the intuitionistic and the classical logics. Interpretations in the logic HT are pairs $\langle I, J\rangle$ of standard propositional interpretations such that $I \subseteq J$. We write $\langle I, J\rangle \models_{HT} \phi$ to denote that a formula ϕ holds in an interpretation $\langle I, J\rangle$ in the logic HT. The relation $\models_{HT}$ is defined recursively as follows: for an atom a, $\langle I, J\rangle \models_{HT} a$ if and only if $a \in I$. The cases of the boolean connectives $\wedge$ and $\vee$ are standard, and $\langle I, J\rangle \models_{HT} \varphi \rightarrow \psi$ if and only if $J \models \varphi \rightarrow \psi$ (classical satisfiability) and $\langle I, J\rangle \not\models_{HT} \varphi$ or $\langle I, J\rangle \models_{HT} \psi$. Finally, $\langle I, J\rangle \not\models_{HT} \bot$.

An *equilibrium model* or *answer set* of a propositional theory T is a standard interpretation I such that $\langle I, I\rangle \models_{HT} T$ and for every proper subset J of I, $\langle J, I\rangle \not\models_{HT} T$. Answer sets of a propositional theory T are also classical models of T. The converse is not true in general.

We denote the set of all answer sets of a theory T by $AS(T)$, and the set of all HT-models of T by $Mod_{HT}(T)$, that is, $Mod_{HT}(T) = \{\langle I, J\rangle \mid I \subseteq J, \langle I, J\rangle \models_{HT} T\}$.

For each of the semantics there are two natural concepts of equivalence. Two theories T_1 and T_2 are *equivalent* if they have the same models (classical or equilibrium, respectively). They are *strongly equivalent* if for every theory S, $T_1 \cup S$ and $T_2 \cup S$ have the same models (again, classical or equilibrium, respectively).

For classical semantics, strong equivalence and equivalence coincide. It is not so for the semantics of equilibrium models. The result by Lifschitz, Pearce, and Valverde (2001) states that two theories T_1 and T_2 are strongly equivalent for equilibrium models if and only if T_1 and T_2 are equivalent in the logic HT, that is, $Mod_{HT}(T_1) = Mod_{HT}(T_2)$.

Example 1 *Let us consider the theory $T_a = \{a \rightarrow a\}$. The classical models are $Mod(T_a) = \{\emptyset, \{a\}\}$, so a being true and a being false are both admissible scenarios. The HT-models are $\langle\emptyset, \emptyset\rangle$, $\langle\emptyset, \{a\}\rangle$, and $\langle\{a\}, \{a\}\rangle$. Hence, there is only one answer set (equilibrium model) $\emptyset$. The other possible candidate, $\{a\}$, is not an answer set. While $\langle\{a\}, \{a\}\rangle \models_{HT} T$ does hold, also $\langle\emptyset, \{a\}\rangle \models_{HT} T$ does. The intuition here is that the theory does not contain any "cause" for a to hold.*

Next, let us consider the theory $T_b = \{a \vee b\}$. The classical models are $\{a\}$, $\{b\}$ and $\{a, b\}$, and the HT-models are $\langle\{a\}, \{a\}\rangle$, $\langle\{b\}, \{b\}\rangle$, $\langle\{a\}, \{a, b\}\rangle$, $\langle\{b\}, \{a, b\}\rangle$ and $\langle\{a, b\}, \{a, b\}\rangle$. The answer sets are therefore $\{a\}$ and $\{b\}$. Again the the intuition is that the theory does not contain any "cause" for a and b to hold simultaneously.

Finally, let us consider the theory $T_c = \{(\neg a \rightarrow b) \wedge (\neg b \rightarrow a)\}$. The classical models are the same as for T_b, that is, $Mod(T_c) = \{\{a\}, \{b\}, \{a, b\}\}$. We also have $Mod_{HT}(T_c) = \{\langle\{a\}, \{a\}\rangle, \langle\{b\}, \{b\}\rangle, \langle\{a\}, \{a, b\}\rangle, \langle\{b\}, \{a, b\}\rangle, \langle\{a, b\}, \{a, b\}\rangle, \langle\emptyset, \{a, b\}\rangle\} = Mod_{HT}(T_b) \cup \{\langle\emptyset, \{a, b\}\rangle\}$. The answer sets are again the same as for T_b: $AS(T_c) = \{\{a\}, \{b\}\}$. Also here the intuition is that the theory does not contain any "cause" for a and b to hold simultaneously.

We observe that T_b and T_c are equivalent for both the classical and equilibrium setting (they have the same classical and equilibrium models). The former implies that they are also strongly equivalent in the classical setting. However, they are not strongly equivalent in the equilibrium setting because $Mod_{HT}(T_b) \neq Mod_{HT}(T_c)$ and indeed we can see that for $S = \{a \rightarrow b, b \rightarrow a\}$, we obtain $AS(T_b \cup S) = \{\{a, b\}\}$, while $AS(T_c \cup S) = \emptyset$.

We recall that optimization problems under the classical interpretation of generators are referred to as *classical optimization problems* or CO problems, and when we use the answer-set semantics for generators, we speak about *answer-set optimization problems* or ASO problems.

Selectors. We follow the definitions of preference modules in ASO programs (Brewka, Niemelä, and Truszczyński 2003), adjusting the terminology to our more general setting. A *selector* is a finite set of *ranked preference rules*

$$\phi_1 > \cdots > \phi_k \stackrel{j}{\leftarrow} \psi \qquad (1)$$

where k and j are positive integers, and ϕ_i, $1 \leq i \leq k$, and ψ are propositional formulas over $\mathcal{U}$. For a rule r of the form (1), the number j is the *rank* of r, denoted by $rank(r)$, $hd(r) = \{\phi_1, \ldots, \phi_k\}$ is the *head* of r and ψ is the *body* of r, $bd(r)$. Moreover, we write $hd_i(r)$ to refer to formula ϕ_i.

If $rank(r) = 1$ for every preference rule r in a selector S, then S is a *simple selector*. Otherwise, S is *ranked*. We often omit "1" from the notation "$\stackrel{1}{\leftarrow}$" for simple selectors. For a selector S, and $i, j \in \{0, 1, 2, \ldots\} \cup \{\infty\}$, we define $S_{[i,j]} = \{r \in S \mid i \leq rank(r) \leq j\}$ (where we assume that for every integer k, $k < \infty$) and write $[i, j]$ for the *rank interval* $\{k \mid i \leq k \leq j\}$. We extend this notation to optimization problems. For $P = (T, S)$ and a rank interval $[i, j]$, we set $P_{[i,j]} = (T, S_{[i,j]})$. For some rank intervals we use shorthands, for example $= i$ for $[i, i]$, $< i$ for $[1, i-1]$, $\geq i$ for $[i, \infty]$, and similar.

For an interpretation I, a *satisfaction degree* of a preference rule r is $v_I(r) = min\{i \mid I \models hd_i(r)\}$, if $I \models bd(r)$ and $I \models \bigvee hd(r)$; otherwise, the rule is *irrelevant* to I, and $v_I(r) = 1$. We note that Brewka, Niemelä, and Truszczyński (2003) represented the satisfaction degree of an irrelevant rule by a special non-numeric degree, treated as being equivalent to 1. The difference is immaterial and the two approaches are equivalent.

Selectors determine a preference relation on interpretations. Given interpretations I and J and a simple selector S, $I \geq^S J$ holds precisely when for all $r \in S$, $v_I(r) \leq v_J(r)$.

Therefore, $I >^S J$ holds if and only if $I \geq^S J$ and there exists $r \in S$ such that $v_I(r) < v_J(r)$; $I \approx^S J$ if and only if for every $r \in S$, $v_I(r) = v_J(r)$.

Given a *ranked* selector S, we define $I \geq^S J$ if for every preference rule $r \in S$, $v_I(r) = v_J(r)$, or if there is a rule $r' \in S$ such that the following three conditions hold:

1. $v_I(r') < v_J(r')$
2. for every $r \in S$ of the same rank as r', $v_I(r) \leq v_J(r)$
3. for every $r \in S$ of smaller rank than r', $v_I(r) = v_J(r)$.

Moreover, $I >^S J$ if and only if there is a rule r' for which the three conditions above hold, and $I \approx^S J$ if and only if for every $r \in S$, $v_I(r) = v_J(r)$. Given an optimization problem $P = (T, S)$ we often write $\geq^P$ for $\geq^S$ (and similarly for $>$ and $\approx$).

Optimal (preferred) outcomes. For an optimization problem P, $\mu(P)$ denotes the set of all outcomes of P, that is, the set of all models (under the selected semantics) of the generator of P. Thus, $\mu(P)$ stands for all models of P in the framework of CO problems and for all answer sets of P, when ASO problems are considered. A model $I \in \mu(P)$ is *optimal* or *preferred* for P if there is no model $J \in \mu(P)$ such that $J >^P I$. We denote the set of all *preferred models* of P by $\pi(P)$.

Relation to ASO programs. Optimization problems extend the formalism of ASO programs (Brewka, Niemelä, and Truszczyński 2003) in several ways. First, the generator programs are *arbitrary* propositional theories. Under the semantics of equilibrium models, our generators properly extend logic programs with the answer-set semantics used as generators in ASO programs. Second, the selectors use arbitrary propositional formulas in the heads of preference rules, as well as for conditions in their bodies. Finally, optimization problems explicitly allow for alternative semantics of generators, a possibility mentioned but not pursued by Brewka, Niemelä, and Truszczyński (2003).

Notions of Equivalence. We define the union of optimization problems as expected, that is, for $P_1 = (T_1, S_1)$ and $P_2 = (T_2, S_2)$, we set $P_1 \cup P_2 = (T_1 \cup T_2, S_1 \cup S_2)$. Two optimization problems P_1 and P_2 are *strongly equivalent* with respect to a class $\mathcal{R}$ of optimization problems (*contexts*) if for every optimization problem $R \in \mathcal{R}$, $\pi(P_1 \cup R) = \pi(P_2 \cup R)$.

We consider three general classes of contexts. First and foremost, we are interested in the class $\mathcal{L}_\mathcal{U}$ of all optimization problems over $\mathcal{U}$. We also consider the families $\mathcal{L}_\mathcal{U}^g$ and $\mathcal{L}_\mathcal{U}^s$ of all optimization problems of the form $(T, \emptyset)$ and $(\emptyset, S)$, respectively. The first class consists of optimization problems where all models of the generator are equally preferred. We call such optimization problems *generator problems*. The second class consists of optimization problems in which every interpretation of $\mathcal{U}$ is an acceptable outcome. We call such optimization problems *selector problems*. These "one-dimensional" contexts provide essential insights into the general case. For the first class, we simply speak of *strong equivalence*, denoted $\equiv_g^s$. For the latter two classes, we speak of *strong gen-equivalence*, denoted $\equiv_g$, and *strong sel-equivalence*, denoted $\equiv^s$, respectively.

Constraining ranks of rules in selectors gives rise to two additional classes of contexts parameterized using rank intervals $[i, j]$:

1. $\mathcal{L}_\mathcal{U}^{s,[i,j]} = \{(\emptyset, S) \in \mathcal{L}_\mathcal{U}^s \mid S = S_{[i,j]}\}$
2. $\mathcal{L}_\mathcal{U}^{[i,j]} = \{(T, S) \in \mathcal{L}_\mathcal{U} \mid S = S_{[i,j]}\}$

The first class of contexts gives rise to *strong sel-equivalence* with respect to rules of rank in $[i, j]$, denoted by $\equiv^{s,[i,j]}$. The second class of contexts yields the concept of *strong equivalence* with respect to rules of rank in $[i, j]$. We denote it by $\equiv_g^{s,[i,j]}$. We call problems in the class $\mathcal{L}_\mathcal{U}^{=1} = \mathcal{L}_\mathcal{U}^{[1,1]}$ *simple optimization problems*.

Examples

We present now examples that illustrate key issues relevant to strong equivalence of optimization problems. They point to the necessity of some conditions that appear later in characterizations of strong equivalence and hint at some constructions used in proofs. In all examples except for the last one, we consider simple CO problems. In all problems only atoms explicitly listed matter, so we disregard all others.

Example 2 *Let $P_1 = (T_1, S_1)$, where $T_1 = \{a \leftrightarrow \neg b\}$ and $S_1 = \{a > b \leftarrow\}$. There are two outcomes here, $\{a\}$ and $\{b\}$, that is, $\mu(P_1) = \{\{a\}, \{b\}\}$. Let r be the only preference rule in S_1. Clearly, $v_{\{a\}}(r) = 1$ and $v_{\{b\}}(r) = 2$. Thus, $\{a\} >^{P_1} \{b\}$ and so, $\pi(P_1) = \{\{a\}\}$.*

In addition, let $P_2 = (T_2, S_1)$, where $T_2 = \{a \wedge \neg b\}$ and S_1 is as above. Then, $\mu(P_2) = \{\{a\}\}$ and, trivially, $\pi(P_2) = \{\{a\}\}$. It follows that P_1 and P_2 are equivalent, as they specify the same optimal outcomes. However, they are not strongly gen-equivalent (and so, also not strongly equivalent). Indeed, let $R = (\{\neg a\}, \emptyset)$. Then $\mu(P_1 \cup R) = \{\{b\}\}$ and so, $\pi(P_1 \cup R) = \{\{b\}\}$. On the other hand, $\mu(P_2 \cup R) = \emptyset$ and, therefore, $\pi(P_2 \cup R) = \emptyset$.

Example 2 suggests that we must have $\mu(P_1) = \mu(P_2)$ if problems P_1 and P_2 are to be strongly (gen-)equivalent. Otherwise, by properly selecting the context generator, we can eliminate all outcomes in one problem still leaving some in the other.

Example 3 *Let $P_3 = (T_3, S_3)$, where $T_3 = \{a \vee b \vee c, \neg(a \wedge b), \neg(a \wedge c), \neg(b \wedge c)\}$ and $S_3 = \{a > b \leftarrow,\ a > c \leftarrow\}$. We have $\mu(P_3) = \{\{a\}, \{b\}, \{c\}\}$. In addition, $\{a\} >^{P_3} \{b\}$, $\{a\} >^{P_3} \{c\}$, and $\{b\}$ and $\{c\}$ are incomparable. Thus, $\pi(P_3) = \{\{a\}\}$. Let now $P_4 = (T_4, S_4)$, where $T_4 = T_3$ and $S_4 = \{a > b > c \leftarrow\}$. Clearly, $\mu(P_4) = \mu(P_3) = \{\{a\}, \{b\}, \{c\}\}$. Moreover, $\{a\} >^{P_4} \{b\} >^{P_4} \{c\}$. Thus, $\pi(P_4) = \{\{a\}\}$ and so, P_3 and P_4 are equivalent. They are not strongly (gen-)equivalent. Indeed, let $R = (\{\neg a\}, \emptyset)$. Then, $\pi(P_3 \cup R) = \{\{b\}, \{c\}\}$ but $\pi(P_4 \cup R) = \{\{b\}\}$.*

This example suggests that for two optimization problems to be strongly (gen-)equivalent, they have to define the same preference relation $>$ on outcomes.

Example 4 *Let $P_5 = (T_5, S_5)$, where $T_5 = \{a, \neg b\}$ and $S_5 = \emptyset$. We have $\mu(P_5) = \{\{a\}\}$ and so, $\pi(P_5) = \{\{a\}\}$. It follows that P_5 is equivalent to P_1. Let $R = (\emptyset, \{b >$*

$a \leftarrow\}$). Since $\mu(P_5 \cup R) = \mu(P_5) = \{\{a\}\}$, $\pi(P_5 \cup R) = \{\{a\}\}$. Further, $P_1 \cup R = (T_1, \{a > b \leftarrow, b > a \leftarrow\})$ and so, $\mu(P_1 \cup R) = \mu(P_1) = \{\{a\},\{b\}\})$ and we get $\pi(P_1 \cup R) = \{\{a\},\{b\}\}$. Thus, P_5 and P_1 are not strongly (sel-)equivalent.

Informally, this example shows that by modifying the selector part, we can make non-optimal outcomes optimal. Thus, as in the case of strong gen-equivalence (Example 2) the equality of sets of models (i.e., equivalence) is important for strong (sel-)equivalence (a more refined condition will be needed for ranked programs, as we show in Theorem 2).

Example 5 *For the next example, let us consider problems $P_6 = (T_1, S_6)$ and $P_7 = (T_1, S_7)$, where T_1 is the generator from Example 2, $S_6 = \{a > b \leftarrow,\ b > a \leftarrow\}$ and $S_7 = \emptyset$. We have $\mu(P_6) = \mu(P_7) = \{\{a\},\{b\}\}$. Moreover, $\{a\} \geq^{P_6} \{b\}$ and $\{b\} \geq^{P_6} \{a\}$. Thus, $\pi(P_6) = \{\{a\},\{b\}\}$. Since $P_7^s = \emptyset$, we also have (trivially) that $\{a\} \approx^{P_7} \{b\}$. Thus, $\pi(P_7) = \{\{a\},\{b\}\}$, too, and the problems P_6 and P_7 are equivalent. They are not strongly sel-equivalent, though. Let $R = (\emptyset, \{a > b \leftarrow\})$. Then, $P_6 \cup R = P_6$ and so, $\pi(P_6 \cup R) = \{\{a\},\{b\}\}$. On the other hand, $\{a\} >^{P_7 \cup R} \{b\}$. Thus, $\pi(P_7 \cup R) = \{\{a\}\}$.*

The example suggests that for strong sel-equivalence the equality of the relation $\geq$ induced by the problems considered is important. The equality of the relation $>$ is not sufficient. In our example, the relations $>^{P_6}$ and $>^{P_7}$ are both empty and hence equal. But they are empty for different reasons, absence of preference versus conflicting preferences, which can give rise to different preferred models when extending selectors.

Our last example involves ranked problems. It is meant to hint at issues that arise when ranked problems are considered. In the general case of ranked selectors, the equality of the relation $>$ induced by programs being evaluated is not related to strong sel-equivalence in any direct way and an appropriate modification of that requirement has to be used together with yet another condition (cf. Theorem 2).

Example 6 *Let $P = (T_1, S)$, where $S = \{a > b \overset{2}{\leftarrow}\}$ and $P' = (T_1, S')$, where $S' = \{a > b \overset{3}{\leftarrow}\}$. Clearly, $\mu(P) = \mu(P') = \{\{a\},\{b\}\}$ and $\pi(P) = \pi(P') = \{a\}$. Thus, the two problems are equivalent. They are not strongly sel-equivalent if arbitrary selectors are allowed. For instance, let $R = (\emptyset, \{b > a \overset{2}{\leftarrow}\})$. Adding this new preference rule to P makes $\{a\}$ and $\{b\}$ incomparable and so, $\pi(P \cup R) = \{\{a\},\{b\}\}$. On the other hand, since the new rule has rank 2, it dominates the preference rule of P', which is of rank 3. Thus, $\mu(P' \cup R) = \{\{b\}\}$. A similar effect occurs with the problem $R' = (\emptyset, \{b > a \overset{3}{\leftarrow}\})$. Since its only preference rule is dominated by the only preference rule in P, $\pi(P \cup R') = \pi(P) = \{\{a\}\}$. On the other hand, $\{a\}$ and $\{b\}$ are incomparable in $P' \cup R'$ and, consequently, $\pi(P' \cup R') = \{\{a\},\{b\}\}$. Thus, extending P and P' with selectors containing rules of rank 2 or 3 may lead to different optimal outcomes. It is so even though the relations $>$ induced by P and P' on the set of all outcomes coincide.*

However, adding selectors consisting only of rules of rank greater than 3 cannot have such an effect, since the existing rules would dominate them. Also, adding rules of rank 1 cannot result in differing preferred answer sets, as such rules would dominate the existing ones. Formally, the problems P and P' are strongly sel-equivalent relative to selectors with preference rules of rank greater than 3 or less than 2.

Strong sel-equivalence

We start with the case of strong sel-equivalence, the core case for our study. Indeed, characterizations of strong sel-equivalence naturally imply characterizations for the general case thanks to the following simple observation.

Proposition 1 *Let P and Q be optimization problems (either under classical or answer-set semantics for the generators) and $[i,j]$ a rank interval. Then $P \equiv_g^{s,[i,j]} Q$ if and only if for every generator $R \in \mathcal{L}_{\mathcal{U}}^g$, $P \cup R \equiv^{s,[i,j]} Q \cup R$.*

Proof. ($\Rightarrow$) Let $R \in \mathcal{L}_{\mathcal{U}}^g$. Since $P \equiv_g^{s,[i,j]} Q$, $P \cup R \equiv_g^{s,[i,j]} Q \cup R$ and so, $P \cup R \equiv^{s,[i,j]} Q \cup R$.

($\Leftarrow$) Let R be any optimization problem in $\mathcal{L}_{\mathcal{U}}^{[i,j]}$. We have $P \cup R = (P \cup (R^g, \emptyset)) \cup (\emptyset, R^s)$ and $Q \cup R = (Q \cup (R^g, \emptyset)) \cup (\emptyset, R^s)$. Moreover, by the assumption we have that $P \cup (R^g, \emptyset) \equiv^{s,[i,j]} Q \cup (R^g, \emptyset)$. Thus,

$$\pi((P \cup (R^g, \emptyset)) \cup (\emptyset, R^s)) = \pi((Q \cup (R^g, \emptyset)) \cup (\emptyset, R^s)).$$

It follows that $\pi(P \cup R) = \pi(Q \cup R)$ and, consequently, that $P \equiv_g^{s,[i,j]} Q$. $\square$

Furthermore, the set of outcomes of an optimization problem P is unaffected by changes in the selector module. It follows that the choice of the semantics for generators does not matter for characterizations of strong sel-equivalence. Thus, whenever in this section we refer to the set of outcomes of an optimization problem P, we use the notation $\mu(P)$, and not the more specific one, $Mod(P^g)$ or $AS(P^g)$, that applies to CO and ASO problems, respectively.

Our first main result concerns strong sel-equivalence relative to selectors consisting of preference rules of ranks in a rank interval $[i,j]$. Special cases for strong sel-equivalence will follow as corollaries. To state the result, we need some auxiliary notation. For an optimization problem P, we define $\mathit{diff}^P(I,J)$ to be the largest k such that $I \approx^{P_{<k}} J$. If for every k we have $I \approx^{P_{<k}} J$, then we set $\mathit{diff}^P(I,J) = \infty$. It is clear that $\mathit{diff}^P(I,J)$ is well-defined. Moreover, as $I \approx^{P_{<1}} J$, $\mathit{diff}^P(I,J) \geq 1$. Furthermore, for a set $V \subseteq 2^{\mathcal{U}}$ and a relation $\succ$ over $2^{\mathcal{U}}$, we write $\succ_V$ for the restriction of $\succ$ to V, that is, $\succ_V = \{(A,B) \in \succ \mid A, B \in V\}$.

Theorem 2 *For all ranked optimization problems P and Q, and every rank interval $[i,j]$, $P \equiv^{s,[i,j]} Q$ if and only if the following conditions hold:*

1. *$\pi(P_{<i}) = \pi(Q_{<i})$*
2. *$>^P_{\pi(P_{<i})} = >^Q_{\pi(Q_{<i})}$*
3. *For every $I, J \in \pi(P_{<i})$ such that $i < \mathit{diff}^P(I,J)$ or $i < \mathit{diff}^Q(I,J)$, $\mathit{diff}^P(I,J) = \mathit{diff}^Q(I,J)$ or both $\mathit{diff}^P(I,J) > j$ and $\mathit{diff}^Q(I,J) > j$.*

We now comment on this characterization and derive some of its consequences.

First, we observe that conditions (1) and (2) are indeed necessary — differences between $\pi(P_{<i})$ and $\pi(Q_{<i})$ or $>^P_{\pi(P_{<i})}$ and $>^Q_{\pi(Q_{<i})}$ can be exploited to construct a selector from $\mathcal{L}^{s,[i,j]}_{\mathcal{U}}$ whose addition to P and Q results in problems with different sets of optimal outcomes. This is illustrated by Examples 4 and 5 in the case of simple problems and simple contexts ($i = j = 1$), where $\pi(P_{<i})$ and $\pi(Q_{<i})$ coincide with $\mu(P)$ and $\mu(Q)$, respectively.

Condition (3) is necessary, too. Intuitively, if the first ranks where P and Q differentiate between two outcomes I and J (which are optimal for ranks less than i) are not equal, these first ranks must both be larger than j. Otherwise, one can find a selector with rules of ranks in $[i, j]$, that will make one of the interpretation optimal in one extended problem but not in the other.

Next, we discuss some special cases of the characterization. First, we consider the case $i = 1$, which allows for a simplification of Theorem 2.

Corollary 3 *For all ranked optimization problems P and Q, and every rank interval $[1, j]$, $P \equiv^{s,[1,j]} Q$ if and only if the following conditions hold:*

1. $\mu(P) = \mu(Q)$
2. $>^P_{\mu(P)} = >^Q_{\mu(Q)}$
3. *For every $I, J \in \mu(P)$, $\mathit{diff}^P(I, J) = \mathit{diff}^Q(I, J)$ or both $\mathit{diff}^P(I, J) > j$ and $\mathit{diff}^Q(I, J) > j$.*

Proof. Starting from Theorem 2, we note that the selector of $P_{<1}$ is empty and hence $\pi(P_{<1}) = \mu(P)$. Moreover, if the precondition $i < \mathit{diff}^P(I, J)$ and $i < \mathit{diff}^Q(I, J)$ in condition (3) of Theorem 2 is not satisfied for $i = 1$ and a pair $I, J \in \mu(P)$, then $\mathit{diff}^P(I, J) = 1$ and $\mathit{diff}^Q(I, J) = 1$ and thus the consequent is satisfied in that case as well, which allows for omitting the precondition. □

If in addition $j = \infty$, we obtain the case of rank-unrestricted selector contexts, and condition (3) can be simplified once more, since $\mathit{diff}^P(I, J) > j$ and $\mathit{diff}^Q(I, J) > j$ never hold for $j = \infty$.

Corollary 4 *For all optimization problems P and Q, $P \equiv^s Q$ (equivalently, $P \equiv^{s,\geq 1} Q$ or $P \equiv^{s,[1,\infty]} Q$) if and only if the following conditions hold:*

1. $\mu(P) = \mu(Q)$
2. $>^P_{\mu(P)} = >^Q_{\mu(Q)}$
3. *for every $I, J \in \mu(P)$, $\mathit{diff}^P(I, J) = \mathit{diff}^Q(I, J)$.*

Next, we note that if an optimization problem is simple then $\mathit{diff}^P(I, J) > 1$ if and only if $\mathit{diff}^P(I, J) = \infty$, which is equivalent to $I \approx^P J$. This observation leads to the following characterization of strong sel-equivalence of simple optimization problems.

Corollary 5 *For all* simple *optimization problems P and Q, the following statements are equivalent:*

(a) $P \equiv^s Q$ (equivalently, $P \equiv^{s,[1,\infty]} Q$)

(b) $P \equiv^{s,=1} Q$ (equivalently, $P \equiv^{s,[1,1]} Q$)

(c) $\mu(P) = \mu(Q)$ and $\geq^P_{\mu(P)} = \geq^Q_{\mu(Q)}$.

Proof. The implication (a)⇒(b) is evident from the definitions.

(b)⇒(c) From Corollary 3 with $j = 1$ we directly obtain $\mu(P) = \mu(Q)$. The condition $\geq^P_{\mu(P)} = \geq^Q_{\mu(Q)}$ follows from conditions (2) and (3) of that corollary. Indeed, let us consider $I, J \in \mu(P)$ such that $I \geq^P J$ and distinguish two cases. If (i) $\mathit{diff}^P(I, J) = 1$ then $I >^P J$ and by condition (2) of Corollary 3, also $I >^Q J$, implying $I \geq^Q J$. If (ii) $\mathit{diff}^P(I, J) > 1$ then by condition (3) of Corollary 3, $\mathit{diff}^Q(I, J) > 1$. Since P, Q are simple, $I \approx^Q J$, and consequently $I \geq^Q J$. By symmetry, we also have that $I \geq^Q J$ implies $I \geq^P J$. Thus, $\geq^P_{\mu(P)} = \geq^Q_{\mu(Q)}$.

(c)⇒(a) From (c) it follows that $>^P_{\mu(P)} = >^Q_{\mu(Q)}$ and $\approx^P_{\mu(P)} = \approx^Q_{\mu(Q)}$. Thus, conditions (1) and (2) of Corollary 4 follow. To prove condition (3), let us first assume $\mathit{diff}^P(I, J) > 1$ for $I, J \in \mu(P)$. It follows that $\mathit{diff}^P(I, J) = \infty$ and thus $I \approx^P_{\mu(P)} J$. By our earlier observation also $I \approx^Q_{\mu(Q)} J$ and thus $\mathit{diff}^Q(I, J) = \infty$. Hence $\mathit{diff}^P(I, J) = \mathit{diff}^Q(I, J)$. For $\mathit{diff}^Q(I, J) > 1$ we reason analogously. In the last remaining case, $\mathit{diff}^P(I, J) = 1$ and $\mathit{diff}^Q(I, J) = 1$, so we directly obtain $\mathit{diff}^P(I, J) = \mathit{diff}^Q(I, J)$. By Corollary 4, $P \equiv^s Q$ follows. □

Corollary 5 shows, in particular, that for simple problems there is no difference between the relations $\equiv^{s,\geq 1}$ and $\equiv^{s,=1}$. This property reflects the role of preference rules of rank 2 and higher. They allow us to break ties among optimal outcomes, as defined by preference rules of rank 1. Thus, they can eliminate some of these outcomes from the family of optimal ones, but they cannot introduce new optimal outcomes. Therefore, they do not affect strong sel-equivalence of simple problems. This property has the following generalization to ranked optimization problems.

Corollary 6 *Let P and Q be ranked optimization problems and let k be the maximum rank of a preference rule in $P \cup Q$. Then the relations $\equiv^{s,\geq k}$ and $\equiv^{s,=k}$ coincide.*

Proof. Clearly, $P \equiv^{s,\geq k} Q$ implies $P \equiv^{s,=k} Q$. Thus, it is enough to prove that if $P \equiv^{s,=k} Q$ then $P \equiv^{s,\geq k} Q$. Using the characterization of Theorem 2, we observe that conditions (1) and (2) are equal for $P \equiv^{s,=k} Q$ and $P \equiv^{s,\geq k} Q$. Condition (3) for $P \equiv^{s,=k} Q$ reads "For every $I, J \in \pi(P_{<k})$ such that $k < \mathit{diff}^P(I, J)$ or $k < \mathit{diff}^Q(I, J)$, $\mathit{diff}^P(I, J) = \mathit{diff}^Q(I, J)$ or both $\mathit{diff}^P(I, J) > k$ and $\mathit{diff}^Q(I, J) > k$." while for $P \equiv^{s,\geq k} Q$ it reads "For every $I, J \in \pi(P_{<k})$ such that $k < \mathit{diff}^P(I, J)$ or $k < \mathit{diff}^Q(I, J)$, $\mathit{diff}^P(I, J) = \mathit{diff}^Q(I, J)$ or both $\mathit{diff}^P(I, J) > \infty$ and $\mathit{diff}^Q(I, J) > \infty$.". Clearly, neither $\mathit{diff}^P(I, J) > \infty$ nor $\mathit{diff}^Q(I, J) > \infty$ can ever hold, so we have to show that $\mathit{diff}^P(I,J) = \mathit{diff}^Q(I,J)$ holds in all cases. Let us consider $I, J \in \pi(P_{<k})$ such that $\mathit{diff}^P(I,J) > k$, from the assumption $P \equiv^{s,=k} Q$ we know $\mathit{diff}^Q(I,J) > k$. Since k is the maximum rank of a preference rule in P or Q, $\mathit{diff}^P(I,J) = \infty$ and $\mathit{diff}^Q(I,J) = \infty$. Thus, $\mathit{diff}^P(I,J) = \mathit{diff}^Q(I,J)$ (the case $\mathit{diff}^Q(I,J) > k$ is similar). □

Our observation on the role of preference rules with ranks higher than ranks of rules in P or Q also implies that P and Q are strongly sel-equivalent relative to selectors consisting exclusively of such rules if and only if P and Q are equivalent (have the same optimal outcomes), and if optimal outcomes that "tie" in P also "tie" in Q and conversely. Formally, we have the following result.

Corollary 7 *Let P and Q be ranked optimization problems and let k be the maximum rank of a preference rule in $P \cup Q$. Then $P \equiv^{s,\geq k+1} Q$ if and only if $\pi(P) = \pi(Q)$ and $\approx^P_{\pi(P)} = \approx^Q_{\pi(Q)}$.*

Proof. Clearly, $P_{<k+1} = P$ and $Q_{<k+1} = Q$ and so, $\pi(P_{<k+1}) = \pi(P)$ and $\pi(Q_{<k+1}) = \pi(Q)$. Thus, the "only-if" part follows by Theorem 2 (condition (1) of that theorem reduces to $\pi(P) = \pi(Q)$ and condition (3) implies $\approx^P_{\pi(P)} = \approx^Q_{\pi(Q)}$). To prove the "if" part, we note that condition (1) of Theorem 2 holds by the assumption. Moreover, the relations $>^P_{\pi(P)}$ and $>^Q_{\pi(Q)}$ are empty and so, they coincide. Thus, condition (2) of Theorem 2 holds. Finally, if $I, J \in \pi(P)$, and $\mathit{diff}^P(I,J) > k+1$, then $\mathit{diff}^P(I,J) = \infty$ and so, $I \approx^P J$. By the assumption, $I \approx^Q J$, that is, $\mathit{diff}^Q(I,J) = \infty = \mathit{diff}^P(I,J)$. The case when $\mathit{diff}^Q(I,J) > k+1$ is similar. Thus, condition (3) of Theorem 2 holds, too, and $P \equiv^{s,\geq k+1} Q$ follows. □

Lastly, we give some simple examples illustrating how our results can be used to "safely" modify or simplify optimization problems, that is rewrite one into another strongly sel-equivalent one.

Example 7 *Let $P = (T, S)$, where $T = \{a \vee b \vee c, \neg(a \wedge b), \neg(a \wedge c), \neg(b \wedge c)\}$ and $S = \{a > c \leftarrow, b > c \leftarrow\}$, and $P' = (T, S')$, where $S' = \{a \vee b > c \leftarrow\}$. Regarding these problems as CO problems, we have that $\mu(P) = \mu(P') = \{\{a\}, \{b\}, \{c\}\}$. Moreover, it is evident that $\geq^P_{\mu(P)} = \geq^{P'}_{\mu(P')}$. Thus, by Corollary 5, P and P' are strongly sel-equivalent. In other words, we can faithfully replace rules $a > c \leftarrow$, $b > c \leftarrow$ in the selector of any optimization problem with generator T by the single rule $a \vee b > c \leftarrow$.*

Next, for an example of a more general principle, we note that removing preference rules with only one formula in the head yields a problem that is strongly sel-equivalent.

Corollary 8 *Let P and Q be two CO or ASO problems such that $P^g = Q^g$ and Q^s is obtained from P^s by removing all preference rules with only one formula in the head (i.e., rules r for which $|hd(r)| = 1$). Then P and Q are strongly sel-equivalent.*

Proof. Conditions (1)-(3) of Theorem 2 all follow from an observation that for every interpretation I and every preference rule r with $|hd(r)| = 1$, $v_I(r) = 1$. □

Strong gen-equivalence

We now focus on the case of strong gen-equivalence. The semantics of generators makes a difference here but the difference concerns only the fact that under the two semantics we consider, the concepts of strong equivalence are different. Other aspects of the characterizations are the same. Specifically, generators have to be strongly equivalent relative to a selected semantics. Indeed, if the generators are not strongly equivalent, one can extend them uniformly so that after the extension one problem has a single outcome, which is then trivially an optimal one, too, while the other one has no outcomes and so, no optimal ones. Second, the preference relation $>$ defined by the selectors of the problems considered must coincide. Thus, a single theorem handles both types of problems.

Theorem 9 *For all CO (ASO, respectively) problems P and Q, $P \equiv_g Q$ if and only if P^g and Q^g are strongly equivalent (that is, $Mod(P^g) = Mod(Q^g)$ for CO problems, and $Mod_{HT}(P^g) = Mod_{HT}(Q^g)$ for ASO problems) and $>^P_{Mod(P^g)} = >^Q_{Mod(Q^g)}$.*

In view of Examples 2 and 3, the result is not unexpected. The two examples demonstrated that the conditions of the characterization cannot, in general, be weakened.

It is clear from Corollary 4 and Theorem 9 that strong sel-equivalence of CO problems is a stronger property than their strong gen-equivalence.

Corollary 10 *For all CO problems P and Q, $P \equiv^s Q$ implies $P \equiv_g Q$.*

In general the implication in Corollary 10 cannot be reversed. The problems P_6 and P_7 considered in Example 5 are not strongly sel-equivalent. However, based on Theorem 9, they are strongly gen-equivalent. Indeed, $Mod(P^g_6) = Mod(P^g_7)$ and, writing M for $Mod(P^g_6) = Mod(P^g_7)$, the relations $>^{P_6}_M$ and $>^{P_7}_M$ are both empty and so, equal.

The relation between strong sel-equivalence and strong gen-equivalence of ASO problem is more complex. In general, neither property implies the other even if both problems P and Q are assumed to be simple. It is so because $P \equiv^s Q$ if and only if $AS(P^g) = AS(Q^g)$ and $\geq^P_{AS(P^g)} = \geq^Q_{AS(Q^g)}$ (Corollary 5), and $P \equiv^g Q$ if and only of $Mod_{HT}(P^g) = Mod_{HT}(Q^g)$ and $>^P_{Mod(P^g)} = >^Q_{Mod(Q^g)}$ (Theorem 9). Now, $AS(P^g) = AS(Q^g)$ (regular equivalence of programs) does not imply $Mod_{HT}(P^g) = Mod_{HT}(Q^g)$ (strong equivalence) and $>^P_{Mod(P^g)} = >^Q_{Mod(Q^g)}$ does not imply $\geq^P_{AS(P^g)} = \geq^Q_{AS(Q^g)}$.

Strong equivalence — the combined case

Finally, we consider the relation $\equiv^s_g$, which results from considering contexts that combine both generators and selectors. Since generators may vary here, as in the previous section, the semantics of generators matters. But, as in the previous section, the difference boils down to different characterizations of strong equivalence of generators.

We start with a result characterizing strong equivalence of CO and ASO problems relative to combined contexts (both generators and selectors possibly non-empty) with selectors consisting of rules of rank at least i and at most j, respectively.

Theorem 11 *For all ranked CO (ASO, respectively) problems P and Q, and every rank interval $[i, j]$, $P \equiv_g^{s,[i,j]} Q$ if and only if the following conditions hold:*

1. *P^g and Q^g are strongly equivalent (that is, $Mod(P^g) = Mod(Q^g)$ for CO problems, and $Mod_{HT}(P^g) = Mod_{HT}(Q^g)$ for ASO problems)*
2. *$>^P_{Mod(P^g)} = >^Q_{Mod(Q^g)}$*
3. *For every $I, J \in Mod(P^g)$ such that $i < \mathit{diff}^P(I,J)$ or $i < \mathit{diff}^Q(I,J)$, $\mathit{diff}^P(I,J) = \mathit{diff}^Q(I,J)$ or both $\mathit{diff}^P(I,J) > j$ and $\mathit{diff}^Q(I,J) > j$*
4. *$>^{P_{<i}}_{Mod(P^g)} = >^{Q_{<i}}_{Mod(Q^g)}$.*

The corresponding characterizations for CO and ASO problems differ only in their respective conditions (1), which now reflect different conditions guaranteeing strong equivalence of generators under the classical and answer-set semantics. Moreover, the four conditions of Theorem 11 can be obtained by suitably combining and extending the conditions of Theorem 2 and Theorem 9. First, as combined strong equivalence implies strong gen-equivalence, condition (1) is taken from Theorem 9. Second, we modify conditions (2) and (3) from Theorem 2 replacing $\pi(P_{<i})$ with $Mod(P^g)$ (and accordingly $\pi(Q_{<i})$ with $Mod(Q^g)$), as each classical model of P^g can give rise to an optimal classical or equilibrium one upon the addition of a context, an aspect also already visible in Theorem 9. Finally, we have to add a new condition stating that the relations $>^{P_{<i}}$ and $>^{Q_{<i}}$ coincide on the sets of models of P^g and Q^g. When generators are allowed to be extended, one can make any two of their models to be the only outcomes after the extension. If the two outcomes, say I and J, are related differently by the corresponding strict relations induced by rules with ranks less than i, then in one extended problem, exactly one of the two outcomes, say I, is optimal. In the other extended problem we cannot have both outcomes be optimal nor J only be optimal as that would contradict the strong equivalence of problems under considerations. If, however, I is the only optimal outcome also in the other extended problem, then I must "win" with J based on rules that have ranks at least j. In such case, there is a way to add new preferences of rank i that will "promote" J to be optimal too, without making it optimal in the first problem. However, that contradicts strong equivalence.

We conclude this section with observations concerning the relation $\equiv_g^s$ for both CO and ASO problems. The contexts relevant here may contain preference rules of arbitrary ranks. We start with the case of CO problems, where the results are stronger. While they can be derived from the general theorems above, we will present here arguments relying on results from previous sections, which is possible since for CO problems equivalence and strong-equivalence of generators coincide.

We saw in the last section that for CO problems $\equiv^s$ is a strictly stronger relation than $\equiv^g$. In fact, for CO problems, $\equiv^s$ coincides with the general relation $\equiv_g^s$.

Theorem 12 *For all CO problems P and Q, $P \equiv_g^s Q$ if and only if $P \equiv^s Q$.*

Proof. The "only-if" implication is evident. To prove the converse implication, we will use Proposition 1 which reduces checking for strong equivalence to checking for strong sel-equivalence. Let $R \in \mathcal{L}_{\mathcal{U}}^g$ be a generator problem. Since $P \equiv^s Q$, from Corollary 4 we have $Mod(P^g) = Mod(Q^g)$. Consequently, $Mod((P \cup R)^g) = Mod((Q \cup R)^g)$. Writing M for $Mod(P^g)$ and M' for $Mod((P \cup R)^g)$ we have $M' \subseteq M$. Thus, also by Corollary 4, $>^{P \cup R}_{M'} = >^{Q \cup R}_{M'}$. Finally, condition (3) of Corollary 4 for P and Q implies condition (3) of that corollary for $P \cup R$ and $Q \cup R$ (as R has no preference rules and $M' \subseteq M$). It follows, again by Corollary 4, that $P \cup R \equiv^s Q \cup R$. Thus, by Proposition 1, $P \equiv_g^s Q$. □

In particular, Corollary 5 implies that the relations $\equiv_g^s$, $\equiv_g^{s,=1}$, $P \equiv^{s,=1} Q$, and $\equiv^s$ coincide on simple CO problems.

Corollary 13 *For all simple CO problems P and Q all properties $P \equiv_g^s Q$, $P \equiv_g^{s,=1} Q$, $P \equiv^{s,=1} Q$ and $P \equiv^s Q$ are equivalent.*

For simple ASO problems we still have that $\equiv_g^s$ and $\equiv_g^{s,=1}$ coincide but in general these notions are different from $\equiv^s$ and $\equiv^{s,=1}$.

Corollary 14 *For all simple ASO problems P and Q, the following conditions are equivalent*

(a) $P \equiv_g^s Q$

(b) $P \equiv_g^{s,=1} Q$

(c) $Mod_{HT}(P^g) = Mod_{HT}(Q^g)$ *and* $\geq^P_{Mod(P^g)} = \geq^Q_{Mod(Q^g)}$.

Proof. The implication (a)⇒(b) is evident.

Let us assume (b). By Theorem 11, we have $Mod_{HT}(P^g) = Mod_{HT}(Q^g)$. This identity implies $Mod(P^g) = Mod(Q^g)$. Let us assume that for some $I, J \in Mod(P^g)$, $I \geq^P_{Mod(P^g)} J$. If $I >^P_{Mod(P^g)} J$ then, by Theorem 11, $I >^Q_{Mod(Q^g)} J$ and so, $I \geq^Q_{Mod(Q^g)} J$. Otherwise, $I \approx^P J$ and so, $\mathit{diff}^P(I,J) = \infty$. By Theorem 11, $\mathit{diff}^Q(I,J) > 1$. Since Q is simple, $\mathit{diff}^Q(I,J) = \infty$. Thus, $I \approx^Q J$ and, also, $I \geq^Q_{Mod(Q^g)} J$. The converse implication follows by symmetry. Thus, (c) holds.

Finally, we assume (c) and prove (a). To this end, we show that conditions (1)–(4) of Theorem 11 hold. Directly from the assumptions, we have that condition (1) holds. Condition (2) follows from the general fact that for all optimization problems P and Q, and every set $V \subseteq 2^{\mathcal{U}}$, $\geq^P_V = \geq^Q_V$ implies $>^P_V = >^Q_V$. Moreover, we also have that

$Mod(P^g) = Mod(Q^g)$. To prove condition (3), let us assume that $I, J \in Mod(P^g)$ and that $\mathit{diff}^P(I, J) > 1$. Since P is simple, $I \approx^P J$. Thus, $I \approx^Q J$ and, consequently, $\mathit{diff}^P(I, J) = \infty = \mathit{diff}^Q(I, J)$. Finally, condition (4), i.e. $>^{P_{<i}}_{Mod(P^g)} = >^{Q_{<i}}_{Mod(Q^g)}$, obviously holds in case $i = 1$ and $Mod(P^g) = Mod(Q^g)$. □

Complexity

In this section, we study the problems of deciding the various notions of strong equivalence. Typically the comparisons between sets of outcomes in the characterizations determine the respective complexity. We start with results concerning strong sel-equivalence.

Theorem 15 *Given optimization problems P and Q, deciding $P \equiv^s Q$ is* co-NP*-complete in case of CO-problems and* Π^P_2*-complete in case of ASO-problems.*

Proof. [Sketch] For membership, one can show that given a pair of interpretations I, J it can be verified in polynomial time (for CO-problems) or in polynomial time using an NP oracle (for ASO-problems) whether they form a witness for the complement of the conditions stated in Corollary 4. The main observation is that model checking is polynomial for the classical semantics, but co-NP-complete for the equilibrium semantics (Theorem 8 (Pearce, Tompits, and Woltran 2009)).

Hardness follows from considering the equivalence problem for optimizations problems with empty selectors, which is known to be co-NP-hard (for classical semantics) and Π^P_2-hard (for equilibrium semantics, Theorem 11 (Pearce, Tompits, and Woltran 2009)). □

For the ranked case, we observe an increase in complexity, which can be explained by the characterization given in Theorem 2: Instead of outcome checking, this characterization involves optimal outcome checking, which is more difficult (unless the polynomial hierarchy collapses).

Theorem 16 *Given optimization problems P and Q and a rank interval $[i, j]$, deciding $P \equiv^{s,[i,j]} Q$ is Π^P_2-complete in case of CO-problems and Π^P_3-complete in case of ASO-problems.*

Proof. [Sketch] The membership part essentially follows the same arguments as the proof of Theorem 15, but here the problem of checking $I \in \pi(P_{<i})$ is in co-NP for CO-problems and in Π^P_2 for ASO-problems.

For the hardness part, we reduce the following problem to sel-equivalence of CO-problems: Given two propositional theories S and T, decide whether they possess the same minimal models. This problem is known to be Π^P_2-complete (e.g. Theorem 6.15 (Eiter, Fink, and Woltran 2007)), and the problem remains hard if S and T are in negation normal form (NNF) given over the same alphabet. We adapt a construction used by Brewka et al. (2011), and given a theory T (where U is the collection of atoms occurring in T) we construct a CO problem P_T where

$$\begin{aligned} P^g_T &= T[\neg u/u'] \cup \{u \leftrightarrow \neg u' \mid u \in U\}, \\ P^s_T &= \{u' > u \leftarrow \mid u \in U\}, \end{aligned}$$

and $T[\neg u/u']$ stands for the theory resulting from replacing all $\neg u$ by u' in T. The elements in $\pi(P_T)$ are in a one-to-one correspondence to the minimal models of T. For theories S and T over U it follows that S and T have the same minimal models if and only if $P_S \equiv^{s,\geq 2} P_T$.

Concerning the hardness part for ASO problems, we use the following problem: given two open quantified boolean formulas (QBFs) $\forall Y \phi(X, Y)$, $\forall Y \psi(X, Y)$, do they possess the same minimal models. This problem is Π^P_3-hard. For $\phi(X, Y)$, we construct P_ϕ as follows:

$$\begin{aligned} P^g_\phi &= \{z \vee z' \mid z \in X \cup Y\} \cup \\ &\quad \{(y \wedge y') \to w, w \to y, w \to y' \mid y \in Y\} \cup \\ &\quad \{\phi[\neg z/z'] \to w, \neg w \to w\}, \\ P^s_\phi &= \{x' > x \leftarrow \mid x \in X\}, \end{aligned}$$

where $\phi[\neg z/z']$ stands for the formula obtained by replacing all $\neg z$ by z' in $\phi(X, Y)$. The elements in $\pi(P_\phi)$ are in a one-to-one correspondence to the minimal models of $\forall Y \phi(X, Y)$. For ϕ and ψ over $X \cup Y$ we get that $\forall Y \phi(X, Y)$ and $\forall Y \psi(X, Y)$ have the same minimal models if and only if $P_\phi \equiv^{s,\geq 2} P_\psi$. □

In Theorem 16 the rank interval $[i, j]$ is given in input. When fixing the interval, the hardness results still hold, provided that $i > 1$. In fact, the critical condition in Corollary 3 is $\pi(P_{<i}) = \pi(Q_{<i})$; for rank intervals $[1, j]$, the selectors become empty and the condition is reduced to $\mu(P) = \mu(Q)$, which is easier to decide.

The remaining problems are all in co-NP. For strong gen-equivalence, completeness follows directly from Theorem 9 and co-NP-completeness of deciding strong equivalence between two propositional theories (for both semantics).

Theorem 17 *Given two CO (ASO, respectively) problems P and Q, deciding $P \equiv_g Q$ is* co-NP*-complete.*

Finally, for the combined case the hardness result follows from Theorem 11 and co-NP-completeness of deciding strong equivalence of propositional theories.

Theorem 18 *Given ranked CO (ASO, respectively) problems P and Q, and rank interval $[i, j]$, deciding $P \equiv^{s,[i,j]}_g Q$ is* co-NP*-complete.*

By construction, all hardness results hold already for simple optimization problem.

Discussion

We introduced the formalism of optimization problems, generalizing the principles of ASO programs, in particular, the separation of hard and soft constraints (Brewka, Niemelä, and Truszczyński 2003). We focused on two important specializations of optimization problems: CO problems and ASO problems. We studied various forms of strong equivalence for these classes of optimization problems, depending on what contexts are considered. Specifically, we considered the following cases: new preference information is added, but the hard constraints remain unchanged (strong sel-equivalence); hard constraints are added but preferences remain unchanged (strong gen-equivalence); both hard constraints and preferences can be added (strong equivalence). To the best of our knowledge, this natural classification of equivalences in preference formalisms has not been studied yet. In certain cases some of these notions coincide (Theorem 12) but this is no longer true when the underlying semantics is changed or ranks in contexts are restricted.

In previous work, the notion of strong equivalence (both hard constraints and preferences can be added) has been studied for logic programs with weak constraints by Eiter et al. (2007) and logic programs with ordered disjunctions (LPODs) by Faber et al. (2008). While for the former formalism, a separation of strong equivalence into different notions – as suggested here for ASO problems – would be possible (also compare Lemma 23 (Eiter et al. 2007) to our results, e.g. Corollary 14), a similar separation for strong equivalence is not straightforward for LPODs. The reason is the syntactic nature of LPOD rules which act like hard constraints and preference rules at the same time. Faber et al. (2008) considered strong equivalence with respect to contexts that are logic programs (which is similar to strong gen-equivalence) and the combined case of strong equivalence (called strong equivalence for arbitrary contexts there), but they did not consider any counterpart to the notion of strong sel-equivalence. In fact, it is even unclear whether in every LPOD the generating and selecting modules can be cleanly separated.

In our paper, we established characterizations of all three types of strong equivalence. They exhibit strong similarities. The characterizations of strong sel-equivalence for CO and ASO problems in Theorem 2 are precisely the same, mirroring the fact that generators are not subject to change. Theorem 9 concerns strong gen-equivalence for CO and ASO problems. In each case, the characterizations consist of two requirements: the strong equivalence of generators, and the equality of the strict preference relations restricted to the class of models of the generators. The only difference comes from the fact that strong equivalence for classical and the equilibrium-model semantics have different characterizations. Theorem 11 which concerns the combined case of strong equivalence also does not differentiate between CO and ASO problems other than implicitly (as before, the conditions of strong equivalence are different for the two semantics). Moreover, the characterizations given arise in a certain systematic way from those given in Theorems 2 and 9. This being the case in each of the different semantics we used strongly suggests that there are some abstract principles at play here. We are currently pursuing this direction, conjecturing that this is an inherent feature of preference formalisms with separation of hard and soft constraints.

Coming back to LPODs, these comments suggest that identifying a "split" representation for that formalism might be of interest. It could lead to alternative characterizations of (combined) strong equivalence derived from the characterizations of the two "one-dimensional" variants.

Next, we note that our results give rise to problem rewriting methods that transform optimization problems into strongly equivalent ones. We provided two simple examples illustrating that application of our results in Example 7 and Corollary 8. Similar examples can be constructed for our results concerning strong gen-equivalence and (combined) strong equivalence. A more systematic study of optimization problem rewriting rules that result in strongly equivalent problems will be a subject of future work.

Finally, we established the complexity of deciding whether optimization problems are strongly equivalent. Notably, in the general case of strong (combined) equivalence this problem remains in co-NP for both CO and ASO problems. It is strong sel-equivalence that is computationally hardest to test (in case of ASO problems, Π_3^P-hard). It is so, as the concept depends of properties of outcomes that are optimal with respect to rules of ranks less than i, while in other cases all models have to be considered. Testing optimality is harder than testing for being a model, explaining the results we obtained.

Acknowledgments

The first author was supported by Regione Calabria and EU under POR Calabria FESR 2007-2013 within the PIA project of DLVSYSTEM s.r.l., and by MIUR under the PRIN project LoDeN. The second author was supported by the NSF grant IIS-0913459.

References

Brewka, G.; Niemelä, I.; and Truszczyński, M. 2003. Answer set optimization. In Gottlob, G., and Walsh, T., eds., *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico*, 867–872. Morgan Kaufmann.

Brewka, G.; Niemelä, I.; and Truszczyński, M. 2011. Answer set optimization. Unpublished manuscript.

Eiter, T.; Faber, W.; Fink, M.; and Woltran, S. 2007. Complexity results for answer set programming with bounded predicate arities and implications. *Annals of Mathematics and Artificial Intelligence* 51(2–4):123–165.

Eiter, T.; Fink, M.; and Woltran, S. 2007. Semantical characterizations and complexity of equivalences in answer set programming. *ACM Transactions on Computational Logic* 8(3).

Faber, W., and Konczak, K. 2006. Strong order equivalence. *Annals of Mathematics and Artificial Intelligence* 47(1–2):43–78.

Faber, W.; Tompits, H.; and Woltran, S. 2008. Notions of strong equivalence for logic programs with ordered disjunction. In *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning*, 433–443. AAAI Press.

Ferraris, P. 2005. Answer sets for propositional theories. In Baral, C.; Greco, G.; Leone, N.; and Terracina, G., eds., *Logic Programming and Nonmonotonic Reasoning, 8th International Conference, LPNMR 2005, Diamante, Italy, September 2005, Proceedings*, volume 3662 of *Lecture Notes in Computer Science*, 119–131. Springer Verlag.

Gelfond, M., and Lifschitz, V. 1991. Classical negation in logic programs and disjunctive databases. *New Generation Computing* 9:365–385.

Goldsmith, J., and Junker, U., eds. 2008. *Special Issue on Preferences*, volume 29(4) of *AI Magazine*.

Heyting, A. 1930. Die formalen Regeln der intuitionistischen Logik. *Sitzungsberichte der Preussischen Akademie der Wissenschaften* 42–56.

Lifschitz, V.; Pearce, D.; and Valverde, A. 2001. Strongly equivalent logic programs. *ACM Transactions on Computational Logic* 2(4):526–541.

Marek, V. W., and Truszczyński, M. 1993. *Nonmonotonic Logic; Context-Dependent Reasoning*. Berlin: Springer.

Pearce, D.; Tompits, H.; and Woltran, S. 2009. Characterising equilibrium logic and nested logic programs: Reductions and complexity. *Theory and Practice of Logic Programming* 9(5):565–616.

Pearce, D. 1997. A new logical characterisation of stable models and answer sets. In Dix, J.; Pereira, L. M.; and Przymusinski, T., eds., *Non-Monotonic Extensions of Logic Programming*, volume 1216 of *Lecture Notes in Computer Science*, 57–70. Springer.

Turner, H. 2003. Strong equivalence made easy: Nested expressions and weight constraints. *Theory and Practice of Logic Programming* 3(4–5):609–622.