Modeling Stable Matching Problems with Answer Set Programming

Sofie De Clercq[a], Steven Schockaert[b], Martine De Cock[a] & Ann Nowé[c]

[a] Department of Applied Mathematics, Computer Science & Statistics, Ghent University, Krijgslaan 281, 9000 Ghent, Belgium {Sofier.DeClercq,Martine.DeCock}@ugent.be

[b] School of Computer Science & Informatics, Cardiff University, 5 The Parade, Roath, Cardiff CF24 3AA, United Kingdom S.Schockaert@cs.cardiff.ac.uk

[c] Computational Modeling Lab, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussel, Belgium ANowe@vub.ac.be

The final publication is available at

link.springer.com

Modeling Stable Matching Problems with Answer Set Programming*

Sofie De Clercq[a], Steven Schockaert[b], Martine De Cock[a], and Ann Nowé[c]

[a]Department of Applied Mathematics, Computer Science & Statistics, Ghent University, Krijgslaan 281, 9000 Ghent, Belgium

{Sofier.DeClercq,Martine.DeCock}@ugent.be

[b]School of Computer Science & Informatics, Cardiff University, 5 The Parade, Roath, Cardiff CF24 3AA, United Kingdom

S.Schockaert@cs.cardiff.ac.uk

[c]Computational Modeling Lab, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussel, Belgium

ANowe@vub.ac.be

Abstract

The Stable Marriage Problem (SMP) is a well-known matching problem first introduced and solved by Gale and Shapley [7]. Several variants and extensions to this problem have since been investigated to cover a wider set of applications. Each time a new variant is considered, however, a new algorithm needs to be developed and implemented. As an alternative, in this paper we propose an encoding of the SMP using Answer Set Programming (ASP). Our encoding can easily be extended and adapted to the needs of specific applications. As an illustration we show how stable matchings can be found when individuals may designate unacceptable partners and ties between preferences are allowed. Subsequently, we show how our ASP based encoding naturally allows us to select specific stable matchings which are optimal according to a given criterion. Each time, we can rely on generic and efficient off-the-shelf answer set solvers to find (optimal) stable matchings.

Keywords: Answer Set Programming, Logic Rules, Stable Marriage Problem, Optimal Stable Matchings.

1 Introduction

The Stable Marriage Problem (SMP) is a matching problem first introduced and solved by Gale and Shapley [7]. Starting from (i) a set of n men and n women, (ii) for each man a ranking of the women as preferred partners, and (iii) for each woman a ranking of the men as preferred partners, the SMP searches for a set of n couples (marriages) such that there are no man and woman who are in different marriages but both prefer each other to their actual partners. Such a man and woman are called a *blocking pair* and a matching without blocking pairs forms a *stable set* of marriages. Due to its practical relevance, countless variants on the SMP have been investigated, making the problem

---

2

assumptions more applicable to a wider range of applications, such as kidney-exchange [12] and the hospital-resident problem [18]. Recently Roth and Shapley won the Nobel Prize for Economics for their theory of stable allocations and the practice of market design, work that has directly resulted from an application of the SMP.

In the literature, typically each time a new variant on the SMP is considered, a new algorithm is developed (see e.g. [10, 13, 20]). In this paper, we propose to use Answer Set Programming (ASP) as a general vehicle for modeling a large class of extensions and variations of the SMP. We show how an ASP encoding allows us to express in a natural way ties in the preferences of men and women, as well as unacceptability constraints (where certain people prefer to remain single over being coupled to undesirable partners). Furthermore, we illustrate how we can use our ASP encoding to find stable matchings that are optimal according to a certain criterion. Although the SMP has been widely investigated, and efficient approximation or exact algorithms are available for several of its variants (see e.g. [20]), to the best of our knowledge, our encoding offers the first exact implementation to find sex-equal, minimum regret, egalitarian or maximum cardinality stable sets for SMP instances with unacceptability and ties.

The paper is structured as follows. In Section 2 we give some background about the SMP and ASP. We introduce our encoding of the SMP with ASP and prove its correctness in the third section. In Section 4, we extend our encoding enabling it to find optimal stable sets. We explore several notions of optimality for stable matchings and show how optimal stable matchings can be found by solving the corresponding disjunctive ASP program. Finally we draw our conclusions.

2 Background

2.1 The Stable Marriage Problem

To solve the standard SMP, Gale and Shapley [7] constructed an iterative algorithm —known as the Gale-Shapley algorithm, G-S algorithm or deferred-acceptance algorithm— to compute a particular solution of an SMP instance. The algorithm works as follows: in round 1 every man proposes to his first choice of all women. A woman, when being proposed, then rejects all men but her first choice among the subset of men who proposed to her. That first choice becomes her temporary husband. In the next rounds, all rejected men propose to their first choice of the subset of women by whom they were not rejected yet, regardless of whether this woman already has a temporary husband. Each woman, when being proposed, then rejects all men but her first choice among the subset of men who just proposed to her and her temporary mate. This process continues until all women have a husband. This point, when everyone has a partner, is always reached after a polynomial number of steps and the corresponding set of marriages is stable [7]. It should be noted, however, that only one of the potentially exponentially many stable matchings is found in this way. We formally define the SMP and introduce two variants that will be considered in this paper. We denote a set of men as $M = \{m_1, \ldots, m_n\}$ and a set of women $W = \{w_1, \ldots, w_p\}$, with $n = p$ for the classical SMP. A set of marriages is a set of man-woman pairs such that each man and each woman occurs in just one pair.

Definition 1 (Classical SMP)
An instance of the classical SMP is a pair (S_M, S_W), with $S_M = \{\sigma_M^1, \ldots, \sigma_M^n\}$ and $S_W = \{\sigma_W^1, \ldots, \sigma_W^n\}$ sets of permutations of the integers $1, \ldots, n$. The permutations σ_M^i and σ_W^i are the preferences of man m_i and woman w_i respectively. If $\sigma_M^i(j) = k$, we say that woman w_k is the jth most preferred woman

for man m_i, and similarly for $\sigma_W^i(j) = k$. Man m and woman w form a blocking pair in a set of marriages S if m prefers w to his partner in S and w prefers m to her partner in S. A solution of an instance is a stable set of marriages, i.e. a set of marriages without blocking pairs.

A first variant of the classical SMP allows men and women to point out unacceptable partners by not including them in their preference list. The number of men n can differ from the number of women p since men and women can remain single. A set of marriages is a set of singles (i.e. persons paired to themselves) and man-woman pairs such that every man and woman occurs in just one pair.

Definition 2 (SMP with unacceptability)
An instance of the SMP with unacceptability is a pair (S_M, S_W), $S_M = \{\sigma_M^1, \ldots, \sigma_M^n\}$, and $S_W = \{\sigma_W^1, \ldots, \sigma_W^p\}$, with each σ_M^i a permutation of a subset of $\{1, \ldots, p\}$ and each σ_W^j a permutation of a subset of $\{1, \ldots, n\}$. If $\sigma_M^i(j) = k$, woman w_k is the jth most preferred woman for man m_i, and similarly for $\sigma_W^i(j) = k$. If there is no l such that $\sigma_M^i(l) = j$, woman w_j is an unacceptable partner for man m_i, and similarly for no l such that $\sigma_W^i(l) = j$. A person x forms a blocking individual in a set of marriages S if x prefers being single to being paired with his or her partner in S. A solution of an instance is a stable set of marriages, i.e. a set of marriages without blocking pairs or individuals.

The length of the permutation σ_M^i is denoted as $|\sigma_M^i|$. A stable matching for an SMP instance with unacceptability always exists and can be found in polynomial time [22] by a slightly modified G-S algorithm.

Example 1
Suppose $M = \{m_1, m_2, m_3\}$, $W = \{w_1, w_2, w_3, w_4\}$, $S_M = \{\sigma_M^1 = (4, 1, 3), \sigma_M^2 = (3, 2), \sigma_M^3 = (1, 3)\}$ and $S_W = \{\sigma_W^1 = (1, 3), \sigma_W^2 = (2), \sigma_W^3 = (3, 2), \sigma_W^4 = (2, 1)\}$. Hence woman w_1 prefers man m_1 to man m_3 while man m_2 is unacceptable. In this setting, there is exactly one stable set of marriages [22]: $\{(m_1, w_4), (m_2, w_3), (m_3, w_1), (w_2, w_2)\}$. Thus woman w_2 stays single.

The second variant of the SMP allows unacceptability and ties, i.e. the preferences do not have to be strict. For this variant there are several ways to define stability, but we will use the notion of weak stability [11].

Definition 3 (SMP with unacceptability and ties)
An instance of the SMP with unacceptability and ties is a pair (S_M, S_W), $S_M = \{\sigma_M^1, \ldots, \sigma_M^n\}$ and $S_W = \{\sigma_W^1, \ldots, \sigma_W^p\}$. For every $i \in \{1, \ldots, n\}$, σ_M^i is a list of disjoint subsets of $\{1, \ldots, p\}$. Symmetrically σ_W^i is a list of disjoint subsets of $\{1, \ldots, n\}$ for every $i \in \{1, \ldots, p\}$. We call σ_M^i and σ_W^i the preferences of man m_i and woman w_i respectively. If $k \in \sigma_M^i(j)$, woman w_k is in man m_i's j^{th} most preferred group of women. All the women in that group are equally preferred by m_i. The case $k \in \sigma_W^i(j)$ is similar. If there is no l such that $j \in \sigma_M^i(l)$, woman w_j is an unacceptable partner for man m_i, and similar for no l such that $j \in \sigma_W^i(l)$. For every k in the set[1] $\sigma_M^i(|\sigma_M^i|)$, man m_i equally prefers staying single to being paired to woman w_k, and symmetrically for the preferences of a woman w_i. This is the only set in σ_M^i that might be empty, and similar for σ_W^i. Man m and woman w form a blocking pair in a set of marriages S if m strictly prefers w to his partner in S and w strictly prefers m to her partner in S. A blocking individual in S is a person who stricly prefers being single to being paired to his partner in S. A solution of an instance is a weakly stable set of marriages, i.e. a set of marriages without blocking pairs or individuals.

---

[1] $|\sigma_M^i|$ denotes the length of the list σ_M^i.

A weakly stable matching always exists for an instance of the SMP with unacceptability and ties and it can be found in polynomial time by arbitrarily breaking the ties [14]. However, as opposed to the previous variant, the number of matched persons is no longer constant for every stable set in this variant. Note that the setting of Definition 3 generalizes the setting of Definition 2, which generalizes the setting of Definition 1. We introduce the notations

$$acceptable_M^i = \underbrace{\sigma_M^i(1) \cup \sigma_M^i(2) \cup \ldots \cup \sigma_M^i(|\sigma_M^i| - 1)}_{= \; preferred_M^i} \cup \underbrace{\sigma_M^i(|\sigma_M^i|)}_{= \; neutral_M^i}$$

Furthermore $unacceptable_M^i = \{1, \ldots, p\} \setminus acceptable_M^i$. We define the ordering $\leq_M^{m_i}$ on $\{w_j \,|\, j \in acceptable_M^i\} \cup \{m_i\}$ as $x \leq_M^{m_i} y$ iff m_i prefers person x at least as much as person y. The strict ordering $<_M^{m_i}$ is defined in the obvious way and analogous notations are used for σ_W^j.

Example 2

Suppose $M = \{m_1, m_2\}$, $W = \{w_1, w_2, w_3, w_4\}$ and $S_M = \{\sigma_M^1 = (\{1,3\}, \{4\}), \sigma_M^2 = (\{2,3\}, \{\})\}$. Hence man m_1 prefers women w_1 and w_3 to woman w_4. There is a tie between woman w_1 and w_3 as well as between woman w_4 and staying single. Woman w_2 is unacceptable for man m_1. Man m_2 prefers woman w_2 and w_3 to staying single, but finds w_1 and w_4 unacceptable. It holds that $w_1 <_M^{m_1} m_1$, i.e. m_1 prefers marrying w_1 over staying single, $acceptable_M^1 = \{1,3,4\}$, $preferred_M^1 = \{1,3\}$, $neutral_M^1 = \{4\}$ and $unacceptable_M^1 = \{2\}$.

2.2 Answer Set Programming

Answer Set Programming or ASP is a form of declarative programming [2]. Its transparence, elegance and ability to deal with Σ_2^P-complete problems make it an attractive method for solving combinatorial search and optimization problems. An ASP program is a finite collection of first-order rules

$$A_1 \vee \ldots \vee A_k \leftarrow B_1, \ldots, B_m, not\, C_1, \ldots, not\, C_n$$

with $A_1, \ldots, A_k, B_1, \ldots, B_m, C_1, \ldots, C_n$ predicates. The semantics are defined by the *ground version* of the program, consisting of all ground instantiations of the rules w.r.t. the constants that appear in it (see e.g. [2] for a good overview). This grounded program is a propositional ASP program. The building blocks of these programs are *atoms*, *literals* and *rules*. The most elementary are *atoms*, which are propositional variables that can be true or false. A *literal* is an atom or a negated atom. Beside strong negation, ASP uses a special kind of negation, namely *negation-as-failure* (naf), denoted with '*not*'. For a literal a we call '*not a*' the naf-literal associated with a. The *extended literals* consist of all literals and their associated naf-literals. A *disjunctive rule* has the following form

$$a_1 \vee \ldots \vee a_k \leftarrow b_1, \ldots, b_m, not\, c_1, \ldots, not\, c_n$$

where $a_1, \ldots, a_k, b_1, \ldots, b_m, c_1, \ldots, c_n$ are literals from a fixed set $\mathcal{L}$, determined by a fixed set $\mathcal{A}$ of atoms. We call $a_1 \vee \ldots \vee a_k$ the head of the rule while the set of extended literals $b_1, \ldots, b_m, not\, c_1, \ldots,$ $not\, c_n$ is called the *body*. The rule above intuitively encodes that a_1, a_2, ... or a_k is true when we have evidence that $b_1, \ldots, b_m$ are true and we have no evidence that at least one of $c_1, \ldots, c_n$ are true. When a rule has an empty body, we call it a *fact*; when the head is empty, we speak of a *constraint*. A rule without occurrences of *not* is called a *simple disjunctive rule*. A *simple disjunctive* ASP program is a finite collection of simple disjunctive rules and similarly a *disjunctive* ASP program $\mathcal{P}$ is a finite collection of disjunctive rules. If each rule head consists of at most one literal, we speak of a *normal* ASP program.

We define an *interpretation I* of a disjunctive ASP program $\mathcal{P}$ as a subset of $\mathcal{L}$. An interpretation I *satisfies* a simple disjunctive rule $a_1 \vee \ldots \vee a_k \leftarrow b_1, \ldots, b_m$ when $a_1 \in I \vee \ldots \vee a_k \in I$ or $\{b_1, \ldots, b_m\} \not\subseteq I$. An interpretation which satisfies all rules of a simple disjunctive program is called a *model* of that program. An interpretation I is an *answer set* of a simple disjunctive program $\mathcal{P}$ iff it is a minimal model of $\mathcal{P}$, i.e. no strict subset of I is a model of $\mathcal{P}$ [9]. The *reduct* $\mathcal{P}^I$ of a disjunctive ASP program $\mathcal{P}$ w.r.t. an interpretation I is defined as the simple disjunctive ASP program $\mathcal{P}^I = \{a_1 \vee \ldots \vee a_k \leftarrow b_1, \ldots, b_m \mid (a_1 \vee \ldots \vee a_k \leftarrow b_1, \ldots, b_m, not\, c_1, \ldots, not\, c_n) \in \mathcal{P}, \{c_1, \ldots, c_n\} \cap I = \emptyset\}$. An interpretation I of a disjunctive ASP program $\mathcal{P}$ is an answer set of $\mathcal{P}$ iff I is an answer set of $\mathcal{P}^I$.

Example 3
Let $\mathcal{P}$ be the ASP program with the following 4 rules:
$$man(john) \leftarrow, \quad person(john) \leftarrow, \quad person(fiona) \leftarrow$$
$$woman(X) \vee child(X) \leftarrow person(X), not\, man(X)$$

The last rule is grounded to 2 rules in which X is resp. replaced by *john* and by *fiona*. We check that the interpretation $I = \{man(john), woman(fiona), person(john), person(fiona)\}$ is an answer set of the ground version of $\mathcal{P}$ by computing the reduct. The grounded rule with $X = john$ is deleted since $man(john)$ is in I. The reduct $\mathcal{P}^I$ is:
$$man(john) \leftarrow, \quad person(john) \leftarrow, \quad person(fiona) \leftarrow$$
$$woman(fiona) \vee child(fiona) \leftarrow person(fiona)$$

The first 3 rules are facts, hence their heads will be in any answer set. The fourth rule encodes that any person who is not a man, is a woman or child. It is clear that I is a minimal model of this simple program, so I is an answer set of $\mathcal{P}$. By replacing $woman(fiona)$ by $child(fiona)$ in I, another answer set is obtained.

To automatically compute the answer sets of the programs in this paper, we have used the ASP solver DLV[2], due to its ability to handle predicates, disjunction and numeric values (with some built-in aggregate functions). The numeric values are only used for grounding.

3   Modeling the Stable Marriage Problem in ASP

In this section we model variations and generalizations of the SMP with ASP. A few proposals of using nonmonotonic reasoning for modeling the SMP have already been described in the literature. For instance in [19] a specific variant of the SMP is mentioned (in which boys each know a subset of a set of girls and want to be matched to a girl they know) and in [4] an abductive program is used to find a stable set of marriages in which two fixed persons are paired, with strict, complete preference lists. To the best of our knowledge, beyond a few specific examples, no comprehensive study has been made of using ASP or related paradigms in this context. In particular, the generality of our ASP framework for weakly stable sets of SMP instances with unacceptablity and/or ties is a significant advantage. The expression $accept(m, w)$ denotes that a man m and a woman w accept each other as partners. The predicate $manpropose(m, w)$ expresses that man m is willing to propose to woman w and analogously $womanpropose(m, w)$ expresses that woman w is willing to propose to man m. Inspired by the Gale-Shapley algorithm, we look for an ASP formalisation to find the stable sets.

---
[2]Available from www.dlvsystems.com

Definition 4 (ASP program induced by SMP with unacc. and ties)
The ASP program $\mathcal{P}$ induced by an instance $(\{\sigma_M^1, \ldots, \sigma_M^n\}, \{\sigma_W^1, \ldots, \sigma_W^p\})$ of the classical SMP with unacceptability and ties is the program containing for every $i \in \{1, \ldots, n\}, j \in \{1, \ldots, p\}$ the following rules:

$$accept(m_i, w_j) \leftarrow manpropose(m_i, w_j), womanpropose(m_i, w_j) \tag{1}$$

$$accept(m_i, m_i) \leftarrow \{not\, accept(m_i, w_k) \mid k \in acceptable_M^i\} \tag{2}$$

$$accept(w_j, w_j) \leftarrow \{not\, accept(m_k, w_j) \mid k \in acceptable_W^j\} \tag{3}$$

and for every $i \in \{1, \ldots, n\}$, $j \in acceptable_M^i$:

$$manpropose(m_i, w_j) \leftarrow \{not\, accept(m_i, x) \mid x \leq_M^{m_i} w_j \ and \ w_j \neq x\} \tag{4}$$

and for every $j \in \{1, \ldots, p\}$, $i \in acceptable_W^j$:

$$womanpropose(m_i, w_j) \leftarrow \{not\, accept(x, w_j) \mid x \leq_W^{w_j} m_i \ and \ m_i \neq x\} \tag{5}$$

Intuitively (1) means that a man and woman accept each other as partners if they propose to each other. Due to (2), a man accepts himself as a partner (i.e. stays single) if no woman in his preference list is prepared to propose to him. Rule (4) states that a man proposes to a woman if he is not paired to a more or equally preferred woman. For $j \in neutral_M^i$ the body of (4) contains $not\, accept(m_i, m_i)$. No explicite rules are stated about the number of persons someone can propose to or accept but Proposition 1 implies that this is unnecessary.

We illustrate the induced ASP program with an example.

Example 4
Consider the following instance (S_M, S_W) of the SMP with unacceptability and ties. Let $M = \{m_1, m_2\}$ and $W = \{w_1, w_2, w_3\}$. Furthermore:

$$\sigma_M^1 = (\{1\}, \{2, 3\}, \{\})$$
$$\sigma_M^2 = (\{2\}, \{1\})$$
$$\sigma_W^1 = (\{1, 2\}, \{\})$$
$$\sigma_W^2 = (\{1\}, \{\})$$
$$\sigma_W^3 = (\{2\}, \{1\}, \{\})$$

The ASP program induced by this SMP instance is:

$$man(m_1) \leftarrow, \quad man(m_2) \leftarrow$$
$$woman(w_1) \leftarrow, \quad , woman(w_2) \leftarrow, \quad woman(w_3) \leftarrow$$
$$accept(X, Y) \leftarrow manpropose(X, Y), womanpropose(X, Y), man(X), woman(Y)$$
$$manpropose(m_1, w_1) \leftarrow$$
$$manpropose(m_1, w_2) \leftarrow not\, accept(m_1, w_1), not\, accept(m_1, w_3)$$
$$manpropose(m_1, w_3) \leftarrow not\, accept(m_1, w_1), not\, accept(m_1, w_2)$$
$$accept(m_1, m_1) \leftarrow not\, accept(m_1, w_1), not\, accept(m_1, w_2), not\, accept(m_1, w_3)$$
$$manpropose(m_2, w_2) \leftarrow$$
$$manpropose(m_2, w_1) \leftarrow not\, accept(m_2, w_2), not\, accept(m_2, m_2)$$
$$accept(m_2, m_2) \leftarrow not\, accept(m_2, w_2), not\, accept(m_2, w_1)$$

$$womanpropose(m_1, w_1) \leftarrow not\, accept(m_2, w_1)$$
$$womanpropose(m_2, w_1) \leftarrow not\, accept(m_1, w_1)$$
$$accept(w_1, w_1) \leftarrow not\, accept(m_1, w_1), not\, accept(m_2, w_1)$$
$$womanpropose(m_1, w_2) \leftarrow$$
$$accept(w_2, w_2) \leftarrow not\, accept(m_1, w_2)$$
$$womanpropose(m_2, w_3) \leftarrow$$
$$womanpropose(m_1, w_3) \leftarrow not\, accept(m_2, w_3)$$
$$accept(w_3, w_3) \leftarrow not\, accept(m_1, w_3), not\, accept(m_2, w_3)$$

Notice that we use the facts *man* and *woman* to capture all the rules of the form (1) at once. If we run this program in DLV, we get three answer sets containing respectively:

- $\{accept(m_1, w_3), accept(m_2, w_1), accept(w_2, w_2)\}$,

- $\{accept(m_1, w_2), accept(m_2, w_1), accept(w_3, w_3)\}$,

- $\{accept(m_1, w_1), accept(m_2, m_2), accept(w_2, w_2), accept(w_3, w_3)\}$.

These correspond to the three weakly stable set of marriages of this SMP instance, namely $\{(m_1, w_3), (m_2, w_1), (w_2, w_2)\}$, $\{(m_1, w_2), (m_2, w_1), (w_3, w_3)\}$ and $\{(m_1, w_1), (m_2, m_2), (w_2, w_2), (w_3, w_3)\}$.

Proposition 1
Let (S_M, S_W) be an instance of the SMP with unacceptability and ties and let $\mathcal{P}$ be the corresponding ASP program. If I is an answer set of $\mathcal{P}$, then a weakly stable matching for (S_M, S_W) is given by $\{(x, y) \mid accept(x, y) \in I\}$.

Proof. Let (S_M, S_W) and $\mathcal{P}$ be as described in the proposition. Because of the symmetry between the men and the women we restrict ourselves to the male case when possible. We prove this proposition in 4 steps.

1. *For every $i \in \{1, \ldots, n\}$, every $j \in \{1, \ldots, p\}$ and for every answer set I of $\mathcal{P}$, it holds that $accept(m_i, w_j) \in I$ implies that $j \in acceptable_M^i$ and $i \in acceptable_W^j$.*
   This can be proved by contradiction. We will prove that for every man m_i and every $j \in unacceptable_M^i$, $accept(m_i, w_j)$ is in no answer set I of the induced ASP program $\mathcal{P}$. For $accept(m_i, w_j)$ to be in an answer set I, the reduct must contain some rule with this literal in the head and a true body. The only rule which can make this happen is the one of the form (1), implying that $manpropose(m_i, w_j)$ should be in I. But since j is not in $acceptable_M^i$ there is no rule with $manpropose(m_i, w_j)$ in the head and so $manpropose(m_i, w_j)$ can never be in I.

2. *For every answer set I of $\mathcal{P}$ and every man m_i, there exists at most one woman w_j such that $accept(m_i, w_j) \in I$. Similarly, for every woman w_j there exists at most one man m_i such that $accept(m_i, w_j) \in I$. Moreover, if $accept(m_i, m_i) \in I$ then $accept(m_i, w_j) \notin I$ for any w_j, and likewise when $accept(w_j, w_j) \in I$ then $accept(m_i, w_j) \notin I$ for any m_i.*
   This can be proved by contradiction. Suppose first that there is an answer set I of $\mathcal{P}$ that contains $accept(m_i, w_j)$ and $accept(m_i, w_{j'})$ for some man m_i and two different women w_j and $w_{j'}$. The first step implies that j and j' are elements of $acceptable_M^i$. Either man m_i prefers woman w_j to woman $w_{j'}$ ($w_j \leq_M^{m_i} w_{j'}$) or the other way around ($w_{j'} \leq_M^{m_i} w_j$) or man m_i has

no preference among them ($w_j \leq^{m_i}_M w_{j'}$ and $w_{j'} \leq^{m_i}_M w_j$). The first two cases are symmetrical and can be handled analogously. The last case follows from the first case because it has stronger assumptions. We prove the first case and assume that man m_i prefers woman w_j to woman $w_{j'}$. The rules (4) imply the presence of a rule $manpropose(m_i, w_{j'}) \leftarrow \ldots,\, not\, accept(m_i, w_j), \ldots$ and this is the only rule which can make $manpropose(m_i, w_{j'})$ true (the only rule with this literal in the head). However, since $accept(m_i, w_j)$ is also in the answer set, this rule has a false body so $manpropose(m_i, w_{j'})$ can never be in I. Consequently $accept(m_i, w_{j'})$ can never be in I since the only rule with this literal in the head is of the form (1) and this body can never be true, which leads to a contradiction.

Secondly assume that $accept(m_i, w_j)$ and $accept(m_i, m_i)$ are both in an answer set I of $\mathcal{P}$. Again step 1 implies that $j \in acceptable^i_M$. Because of the rules (2) $\mathcal{P}$ will contain the rule $accept(m_i, m_i) \leftarrow \ldots,\, not\, accept(m_i, w_j), \ldots$. An analogous reasoning as above implies that since $accept(m_i, w_j)$ is in the answer set I, $accept(m_i, m_i)$ can never be in I.

3. *For every man m_i, in every answer set I of $\mathcal{P}$ exactly one of the following conditions is satisfied*:

   (a) *there exists a woman w_j such that $accept(m_i, w_j) \in I$,*

   (b) *$accept(m_i, m_i) \in I$,*

   and similarly for every woman w_i.
   Suppose I is an arbitrary answer set of $\mathcal{P}$ and m_i is an arbitrary man. We already know from step 2 that a man cannot be paired to a woman while being single, so both possibilities are disjoint. So suppose there is no woman w_j such that $accept(m_i, w_j)$ is in I. $\mathcal{P}$ will contain the rule (2). Because of our assumptions and the definition of the reduct, this rule will be reduced to $accept(m_i, m_i) \leftarrow$, and so $accept(m_i, m_i)$ will be in I.

4. For an arbitrary answer set I of $\mathcal{P}$ the previous steps imply that I produces a set of marriages without blocking individuals. Weak stability also demands the absence of blocking pairs. Suppose by contradiction that there is a blocking pair (m_i, w_j), implying that there exist $i \neq i'$ and $j \neq j'$ such that $accept(m_i, w_{j'}) \in I$ and $accept(m_{i'}, w_j) \in I$ while $w_j <^{m_i}_M w_{j'}$ and $m_i <^{w_j}_W m_{i'}$. The rules of the form (1), the only ones with the literals $accept(m_i, w_{j'})$ and $accept(m_{i'}, w_j)$ in the head, imply that literals $manpropose(m_i, w_{j'})$ and $womanpropose(m_{i'}, w_j)$ should be in I. But since $w_j <^{m_i}_M w_{j'}$ and because of the form of the rules (4) there are fewer conditions to be fulfilled for $manpropose(m_i, w_j)$ to be in I than for $manpropose(m_i, w_{j'})$ to be in I. So $manpropose(m_i, w_j)$ should be in I as well. A similar reasoning implies that $womanpropose(m_i, w_j)$ should be in I. But now the rules of the form (1) imply that $accept(m_i, w_j)$ should be in I, contradicting step 2 since $accept(m_i, w_{j'})$ and $accept(m_{i'}, w_j)$ are already in I.

$\square$

Proposition 2
Let (S_M, S_W) be an instance of the SMP with unacceptability and ties, and let $\mathcal{P}$ be the corresponding ASP program. If $\{(x_1, y_1), \ldots, (x_k, y_k)\}$ is a weakly stable matching for (S_M, S_W) then $\mathcal{P}$ has the following answer set I:

$$\{manpropose(x_i, y) \,|\, i \in \{1, \ldots, k\}, x_i \in M, y <^{x_i}_M y_i)\}$$
$$\cup \{womanpropose(x, y_i) \,|\, i \in \{1, \ldots, k\}, y_i \in W, x <^{y_i}_W x_i\}$$

$$\cup \{accept(x_i, y_i) \mid i \in \{1, \ldots, k\}\}$$
$$\cup \{manpropose(x_i, y_i) \mid i \in \{1, \ldots, k\}, x_i \neq y_i\}$$
$$\cup \{womanpropose(x_i, y_i) \mid i \in \{1, \ldots, k\}, x_i \neq y_i\}$$

Proof. Suppose we have a stable set of marriages $S = \{(x_1, y_1), \ldots, (x_k, y_k)\}$, implying that every y_i is an acceptable partner of x_i and the other way around. The rules of the form (1) do not alter when forming the reduct, but the other rules do as those contain naf-literals. Notice first that the stability of S implies that there cannot be an unmarried couple (m, w), with m a man and w a woman, such that $manpropose(m, w)$ is in I and $womanpropose(m, w)$ is in I. By definition of I this would mean that they both strictly prefer each other to their current partner in S. This means they would form a blocking pair, but since S was stable, that is impossible. So the rules of the form (1) will be applied exactly for married couples (m_i, w_j), since by definition of I $manpropose(m_i, w_j)$ and $womanpropose(m_i, w_j)$ are both in I under these conditions. For other cases the rule will also be fulfilled since the body will be false. This reasoning implies that the unique minimal model of the reduct w.r.t. I should indeed contain $accept(m_i, w_j)$ for every married couple (m_i, w_j) in S. Since S is a stable set of marriages, every person is either married or single. If a man m_i is single, there will be no other literal of the form $accept(m_i, .)$ in I, so rule (2) will reduce to a fact $accept(m_i, m_i) \leftarrow$, which is obviously fulfilled by I. Similarly if a woman w_j is single. Any other rule of the form (2) or (3) is deleted because m_i or w_j is not single in that case and thus there is some literal of the form $accept(m_i, w)$ for some woman w and some literal of the form $accept(m, w_j)$ for some man m in I, falsifying the body of the rules. If m_i is single, then $accept(m_i, m_i)$ is in I and this is the only literal of the form $accept(m_i, .)$ in I, so the rules of the form (4) will all be reduced to facts. The rule heads of these facts should be in the minimal model of the reduct and are indeed in I since the women w for which $manpropose(m_i, w)$ is in I are exactly those who are strictly preferred to staying single. The rules of the form (4) for women w_j in $neutral_M^i$ will all be deleted in this case, because $accept(m_i, m_i)$ is in I. If man m_i is married to a certain woman w_j in the stable set S then the rules of the form (4) will reduce to facts of the form $manpropose(m_i, w) \leftarrow$ for every woman w who is strictly preferred to w_j and will be deleted for every other woman appearing in the head, because those rules will contain $not\ accept(m_i, w_j)$ in the body. Again I contains these facts by definition, as the minimal model of the reduct should. We can use an analogous reasoning for the women. So the presence of the literals of the form $manpropose(., .)$, $womanpropose(., .)$ and $accept(., .)$ in I is required in the unique minimal model of the reduct w.r.t. I. We have proved that every literal in I should be the minimal model of the reduct and that every rule of the reduct is fulfilled by I, implying that I is an answer set of $\mathcal{P}$. $\square$

In [18] it is shown that the decision problem 'is the pair (m, w) stable?' for a given SMP instance with unacceptablity and ties is an NP-complete problem, even in the absence of unacceptability. A pair (m, w) is *stable* if there exists a stable set that contains (m, w). It is straightforward to see that we can reformulate this decision problem as 'does there exist an answer set of the induced normal ASP program $\mathcal{P}$ which contains the literal $accept(m, w)$?' (i.e. brave reasoning), which is known to be an NP-complete problem [1]. So our model forms a suitable framework for these kind of decision problems concerning the SMP.

4 Selecting Preferred Stable Sets

4.1 Notions of Optimality of Stable Sets

When several stable matchings can be found for an instance of the SMP, some may be more interesting than others. The stable set found by the G-S algorithm is *M-optimal* [22], i.e. every man likes this set at least as well as any other stable set. Exchanging the roles of men and women in the G-S algorithm yields a *W-optimal* stable set [7], optimal from the point of view of the women.

While some applications may require us to favour either the men or the women, in others it makes more sense to treat both parties equally. To formalize some commonly considered notions of fairness and optimality w.r.t. the SMP, we define the cost $c_x(S)$ of a stable set S to an individual x, where $c_x(S) = k$ if x has been matched with his or her k^{th} preferred partner. More precisely, for $x = m_i$ a man, we define $c_{m_i}(S) = |\{z : z <_M^{m_i} y\}| + 1$ where y is the partner of x in S; for $x = w_j$ a woman, c_x is defined analogously. So in case of ties we assign the same list position to equally preferred partners, as illustrated in Example 5.

Example 5

Let $x = m_1$ be a man with preference list $\sigma_M^1 = (\{1\}, \{2,3\}, \{4\})$ then w_1 as partner of x in some set of marriages S would yield $c_x(S) = 1$, w_2 and w_3 yield $c_x(S) = 2$ and w_4 yields $c_x(S) = 4$. If m_1 would be single in S, then the cost $c_x(S)$ is 4, since m_1 prefers women w_1, w_2 and w_3 to being single, but is indifferent between being paired to w_4 or staying single.

Definition 5

For S a set of marriages,

- *the sex-equalness cost is defined as $c_{sexeq}(S) = |\sum_{x \in M} c_x(S) - \sum_{x \in W} c_x(S)|$,*

- *the egalitarian cost is defined as $c_{weight}(S) = \sum_{x \in M \cup W} c_x(S)$,*

- *the regret cost is defined as $c_{regret}(S) = \max_{x \in M \cup W} c_x(S)$, and*

- *the cardinality cost is defined as $c_{singles}(S) = |\{z : (z,z) \in S\}|$.*

S is a sex-equal stable set iff S is a stable set with minimal sex-equalness cost. Similarly, S is an egalitarian (resp. minmum regret, maximum cardinality) stable set iff S is a stable set with minimal egalitarian (resp. regret or cardinality) cost.

A *sex-equal stable set* assigns an equal importance to the preferences of the men and women. An *egalitarian stable set* is a stable set in which the preferences of every individual are considered to be equally important. In [23] the use of an egalitarian stable set is proposed to optimally match virtual machines (VM) to servers in order to improve cloud computing by equalizing the importance of migration overhead in the data center network and VM migration performance. A *minimum regret stable set* is optimal for the person who is worst off. A *maximal or minimal cardinality stable set* is a stable set with resp. as few or as many singles as possible. Examples of practical applications include an efficient kidney exchange program [21] and the National Resident Matching Program[3] [18]. Maximizing cardinality garantuees that as many donors as possible will get a compatible donor and as many medical graduates as possible will get a position.

---

[3]www.nrmp.org

Table 1 presents an overview of known complexity results[4] concerning finding an optimal stable set. Typically the presence of ties leads to an increase of complexity. Manlove et al. [17, 18] proved that the problem of finding a maximum (or minimum) cardinality stable set for a given instance of the SMP with unacceptability and ties is NP-hard. Using this result, the problem of finding an egalitarian or minimum regret stable matching for a given SMP instance with ties is proved to be NP-hard [18], even if the ties occur on one side only and each tie is of length 2 (i.e. each set in a preference list has size at most 2). If there are no ties, the problem of finding an egalitarian or minimum regret stable set is solvable in polynomial time [13, 10]. Since all stable sets consist of n couples in the classical SMP, the G-S algorithm trivially finds a maximum (or minimum) cardinality [7]. For an SMP instance with unacceptability the number of couples in a stable set is constant [8], so finding a maximum cardinality stable set reduces to finding a stable set, which is known to be solvable in polynomial time. Surprisingly, finding a sex-equal stable set for a classical SMP instance is NP-hard [16], even if the preference lists are bound in length by 3 [20].

Table 1: Literature complexity results for finding an optimal stable set

	sex-equal	egalitarian	min. regret	max. card.
SMP	NP-hard [16]	P ($O(n^4)$ [13])	P ($O(n^2)$ [10])	P ($O(n^2)$ [7])
SMP + unacc	NP-hard [20]			P [8]
SMP + ties		NP-hard [18]	NP-hard [18]	
SMP + {unacc,ties}				NP-hard [17, 18]

Between brackets we mention in Table 1 the complexity of an algorithm that finds an optimal stable set if one exists, in function of the number of men n. To the best of our knowledge, the only exact algorithm tackling an NP-hard problem from Table 1 finds a sex-equal stable set for an SMP instance in which the strict preference lists of men and/or women are bounded in length by a constant [20]. To the best of our knowledge, no exact implementations exist to find an optimal stable set for an SMP instance with ties, regardless of the presence of unacceptability and regardless which notion of optimality from Table 1 is used. Our approach yields an exact implementation of all problems mentioned in Table 1.

4.2 Finding Optimal Stable Sets using Disjunctive ASP

As we discuss next, we can extend our ASP encoding of the SMP such that the optimal stable sets correspond to the answer sets of an associated ASP program. In particular, we use a saturation technique [5, 1] to filter non-optimal answer sets. Intuitively, the idea is to create a program with 3 components: (i) a first part describing the solution candidates, (ii) a second part also describing the solution candidates since comparison of solutions requires multiple solution candidates within the same answer set whereas the first part in itself produces one solution per answer set, (iii) a third part comparing the solutions described in the first two parts and selecting the preferred solutions by saturation. It is known that the presence of negation-as-failure can cause problems when applying saturation. Therefore, we use a SAT encoding [15] of the ASP program in Definition 4 and define a

---

[4]Throughout this paper we assume that P $\neq$ NP.

disjunctive naf-free ASP program in Definition 6 which selects particular models of the SAT problem. Notice that our original normal program is absolutely tight, i.e. there is no finite sequence $l_1, l_2, \ldots$ of literals such that for every i there is a program rule for which l_{i+1} is a positive body literal and l_i is in the head [6]. We use the completion and a translation of our ASP program to SAT to derive Definition 6. The completion of a normal ASP program is a set of propositional formulas. For every atom a with $a \leftarrow body_i$ $(i \in \{1, \ldots, k\})$ all the program rules with head a, the propositional formula $a \equiv body'_1 \vee \ldots \vee body'_k$ is in the completion of that program. If an atom a of the program does not occur in any rule head, than $a \equiv \bot$ is in the completion of the program. Similarly the completion of the program contains the propositional formula $\bot \equiv body'_1 \vee \ldots \vee body'_l$ with $\leftarrow body_i$ $(i \in \{1, \ldots, l\})$ all the program constraints. For every i, $body'_i$ is the conjunction of literals derived from $body_i$ by replacing every occurence of 'not' with '$\neg$'. Because our program is absolutely tight, we know that the completion will correspond to it [6]. Applied to the induced normal ASP program in Definition 4, the completion becomes:

$$\{accept(m_i, w_j) \equiv manpropose(m_i, w_j) \wedge womanpropose(m_i, w_j) \,|\, i \in \{1, \ldots, n\}, j \in \{1, \ldots, p\}\}$$

$$\cup \{accept(m_i, m_i) \equiv \bigwedge_{k \in acceptable^i_M} \neg accept(m_i, w_k) \,|\, i \in \{1, \ldots, n\}\}$$

$$\cup \{accept(w_j, w_j) \equiv \bigwedge_{k \in acceptable^j_W} \neg accept(m_k, w_j) \,|\, j \in \{1, \ldots, p\}\}$$

$$\cup \{manpropose(m_i, w_j) \equiv \bigwedge_{x \leq^{m_i}_M w_j, x \neq w_j} \neg accept(m_i, x) \,|\, i \in \{1, \ldots, n\}, j \in acceptable^i_M\}$$

$$\cup \{womanpropose(m_i, w_j) \equiv \bigwedge_{x \leq^{w_j}_W m_i, x \neq m_i} \neg accept(x, w_j) \,|\, j \in \{1, \ldots, p\}, i \in acceptable^j_W\}$$

$$\cup \{manpropose(m_i, w_j) \equiv \bot \,|\, i \in \{1, \ldots, n\}, j \in unacceptable^i_M\}$$

$$\cup \{womanpropose(m_i, w_j) \equiv \bot \,|\, j \in \{1, \ldots, p\}, i \in unacceptable^j_W\}$$

Using the formulas of the completion corresponding to the normal ASP program in Definition 4, we can define a corresponding disjunctive ASP program without negation-as-failure. Lemma 3 follows form the fact that the completion corresponds to the original program [6].

Definition 6 (Induced disj. naf-free ASP program)
The disjunctive naf-free ASP program $\mathcal{P}_{disj}$ induced by an SMP instance (S_M, S_W) with unacceptability and ties contains the following rules for $i \in \{1, \ldots, n\}, j \in \{1, \ldots, p\}$:

$$\neg accept(m_i, w_j) \vee manpropose(m_i, w_j) \leftarrow$$
$$\neg accept(m_i, w_j) \vee womanpropose(m_i, w_j) \leftarrow$$
$$accept(m_i, w_j) \vee \neg manpropose(m_i, w_j) \vee \neg womanpropose(m_i, w_j) \leftarrow$$

For every $i \in \{1, \ldots, n\}, l \in unacceptable^i_M, j \in acceptable^i_M, x \leq^{m_i}_M w_j, x \neq w_j$ $\mathcal{P}_{disj}$ contains:

$$\bigvee_{k \in acceptable^i_M} accept(m_i, w_k) \vee accept(m_i, m_i) \leftarrow$$
$$\neg accept(m_i, m_i) \vee \neg accept(m_i, w_j) \leftarrow$$
$$\neg manpropose(m_i, w_j) \vee \neg accept(m_i, x) \leftarrow$$

$$\bigvee_{x \leq_M^{m_i} w_j, x \neq w_j} accept(m_i, x) \lor manpropose(m_i, w_j) \leftarrow$$

$$\neg manpropose(m_i, w_l) \leftarrow$$

and symmetrical for $j \in \{1, \ldots, p\}$ and womanpropose.

Lemma 3
Let $\mathcal{P}$ be the normal ASP program from Definition 4 and $\mathcal{P}_{disj}$ the disjunctive ASP program from Definition 6. It holds that for any answer set I of $\mathcal{P}$ there exists an answer set I_{disj} of $\mathcal{P}_{disj}$ such that the atoms of I and I_{disj} coincide. Conversely for any answer set I_{disj} of $\mathcal{P}_{disj}$ there exists an answer set I of $\mathcal{P}$ such that the atoms of I and I_{disj} coincide.

4.3 ASP Program to Select Optimal Solutions

Let (S_M, S_W) be an SMP instance with unacceptability and ties, with $S_M = \{\sigma_M^1, \ldots, \sigma_M^n\}$ and $S_W = \{\sigma_W^1, \ldots, \sigma_W^p\}$, and let $\mathcal{P}_{norm}$ be the induced normal ASP program from Definition 4. Our technique for extending this program to a program that can respectively optimize for the sex-equalness, egalitarian, minimum regret and maximum cardinality criterion is in each case very similar. We start by explaining it for the case of sex-equalness. Our first step is to add a set of rules that compute the sex-equalness cost of a set of marriages. For every man m_i and every woman w_j such that $j \in \sigma_M^i(k)$ we use the following rule to determine the cost for m_i if w_j would be his partner:

$$mancost(i, k) \leftarrow accept(m_i, w_j) \tag{6}$$

and similarly for every w_j and every m_i such that $i \in \sigma_W^j(k)$:

$$womancost(j, k) \leftarrow accept(m_i, w_j) \tag{7}$$

We also use the following rules with i ranging from 1 to n and j from 1 to p:

$$mancost(i, |\sigma_M^i|) \leftarrow accept(m_i, m_i) \tag{8}$$

$$womancost(j, |\sigma_W^j|) \leftarrow accept(w_j, w_j) \tag{9}$$

$$manweight(Z) \leftarrow \#sum\{B, A : mancost(A, B)\} = Z, \#int(Z) \tag{10}$$

$$womanweight(Z) \leftarrow \#sum\{B, A : womancost(A, B)\} = Z, \#int(Z) \tag{11}$$

$$sexeq(Z) \leftarrow manweight(X), womanweight(Y), Z = X - Y$$

$$sexeq(Z) \leftarrow manweight(X), womanweight(Y), Z = Y - X \tag{12}$$

Rules (8) and (9) state staying single leads to the highest cost. Rule (10) determines the sum of the male costs[5] and similarly (11) determines the sum of the female costs. According to Definition 5 the absolute difference of these values yields the sex-equalness cost, as determined by rules (12). Since numeric variables are restricted to positive integers in DLV, we omit conditions as '$X \geq Y$' or '$X < Y$'. The program $\mathcal{P}_{norm}$ extended with rules (6) – (12) is denoted $\mathcal{P}_{ext}^{sexeq}$. We construct a program $\mathcal{P}_{sexeq}$, composed by subprograms, that selects optimal solutions. Let $\mathcal{P}_{disj}'$ be the disjunctive naf-free ASP program, induced by the same SMP instance, in which a prime symbol is added to all literal names (e.g. $accept$ becomes $accept'$). Define a new program $\mathcal{P}_{ext}'^{sexeq}$ with all the rules of $\mathcal{P}_{disj}'$ in

---

[5]$\#sum$, $\#max$, $\#int$ and $\#count$ are DLV aggregate functions. The 'A' mentioned as variable in $\#sum$ indicates that a cost must be included for every person (otherwise the cost is included only once when persons have the same cost).

which every occurrence of $\neg atom$ is changed into $natom$ for every atom $atom$, i.e. replace all negation symbols by a prefix 'n'. For every occurring atom $atom$ in $\mathcal{P}'^{sexeq}_{ext}$, add the following rule to exclude non-consistent solutions[6]:

$$sat \leftarrow atom, natom \tag{13}$$

Finally add rules (6) – (12) with prime symbols to the literal names to $\mathcal{P}'^{sexeq}_{ext}$ but replace rule (10) and rule (11) by:

$$mansum(n, X) \leftarrow mancost(n, X)$$
$$mansum(J, Z) \leftarrow mansum(I, X), mancost(J, Y), Z = X + Y, \#succ(J, I)$$
$$manweight(Z) \leftarrow mansum(1, Z)$$
$$womansum(p, X) \leftarrow womancost(p, X)$$
$$womansum(J, Z) \leftarrow womansum(I, X), womancost(J, Y), Z = X + Y, \#succ(J, I)$$
$$womanweight(Z) \leftarrow womansum(1, Z) \tag{14}$$

The DLV aggregate function $\#succ(J, I)$ is true whenever $J + 1 = I$. The reason we replace the rules with the aggregate function $\#max$ by these rules is to make sure the saturation happens correct. When saturation is used, the DLV aggregate function $\#max$, $\#sum$ and $\#count$ would not yield the right criteriumvalues. Moreover, DLV does not accept these aggregate function in saturation because of the cyclic dependency of literals within the aggregate functions created by the rules for saturation. These adjusted rules, however, will do the job because of the successive way they compute the criteriumvalues. This becomes more clear in the proof of Proposition 4. We define the ASP program $\mathcal{P}_{sexeq}$ as the union of $\mathcal{P}^{sexeq}_{ext}$, $\mathcal{P}'^{sexeq}_{ext}$ and $\mathcal{P}_{sat}$. The ASP program $\mathcal{P}_{sat}$ contains the following rules to select minimal solutions based on sex-equalness:

$$sat \leftarrow sexeq(X), sexeq'(Y), X \leq Y \tag{15}$$
$$\leftarrow not\ sat \tag{16}$$
$$mancost'(X, Y) \leftarrow sat, manargcost'_1(X), manargcost'_2(Y)$$
$$womancost'(X, Y) \leftarrow sat, womanargcost'_1(X), womanargcost'_2(Y) \tag{17}$$
$$manpropose'(X, Y) \leftarrow sat, man(X), woman(Y)$$
$$womanpropose'(X, Y) \leftarrow sat, man(X), woman(Y)$$
$$accept'(X, X) \leftarrow sat, man(X)$$
$$accept'(X, X) \leftarrow sat, woman(X)$$
$$accept'(X, Y) \leftarrow sat, man(X), woman(Y) \tag{18}$$

and analogous to (18) a set of rules with prefix 'n' for the head predicates. Finally we add the facts[7] $manargcost'_1(1..n) \leftarrow$, $manargcost'_2(1..(p+1)) \leftarrow$, $womanargcost'_1(1..p) \leftarrow$, $womanargcost'_2(1..(n+1)) \leftarrow$, $man(x) \leftarrow$ for every man x and $woman(x) \leftarrow$ for every woman x to $\mathcal{P}_{sat}$. Intuitively the rules of $\mathcal{P}_{sat}$ express the key idea of saturation. First every answer set is forced to contain the atom sat by rule (16). Then the rules (17) – (18) and the facts make sure that any answer set should contain all possible literals with a prime symbol that occur in $\mathcal{P}_{sexeq}$. Rule (15) will establish that only optimal solutions will correspond to minimal models and thus lead to answer sets. For any

---

[6]For instance, $sat \leftarrow accept'(m_1, w_1), naccept'(m_1, w_1)$

[7]The rule $manargcost'_1(1..n) \leftarrow$ is DLV-syntax for the n facts $manargcost'_1(1) \leftarrow, \ldots, manargcost'_1(n) \leftarrow$.

non-optimal solution, the corresponding interpretation containing *sat* will never be a minimal model of the reduct. It is formally proved in Proposition 4 below that $\mathcal{P}_{sexeq}$ produces exactly the stable matchings with minimal sex-equalness cost.

Furthermore, only small adjustments to $\mathcal{P}_{sexeq}$ are needed to create programs $\mathcal{P}_{weight}$, $\mathcal{P}_{regret}$, and $\mathcal{P}_{singles}$ that resp. produce egalitarian, minimum regret and maximum cardinality stable sets. Indeed, the ASP program $\mathcal{P}_{weight}$ can easily be defined as $\mathcal{P}_{sexeq}$ in which the predicates *sexeq* and *sexeq*′ are resp. replaced by *weight* and *weight*′ and the rules (12) are replaced by (19), determining the egalitarian cost of Definition 5 as the sum of the male and female costs:

$$weight(Z) \leftarrow manweight(X), womanweight(Y), Z = X + Y \tag{19}$$

Similarly the ASP program $\mathcal{P}_{regret}$ is defined as $\mathcal{P}_{sexeq}$ in which the predicates *sexeq* and *sexeq*′ are resp. replaced by *regret* and *regret*′ and rules (10) – (12) are replaced by the following rules:

$$manregret(Z) \leftarrow \#max\{B : mancost(A, B)\} = Z, \#int(Z) \tag{20}$$
$$womanregret(Z) \leftarrow \#max\{B : womancost(A, B)\} = Z, \#int(Z) \tag{21}$$
$$regret(X) \leftarrow manregret(X), womanregret(Y), X > Y$$
$$regret(Y) \leftarrow manregret(X), womanregret(Y), X \leq Y \tag{22}$$

Rule (20) determines the regret cost but only for the men. Similarly (21) determines the regret cost for the women. The regret cost as defined in Definition 5 is the maximum of these two values, determined by the rules in (22). Again we adjust rules (20) and (21) for the program part $\mathcal{P}_{ext}'^{regret}$ by replacing them with a successively computing variant:

$$manmax(n, X) \leftarrow mancost(n, X)$$
$$manmax(J, X) \leftarrow manmax(I, X), mancost(J, Y), X \geq Y, \#succ(J, I)$$
$$manmax(J, Y) \leftarrow manmax(I, X), mancost(J, Y), X < Y, \#succ(J, I)$$
$$manregret(Z) \leftarrow manmax(1, Z)$$
$$womanmax(p, X) \leftarrow womancost(p, X)$$
$$womanmax(J, X) \leftarrow womanmax(I, X), womancost(J, Y), X \geq Y, \#succ(J, I)$$
$$womanmax(J, Y) \leftarrow womanmax(I, X), womancost(J, Y), X < Y, \#succ(J, I)$$
$$womanregret(Z) \leftarrow womanmax(1, Z) \tag{23}$$

Finally we define the ASP program $\mathcal{P}_{singles}$ as $\mathcal{P}_{sexeq}$ in which the predicates *sexeq* and *sexeq*′ are resp. replaced by *singles* and *singles*′. Furthermore we replace rules (6) – (12) by (24), determining the number of singles:

$$singles(Z) \leftarrow \#count\{B : accept(B, B)\} = Z, \#int(Z) \tag{24}$$

This time we adjust rule (24) for the program part $\mathcal{P}_{ext}'^{singles}$ as follows:

$$single(p + i, 1) \leftarrow accept(m_i, m_i)$$
$$single(p + i, 0) \leftarrow naccept(m_i, m_i)$$
$$single(j, 1) \leftarrow accept(w_j, w_j)$$
$$single(j, 0) \leftarrow naccept(w_j, w_j)$$
$$singlesum(n + p, X) \leftarrow single(n + p, X)$$
$$singlesum(J, Z) \leftarrow singlesum(I, X), single(J, Y), Z = X + Y, \#succ(J, I)$$

$$singles(Z) \leftarrow sat, singlesum(1, Z) \qquad (25)$$

Example 6
We reconsider Example 4. This SMP instance had 3 stable sets of marriages:

- $S_1 = \{accept(m_1, w_3), accept(m_2, w_1), accept(w_2, w_2)\}$,

- $S_2 = \{accept(m_1, w_2), accept(m_2, w_1), accept(w_3, w_3)\}$,

- $S_3 = \{accept(m_1, w_1), accept(m_2, m_2), accept(w_2, w_2), accept(w_3, w_3)\}$.

It is easy to compute the respective regret costs as $c_{regret}(S_1) = 2$ and $c_{regret}(S_2) = c_{regret}(S_3) = 3$. The corresponding program selecting this minimum regret stable set is the program consisting of the rules in Example 4 in addition with:

$$mancost(1, 1) \leftarrow accept(m_1, w_1)$$
$$mancost(1, 2) \leftarrow accept(m_1, w_2)$$
$$mancost(1, 2) \leftarrow accept(m_1, w_3)$$
$$mancost(1, 3) \leftarrow accept(m_1, m_1)$$
$$mancost(2, 2) \leftarrow accept(m_2, w_1)$$
$$mancost(2, 1) \leftarrow accept(m_2, w_2)$$
$$mancost(2, 2) \leftarrow accept(m_2, m_2)$$
$$womancost(1, 1) \leftarrow accept(m_1, w_1)$$
$$womancost(1, 1) \leftarrow accept(m_2, w_1)$$
$$womancost(1, 2) \leftarrow accept(w_1, w_1)$$
$$womancost(2, 1) \leftarrow accept(m_1, w_2)$$
$$womancost(2, 2) \leftarrow accept(w_2, w_2)$$
$$womancost(3, 2) \leftarrow accept(m_1, w_3)$$
$$womancost(3, 1) \leftarrow accept(m_2, w_3)$$
$$womancost(3, 3) \leftarrow accept(w_3, w_3)$$
$$manregret(Z) \leftarrow \#max\{B : mancost(A, B)\} = Z, \#int(Z)$$
$$womanregret(Z) \leftarrow \#max\{B : womancost(A, B)\} = Z, \#int(Z)$$
$$regret(X) \leftarrow manregret(X), womanregret(Y), X > Y$$
$$regret(Y) \leftarrow manregret(X), womanregret(Y), X <= Y$$

$$naccept'(M, W) \vee manpropose'(M, W) \leftarrow man(M), woman(W)$$
$$naccept'(M, W) \vee womanpropose'(M, W) \leftarrow man(M), woman(W)$$
$$accept'(M, W) \vee nmanpropose'(M, W) \vee nwomanpropose'(M, W) \leftarrow man(M), woman(W)$$
$$accept'(m_1, w_1) \vee accept'(m_1, w_2) \vee accept'(m_1, w_3) \vee accept'(m_1, m_1) \leftarrow$$
$$accept'(m_2, w_1) \vee accept'(m_2, w_2) \vee accept'(m_2, m_2) \leftarrow$$
$$naccept'(m_1, m_1) \vee naccept'(m_1, w_1) \leftarrow$$
$$naccept'(m_1, m_1) \vee naccept'(m_1, w_2) \leftarrow$$

$$naccept'(m_1, m_1) \lor naccept'(m_1, w_3) \leftarrow$$
$$naccept'(m_2, m_2) \lor naccept'(m_2, w_1) \leftarrow$$
$$naccept'(m_2, m_2) \lor naccept'(m_2, w_2) \leftarrow$$
$$accept'(m_1, w_1) \lor accept'(m_2, w_1) \lor accept'(w_1, w_1) \leftarrow$$
$$accept'(m_1, w_2) \lor accept'(w_2, w_2) \leftarrow$$
$$accept'(m_1, w_3) \lor accept'(m_2, w_3) \lor accept'(w_3, w_3) \leftarrow$$
$$naccept'(w_1, w_1) \lor naccept'(m_1, w_1) \leftarrow$$
$$naccept'(w_1, w_1) \lor naccept'(m_2, w_1) \leftarrow$$
$$naccept'(w_2, w_2) \lor naccept'(m_1, w_2) \leftarrow$$
$$naccept'(w_3, w_3) \lor naccept'(m_1, w_3) \leftarrow$$
$$naccept'(w_3, w_3) \lor naccept'(m_2, w_3) \leftarrow$$
$$nmanpropose'(m_1, w_2) \lor naccept'(m_1, w_1) \leftarrow$$
$$nmanpropose'(m_1, w_2) \lor naccept'(m_1, w_3) \leftarrow$$
$$nmanpropose'(m_1, w_3) \lor naccept'(m_1, w_1) \leftarrow$$
$$nmanpropose'(m_1, w_3) \lor naccept'(m_1, w_2) \leftarrow$$
$$manpropose'(m_1, w_1) \leftarrow$$
$$accept'(m_1, w_1) \lor accept'(m_1, w_3) \lor manpropose'(m_1, w_2) \leftarrow$$
$$accept'(m_1, w_1) \lor accept'(m_1, w_2) \lor manpropose'(m_1, w_3) \leftarrow$$
$$nmanpropose'(m_2, w_1) \lor naccept'(m_2, w_2) \leftarrow$$
$$nmanpropose'(m_2, w_1) \lor naccept'(m_2, m_2) \leftarrow$$
$$manpropose'(m_2, w_2) \leftarrow$$
$$accept'(m_2, w_2) \lor accept'(m_2, m_2) \lor manpropose'(m_2, w_1) \leftarrow$$
$$nwomanpropose'(m_1, w_1) \lor naccept'(m_2, w_1) \leftarrow$$
$$nwomanpropose'(m_2, w_1) \lor naccept'(m_1, w_1) \leftarrow$$
$$accept'(m_1, w_1) \lor womanpropose'(m_2, w_1) \leftarrow$$
$$accept'(m_2, w_1) \lor womanpropose'(m_1, w_1) \leftarrow$$
$$womanpropose'(m_1, w_2) \leftarrow$$
$$nwomanpropose'(m_1, w_3) \lor naccept'(m_2, w_3) \leftarrow$$
$$womanpropose'(m_2, w_3) \leftarrow$$
$$accept'(m_2, w_3) \lor womanpropose'(m_1, w_3) \leftarrow$$
$$nmanpropose'(m_2, w_3) \leftarrow$$
$$nwomanpropose'(m_2, w_2) \leftarrow$$

$$sat \leftarrow manpropose'(X, Y), nmanpropose'(X, Y), man(X), woman(Y)$$
$$sat \leftarrow womanpropose'(X, Y), nwomanpropose'(X, Y), man(X), woman(Y)$$
$$sat \leftarrow accept'(X, Y), naccept'(X, Y), man(X), woman(Y)$$
$$sat \leftarrow accept'(X, X), naccept'(X, X), man(X)$$

$$sat \leftarrow accept'(X, X), naccept'(X, X), woman(X)$$
$$mancost'(1, 1) \leftarrow accept'(m_1, w_1)$$
$$mancost'(1, 2) \leftarrow accept'(m_1, w_2)$$
$$mancost'(1, 2) \leftarrow accept'(m_1, w_3)$$
$$mancost'(1, 3) \leftarrow accept'(m_1, m_1)$$
$$mancost'(2, 2) \leftarrow accept'(m_2, w_1)$$
$$mancost'(2, 1) \leftarrow accept'(m_2, w_2)$$
$$mancost'(2, 2) \leftarrow accept'(m_2, m_2)$$
$$womancost'(1, 1) \leftarrow accept'(m_1, w_1)$$
$$womancost'(1, 1) \leftarrow accept'(m_2, w_1)$$
$$womancost'(1, 2) \leftarrow accept'(w_1, w_1)$$
$$womancost'(2, 1) \leftarrow accept'(m_1, w_2)$$
$$womancost'(2, 2) \leftarrow accept'(w_2, w_2)$$
$$womancost'(3, 2) \leftarrow accept'(m_1, w_3)$$
$$womancost'(3, 1) \leftarrow accept'(m_2, w_3)$$
$$womancost'(3, 3) \leftarrow accept'(w_3, w_3)$$
$$manmax'(2, X) \leftarrow mancost'(2, X)$$
$$manmax'(J, X) \leftarrow manmax'(I, X), mancost'(J, Y), X >= Y, \#succ(J, I)$$
$$manmax'(J, X) \leftarrow manmax'(I, X), mancost'(J, Y), X >= Y, \#succ(J, I)$$
$$manregret'(Z) \leftarrow manmax'(1, Z)$$
$$womanmax'(2, X) \leftarrow womancost'(2, X)$$
$$womanmax'(J, X) \leftarrow womanmax'(I, X), womancost'(J, Y), X >= Y, \#succ(J, I)$$
$$womanmax'(J, X) \leftarrow womanmax'(I, X), womancost'(J, Y), X >= Y, \#succ(J, I)$$
$$womanregret'(Z) \leftarrow womanmax'(1, Z)$$
$$regret'(X) \leftarrow manregret'(X), womanregret'(Y), X > Y$$
$$regret'(Y) \leftarrow manregret'(X), womanregret'(Y), X <= Y$$
$$sat \leftarrow regret(X), regret'(Y), X <= Y$$
$$\leftarrow not\ sat$$
$$manargcost'_1(1..2) \leftarrow$$
$$manargcost'_2(1..4) \leftarrow$$
$$womanargcost'_1(1..3) \leftarrow$$
$$womanargcost'_1(1..3) \leftarrow$$
$$mancost'(X, Y) \leftarrow sat, manargcost'_1(X), manargcost'_2(Y)$$
$$womancost'(X, Y) \leftarrow sat, womanargcost'_1(X), womanargcost'_2(Y)$$
$$manpropose'(X, Y) \leftarrow sat, man(X), woman(Y)$$
$$nmanpropose'(X, Y) \leftarrow sat, man(X), woman(Y)$$

$$womanpropose'(X, Y) \leftarrow sat, man(X), woman(Y)$$
$$nwomanpropose'(X, Y) \leftarrow sat, man(X), woman(Y)$$
$$accept'(X, Y) \leftarrow sat, man(X), woman(Y)$$
$$accept'(X, X) \leftarrow sat, man(X)$$
$$accept'(X, X) \leftarrow sat, woman(X)$$
$$naccept'(X, Y) \leftarrow sat, man(X), woman(Y)$$
$$naccept'(X, X) \leftarrow sat, man(X)$$
$$naccept'(X, X) \leftarrow sat, woman(X)$$

Letting DLV compute the unique answer set of this disjunctive ASP program and filtering it to the literals $accept$ and $regret$, yields $\{accept(m_2, w_1),\ accept(m_1, w_3),\ accept(w_2, w_2),\ regret(2)\}$, corresponding exactly to the minimum regret stable set of the SMP instance and the corresponding regret cost.

Proposition 4
Let the criterion crit be an element of $\{sexeq, weight, regret, singles\}$. For every answer set I of the program $\mathcal{P}_{crit}$ induced by an SMP instance with unacceptability and ties the set $S_I = \{(m, w) \mid accept(m, w) \in I\}$ forms an optimal stable set of marriages w.r.t. criterion crit and the optimal criterion value is given by the unique value v_I for which $crit(v_I) \in I$. Conversely for every optimal stable set $S = \{(x_1, y_1), \ldots, (x_k, y_k)\}$ with optimal criterion value v there exists an answer set I of $\mathcal{P}_{crit}$ such that $\{(x, y) \mid accept(x, y) \in I\} = \{(x_i, y_i) \mid i \in \{1, \ldots, k\}\}$ and v is the unique value for which $crit(v) \in I$.

Proof. Let (S_M, S_W) is an instance of the SMP with unacceptability and ties.
$\boxed{\text{Answer set} \Rightarrow \text{Optimal stable set}}$ Let I be an arbitrary answer set of $\mathcal{P}_{crit}$ and let S_I be as in the proposition. It is clear that the only rules in $\mathcal{P}_{crit}$ that influence the literals of the form $manpropose(., .)$, $womanpropose(., .)$ and $accept(., .)$ are the rules in $\mathcal{P}_{norm}$. Hence any answer set I of $\mathcal{P}_{crit}$ should contain an answer set I_{norm} of $\mathcal{P}_{norm}$ as a subset. Proposition 1 implies that I_{norm} corresponds to a stable set $S_I = \{(m, w) \mid accept(m, w) \in I_{norm}\}$. Moreover, the only literals of the form $manpropose(., .)$, $womanpropose(., .)$ and $accept(., .)$ in I are those in I_{norm}, so $S_I = \{(m, w) \mid accept(m, w) \in I\}$. If $crit = sexeq$, it is straightforward to see that the literals of the form $accept(., .)$ in I_{norm} uniquely determine which literals of the form $mancost(., .)$, $womancost(., .)$, $manweight(.)$, $womanweight(.)$ and $sexeq(.)$ should be in the answer set I. These literals do not occur in rules of $\mathcal{P}_{crit}$ besides those in $\mathcal{P}_{ext}^{sexeq}$. Notice that the rules which do contain these literals will imply that there will be just one literal of the form $sexeq(.)$ in I, namely $sexeq(v)$ with v the sex-equalness cost of S_I. Analogous results can be derived for $crit \in \{weight, regret, singles\}$. It remains to be shown that S_I is an optimal stable set. Suppose by contradiction that S_I is not optimal, so there exists a stable set S^* such that $v_I > v^*$, with v^* the criterion value of S^* to be minimized. We will prove that this implies that I cannot be an answer set of $\mathcal{P}_{crit}$, contradicting our initial assumption.
Proposition 2 and Lemma 3 imply that there exists an interpretation I_{disj}^* of the ASP program $\mathcal{P}_{disj}$ induced by (S_M, S_W) that corresponds to the stable set S^*. Moreover this interpretation is consistent, i.e. it will not contain $atom$ and $\neg atom$ for some atom $atom$. This implies that the interpretation I'_{disj} defined as I_{disj}^* in which $\neg atom$ is replaced by $natom$ for every atom $atom$ will falsify the body of the rules of the form (13) of $\mathcal{P}_{ext}^{\prime crit}$. An analogous reasoning as above yields that the literals of the

form $accept'(.,.)$ in I'_{disj} uniquely determine which literals of the form $mancost'(.,.)$, $womancost'(.,.)$, $mansum'(.,.)$, $womansum'(.,.)$, $manweight'(.)$, $womanweight'(.)$ and $sexeq'(.)$ should be in I'_{disj}. With those extra literals added to I'_{disj}, I'_{disj} satisfies all the rules of $\mathcal{P}'^{crit}_{ext}$. Moreover, $crit(v^*)$ is the unique literal of the form $crit(.)$ in I'_{disj}. Notice that I'_{disj} does not contain the atom sat.

Define the interpretation $J = I_{norm} \cup I'_{disj}$. From the previous argument it follows that J will satisfy every rule of $\mathcal{P}^{crit}_{ext} \cup \mathcal{P}'^{crit}_{ext}$ since the predicates occurring in both programs do not overlap. Moreover J contains $crit(v_I)$ and $crit'(v^*)$ and these are the only literals of the form $crit(.)$ or $crit'(.)$. Since $v_I > v^*$ the rules of the form (15) will be satisfied by J since their body is always false. Call J' the set $J \cup \{a \,|\, (a \leftarrow) \in \mathcal{P}_{sat}\}$. Since J' does not contain sat, the rules of $\mathcal{P}_{sat}$ will all be satisfied by J', with exception of the rule $\leftarrow not\,sat$.

The rule of the form (16) implies that I as answer set of $\mathcal{P}_{crit}$ should contain sat. Now the set of rules (17) – (18) imply that I should also contain the literals $mancost'(.,.)$, $womancost'(.,.)$ and $manpropose'(.,.)$, $womanpropose'(.,.)$, $accept'(.,.)$ with the corresponding literals prefixed by n for every possible argument stated by the facts in $\mathcal{P}_{sat}$. The successively computing rules (14) resp. (23) and (25) in $\mathcal{P}'^{crit}_{ext}$, by which we replaced rules (10) – (11) resp. (20) – (21) and (24), garantuee that for every possible set of marriages and its corresponding criterium value c I will contain $crit(c)$ and all associated intermediate results. E.g. for $crit = sexeq$, the rules will garantuee that I also contains $mansum'(.,.)$, $manweight(.)$, $womansum(.,.)$ and $womanweight(.)$ for every argument that could occur in a model of $\mathcal{P}'^{crit}_{ext}$. [8] This implies that $I'_{disj} \subseteq I$. We already reasoned in the beginning of the proof that $I_{norm} \subseteq I$ holds so it follows that $J \subseteq I$. Since the literals of $J' \setminus J$ are stated as facts of $\mathcal{P}^{crit}_{ext}$, they should be in I, hence $J' \subseteq I$. Moreover $J' \subset I$ since $sat \in I \setminus J'$.

We use the notation $red(\mathcal{P}, I)$ to denote the reduct of an ASP program $\mathcal{P}$ w.r.t. an interpretation I. There is no rule in $\mathcal{P}'^{crit}_{ext}$ with negation-as-failure in the body, hence $red(\mathcal{P}'^{crit}_{ext}, I) = red(\mathcal{P}'^{crit}_{ext}, J') = \mathcal{P}'^{crit}_{ext}$. We already reasoned that J' satifies all the rules of the latter. We also reasoned that I does not contain any other literals of the form $accept(.,.)$ than those who are also in I_{norm}, and by construction the same holds for J'. Hence $red(\mathcal{P}^{crit}_{ext}, I) = red(\mathcal{P}^{crit}_{ext}, J')$ and by construction J' satisfies all the rules of this reduct. It is clear that $red(\mathcal{P}_{sat}, I)$ is $\mathcal{P}_{sat}$ without the rule $\leftarrow not\,sat$, since $sat \in I$. Again we already argued that J' satisfies $red(\mathcal{P}_{sat}, I)$. Hence J' satisfies all the rules of $red(\mathcal{P}_{crit}, I)$, implying that I, which strictly contains J', cannot be an answer set of $\mathcal{P}_{crit}$ since it is not a minimal model of the negation-free ASP program $red(\mathcal{P}_{crit}, I)$ [9].

$\boxed{\text{Optimal stable set} \Rightarrow \text{Answer set}}$ Let (S_M, S_W) be an instance of the SMP with unacceptability and ties and let $S = \{(x_1, y_1), \ldots, (x_k, y_k)\}$ be an optimal stable set with optimal criterion value v. To see that the second part of the proposition holds it suffices to verify that the following interpretation I is an answer set of $\mathcal{P}_{crit}$, with the notation $P_{x_i}(y)$ as the index a for which $y \in \sigma^l_M(a)$ if $x_i = m_l$ and symmetrically $P_{y_i}(x)$ as the index a for which $x \in \sigma^{l'}_W(a)$ if $y_i = w_{l'}$. If $x_i = y_i$ we set $P_{x_i}(y_i) = P_{y_i}(x_i) = |\sigma^i_M|$ if x_i is a man and $|\sigma^i_W|$ otherwise. So let I be given by:

$$I = I_1 \cup I_2$$

---

[8] Notice that this would not be the case if we use the original rules with $\#sum$, $\#max$ and $\#count$ in $\mathcal{P}'^{crit}_{ext}$, since these rules would lead to only one value c_M for which e.g. $manweight(c_M)$ should be in I, and similarly only one value c_W for which $womanweight(c_W)$ should be in I. Consequently there would be only one value c such that $crit(c)$ should be in I. This value would not necessarily correspond to v^* and so we would not be able to conclude that $I'_{disj} \subseteq I$. Moreover DLV does not allow the use of these rules because of the cyclic dependency of literals they would create, involving the variables in the aggregate functions.

with

$$I_1 = \{accept(x_i, y_i) \mid i \in \{1, \ldots, k\}\}\{crit(v)\} \cup \{sat\}$$
$$\cup \{womanpropose(x_i, y_i) \mid x_i \neq y_i\}\{manpropose(x_i, y_i) \mid x_i \neq y_i\}$$
$$\cup \{manpropose(x_i, y) \mid i \in \{1, \ldots, k\}, x_i = m_l, \exists a < P_{x_i}(y_i) : y \in \sigma_M^l(a)\}$$
$$\cup \{womanpropose(x, y_i) \mid i \in \{1, \ldots, k\}, y_i = w_{l'}, \exists a < P_{y_i}(x_i) : x \in \sigma_W^{l'}(a)\}$$
$$\cup \{mancost(l, P_{x_i}(y_i)) \mid crit \neq singles, i \in \{1, \ldots, k\}, x_i = m_l\}$$
$$\cup \{womancost(P_{y_i}(x_i), l') \mid crit \neq singles, i \in \{1, \ldots, k\}, y_i = w_{l'}\}$$
$$\cup \{manweight(c_M(S)) \mid crit \in \{sexeq, weight\}\}$$
$$\cup \{womanweight(c_W(S)) \mid crit \in \{sexeq, weight\}\}$$
$$\cup \{manregret(c_{regret,M}(S)) \mid crit = regret\}$$
$$\cup \{womanregret(c_{regret,W}(S)) \mid crit = regret\}$$

and

$$I_2 = \{manargcost'_1(z) \mid z \in \{1, \ldots, n\}\}$$
$$\cup \{manargcost'_2(z) \mid z \in \{1, \ldots, p+1\}\}$$
$$\cup \{womanargcost'_1(z) \mid z \in \{1, \ldots, p\}\}$$
$$\cup \{womanargcost'_2(z) \mid z \in \{1, \ldots, n+1\}\}$$
$$\cup \{man(x) \mid x \in M\} \cup \{woman(x) \mid x \in W\} \tag{26}$$
$$\cup \{mancost'(i, j) \mid crit \neq singles, i \in \{1, \ldots, n\}, j \in \{1, \ldots, p+1\}\}$$
$$\cup \{womancost'(j, i) \mid crit \neq singles, i \in \{1, \ldots, n+1\}, j \in \{1, \ldots, p\}\} \tag{27}$$
$$\cup \{manpropose'(x, y) \mid x \in M, y \in W\} \cup \{womanpropose'(x, y) \mid x \in M, y \in W\}$$
$$\cup \{accept'(x, y) \mid x \in M, y \in W\} \cup \{accept'(x, x) \mid x \in M \cup W\}$$
$$\cup \{nmanpropose'(x, y) \mid x \in M, y \in W\}$$
$$\cup \{nwomanpropose'(x, y) \mid x \in M, y \in W\}$$
$$\cup \{naccept'(x, y) \mid x \in M, y \in W\} \cup \{naccept'(x, x) \mid x \in M \cup W\} \tag{28}$$
$$\cup \{crit'(val) \mid val \in \arg(crit)\}$$
$$\cup \{single'(i, j) \mid crit = singles, i \in \{1, \ldots, n+p\}, j \in \{0, 1\}\}$$
$$\cup \{singlesum'(i, j) \mid crit = singles, i \in \{1, \ldots, n+p\}, j \in \{1, \ldots, (n+p-i+1)\}\}$$
$$\cup \{mansum'(i, j) \mid crit \in \{sexeq, weight\}, i \in \{1, \ldots, n\}, j \in \{n-i+1, \ldots, (n-i+1)(p+1)\}\}$$
$$\cup \{womansum'(j, i) \mid crit \in \{sexeq, weight\}, j \in \{1, \ldots, p\}, i \in \{p-j+1, \ldots, (p-i+1)(n+1)\}\}$$
$$\cup \{manweight'(z) \mid crit \in \{sexeq, weight\}, z \in \{n, \ldots, n(p+1)\}\}$$
$$\cup \{womanweight'(z) \mid crit \in \{sexeq, weight\}, z \in \{p, \ldots, p(n+1)\}\}$$
$$\cup \{manmax'(i, j) \mid crit = regret, i \in \{1, \ldots, n\}, j \in \{1, \ldots, p+1\}\}$$
$$\cup \{womanmax'(j, i) \mid crit = regret, j \in \{1, \ldots, p\}, i \in \{1, \ldots, n+1\}$$
$$\cup \{manregret'(z) \mid crit = regret, z \in \{1, \ldots, p+1\}\}$$
$$\cup \{womanregret'(z) \mid crit = regret, z \in \{1, \ldots, n+1\}\} \tag{29}$$

The notation $\arg(c)$ stands for the possible values the criterion can take within this problem instance:

- $crit = sexeq \Rightarrow \arg(crit) = \{0, \ldots, \max(np + n - p, np + p - n)\}$,

- $crit = weight \Rightarrow \arg(crit) = \{n + p, \ldots, 2np + p + n\}$,

- $crit = regret \Rightarrow \arg(crit) = \{1, \ldots, \max(p, n) + 1)\}$,

- $crit = singles \Rightarrow \arg(crit) = \{0, \ldots, n + p\}$.

To verify wether this interpretation is an answer set of $\mathcal{P}_{crit}$, we should compute the reduct w.r.t. I and check wether I is a minimal model of the reduct [9]. It can readily be checked that I satisfies all the rules of $red(\mathcal{P}_{crit}, I)$. It remains te be shown that there is no strict subset of I with satisfies all the rules. First of all all the facts of $\mathcal{P}_{crit}$ must be in the minimal model of the reduct, explaining why the sets of literals (26) should be in I. The only rules with negation-as-failure are part of $\mathcal{P}_{ext}^{crit}$. As in the previous part of the proof, it is straightforward to see that I_1 is the unique minimal model of the reduct of $\mathcal{P}_{ext}^{crit}$ w.r.t. I, considering that the literals in I_2 don't occur in $\mathcal{P}_{ext}^{crit}$. So any minimal model of $red(\mathcal{P}_{crit}, I)$ must contain I_1.

The key rule which makes sure that I is a minimal model of the reduct is (15). The rules (13) imply that for each model of $red(\mathcal{P}_{crit}, I)$ that does not contain sat, the literals of $\mathcal{P}_{ext}^{\prime crit}$ in that model will correspond to a stable set of the SMP instance. In that case rule (15) will have a true body, since S is optimal, implying that sat should have been in the model. And the presence of sat in any minimal model implies the presence of the set of literals (29) in any minimal model of the reduct. This can be seen with the following reasoning. Due to the presence of the facts (26) and sat in any minimal model of the reduct, rules (17) imply the presence of the literals (27) in any minimal model. For the same reason rules (18) imply that the literals (28) should be in any minimal model of $red(\mathcal{P}_{crit}, I)$. For $crit = sexeq$ the presence of the literals of the form (27) in any minimal model of the reduct together with rules (14) imply that $mansum'(i, j)$ should be in any minimal model for every $i \in \{1, \ldots, n\}$ and $j \in \{n - i + 1, \ldots, (n - i + 1)(p + 1)\}$: for $i = n$ the first rule of (14) implies that $mansum'(n, x)$ is in any minimal model for every x such that $manargcost'_2(x)$ is in it, i.e. any $x \in \{1, \ldots, p+1\}$. Now the second rule of (14) implies that $mansum'(n-1, x)$ is in any minimal model for every $x + y$ such that $manargcost'_2(x)$ and $mansum'(n, y)$ are in it, i.e. any $x + y \in \{2, \ldots, 2(p+1)\}$. If we continue like this, it is straightforward that every literal of the form $mansum'(., .)$ of I_2 should be in any minimal model. The third rule of (14) now implies that $manweight'(x)$ should be in any minimal model for every x such that $mansum'(1, x)$ is in it, i.e. $x \in \{n, \ldots, n(p + 1)\}$. The same reasoning can be repeated for the literals $womansum'$ and $womanweight'$. At this point rules (12) imply that $sexeq'(|x - y|)$ should be in any minimal model which contains $manweight'(x)$ and $womanweight'(y)$. Notice that only one of the two rules in (12) will apply for every x and y since the numerical variables in DLV are positive. Considering the arguments for which $manweight'$ and $womanweight'$ should be in any minimal model, it follows that $sexeq'(x)$ should be in any minimal model for every $x \in \{0, \ldots, \max(p(n+1) - n, n(p+1) - p)\}$, which is exactly $\arg(crit)$. For the other criteria, an analogous reasoning shows that the presence of all literals of I_2 is required in any minimal model of the reduct.

Considering the fact that we have proved that all literals of I should be in any minimal model of the reduct and I fulfils all the rules of the reduct, we know that I is a minimal model of the reduct and thus an answer set of $\mathcal{P}_{crit}$.

$\square$

If we delete from $\mathcal{P}_{sexeq}$ the rules (11) – (12) and replace rule (15) by the rule $sat \leftarrow manweight(X)$, $manweight'(Y), X \leq Y$, then we obtain the M-optimal stable sets. Analogously we can obtain the W-optimal stable sets.

If a criterion is to be maximized, the symbol $\leq$ in rule (15) is simply replaced by $\geq$. E.g. for $crit = singles$ we will get minimum cardinality stable sets.

5 Complexity and Future Work

The NP-complete decision problem 'does there exist a stable set with cardinality $\geq k$ (resp. $\leq k$) for an SMP instance with unacceptability and ties with k a positive integer?' [17, 18] has practical importance, e.g. in the National Resident Matching Program [18]. If we add a rule $sat \leftarrow singles(X), X \leq (n+p-2k)$ to the extended induced program $\mathcal{P}_{ext}^{singles}$ defined in Subsect. 4.3, then this problem can be formulated as 'does there exist an answer set of the normal ASP program $\mathcal{P}_{ext}^{singles}$ which contains the literal sat?' (i.e. brave reasoning), another NP-complete problem [1]. So our model forms a suitable framework for these kind of decision problems concerning optimality of stable sets in the SMP.

Notice that the complexity of this kind of decision problem and the one mentioned in the last paragraph of Subsect. 3 are a good indication how hard it is to find an (optimal) stable set, as opposed to the problems 'does there exist an (optimal) stable set?', which tell us how hard it is to know whether there exists a solution but not necessarily how hard it is to find one.

Combining these problems leads to a new decision problem: 'is the pair (m, w) *optimally stable* for an instance of the SMP with unacceptability and ties?'. We define an *optimally stable pair* as a pair (m, w) for which there exists an optimal stable set in which m and w are matched. As far as we know this problem has not been studied yet, although it could be useful in practice, for instance if one wants to find a maximum cardinality matching but also wants to prioritize some couple or a person. Optimality is still desirable, because it ensures the others from not being put too much at a disadvantage. For instance in the kidney exchange problem, in which kidney patients with a willing but incompatible donor try to interchange each other's donors to get a transplant, this is a realistic situation: if two patients with intercompatible donors urgently need a transplant, they should get priority, but of course we still want to match as many patients to donors as possible. Considering the complexity of the separate decision problems, the combined problem might have a higher complexity, perhaps corresponding to the Σ_2^P-complexity of our grounded disjunctive normal ASP program with aggregate functions [1, 3]. It should be noticed however that the addition of constraints not necessarily increases complexity and a precise classification of complexity is desirable.

6 Conclusion

We formalized and solved different variants of the SMP using ASP programs, which can easily be adapted to yet other variants. Moreover we applied saturation to compute optimal stable sets, with the advantage that these programs can be handled with the efficient off-the-shelf ASP solver DLV. To the best of our knowledge, our encoding offers the first exact implementation of finding sex-equal, egalitarian, minimum regret, or maximum cardinality stable sets for an instance of the SMP with unacceptability and ties. Hence, our general framework allows us to tackle a class of problems and requires only small adaptions to easily shift between them.

References

[1] C. Baral. *Knowledge Representation, Reasoning, and Declarative Problem Solving*. Cambridge University Press, New York, NY, USA, 2003.

[2] G. Brewka, T. Eiter, and M. Truszczyński. Answer set programming at a glance. *Communications of the ACM*, 54(12):92–103, 2011.

[3] T. Dell'Armi, W. Faber, G. Ielpa, N. Leone, and G. Pfeifer. Aggregate functions in disjunctive logic programming: Semantics, complexity, and implementation in DLV. In G. Gottlob and T. Walsh, editors, *IJCAI*, pages 847–852. M. Kaufmann, 2003.

[4] P.M. Dung. An argumentation-theoretic foundation for logic programming. *The Journal of Logic Programming*, 22(2):151–177, 1995.

[5] T. Eiter, G. Gottlob, and H. Mannila. Disjunctive datalog. *ACM Transactions on Database Systems*, 22(3):364–418, 1997.

[6] E. Erdem and V. Lifschitz. Tight logic programs. *Theory and Practice of Logic Programming*, 3:499–518, 2003.

[7] D. Gale and L.S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[8] D. Gale and M. Sotomayor. Some remarks on the stable matching problem. *Discr. Appl. Math.*, 11:223–232, 1985.

[9] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *ICLP/SLP*, pages 1070–1080, 1988.

[10] D. Gusfield. Three fast algorithms for four problems in stable marriage. *SIAM J. Comput.*, 16(1):111–128, 1987.

[11] R.W. Irving. Stable marriage and indifference. *Discr. Appl. Math.*, 48(3):261–272, 1994.

[12] R.W. Irving. The cycle roommates problem: a hard case of kidney exchange. *Inf. Process. Lett.*, 103(1):1–4, 2007.

[13] R.W. Irving, P. Leather, and D. Gusfield. An efficient algorithm for the "optimal" stable marriage. *J. ACM*, 34(3):532–543, 1987.

[14] K. Iwama and S. Miyazaki. A survey of the stable marriage problem and its variants. In *Proc. of the Intern. Conf. on Informatics Educ. and Research for Knowledge-Circulating Society*, ICKS'08, pages 131–136. IEEE Computer Society, 2008.

[15] T. Janhunen. Representing normal programs with clauses. In *In Proc. of the 16th European Conference on Artificial Intelligence*, pages 358–362. IOS Press, 2004.

[16] A. Kato. Complexity of the sex-equal stable marriage problem. *Japan Journal of Industrial ans Applied Mathematics (JJIAM)*, 10:1–19, 1993.

[17] D. Manlove. Stable marriage with ties and unacceptable partners. Technical report, University of Glasgow, Department of Computing Science, 1999.

[18] D. Manlove, R.W. Irving, K. Iwama, S. Miyazaki, and Y. Morita. Hard variants of stable marriage. *Theoretical Computer Science*, 276(1-2):261–279, 2002.

[19] V.W. Marek, A. Nerode, and J.B. Remmel. A theory of nonmonotonic rule systems I. *Ann. Math. Artif. Intell.*, 1:241–273, 1990.

[20] E. McDermid and R.W. Irving. Sex-equal stable matchings: Complexity and exact algorithms. *Algorithmica*, pages 1–26, 2012.

[21] A.E. Roth, T. Sömnez, and M.U. Ünver. Pairwise kidney exchange. *J. Econ. Theory*, 125(2):151 – 188, 2005.

[22] A.E. Roth and M.A.O. Sotomayor. *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*. Cambridge University Press, 1990.

[23] H. Xu and B. Li. Egalitarian stable matching for VM migration in cloud computing. In *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, pages 631–636, 2011.