TLP **16** (3): 247–268, 2016. © Cambridge University Press 2016
doi:10.1017/S147106841600003X First published online 7 March 2016

Solving stable matching problems using answer set programming

SOFIE DE CLERCQ
Department of Applied Mathematics, Computer Science & Statistics, Ghent University, Ghent, Belgium
(*e-mail:* SofieR.DeClercq@ugent.be)

STEVEN SCHOCKAERT
School of Computer Science & Informatics, Cardiff University, Cardiff, UK
(*e-mail:* SchockaertS1@cardiff.ac.uk)

MARTINE DE COCK
Center for Data Science, UW Tacoma, Tacoma, US
Deparment of Applied Mathematics, Computer Science & Statistics, Ghent University, Ghent, Belgium
(*e-mail:* MDeCock@u.washington.edu)

ANN NOWE
Computational Modeling Lab, Vrije Universiteit Brussel, Brussels, Belgium
(*e-mail:* ANowe@vub.ac.be)

submitted 9 December 2014; revised 2 July 2015; accepted 14 December 2015

Abstract

Since the introduction of the stable marriage problem (SMP) by Gale and Shapley (1962), several variants and extensions have been investigated. While this variety is useful to widen the application potential, each variant requires a new algorithm for finding the stable matchings. To address this issue, we propose an encoding of the SMP using answer set programming (ASP), which can straightforwardly be adapted and extended to suit the needs of specific applications. The use of ASP also means that we can take advantage of highly efficient off-the-shelf solvers. To illustrate the flexibility of our approach, we show how our ASP encoding naturally allows us to select optimal stable matchings, i.e. matchings that are optimal according to some user-specified criterion. To the best of our knowledge, our encoding offers the first exact implementation to find sex-equal, minimum regret, egalitarian or maximum cardinality stable matchings for SMP instances in which individuals may designate unacceptable partners and ties between preferences are allowed.

KEYWORDS: Answer set programming, Logic rules, Stable marriage problem, Optimal stable matchings

1 Introduction

The SMP is a well-known matching problem introduced by Gale and Shapley (1962). The input of an SMP instance consists of (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. A *blocking pair* of an SMP instance

https://doi.org/10.1017/S147106841600003X Published online by Cambridge University Press

consists of a man and a woman who are in different marriages but both prefer each other to their actual partners. Given the problem, one can compute a *stable matching* or *stable set of marriages*, which is defined as a set of n couples (marriages) such that there are no blocking pairs. Due to its practical relevance, countless variants on the SMP have been investigated, enabling a wider range of applications. Examples of such applications include the kidney-exchange problem (Irving 2007), which matches donors in incompatible donor-recipient pairs to compatible recipients in other incompatible pairs and vice versa, and the hospital-resident problem (Manlove *et al.* 2002), which matches residents to the free positions in hospitals. In 2012, Roth and Shapley won the Nobel Prize for Economics for their theory of stable allocations and the practice of market design, which directly resulted from an application of the SMP.

In the literature, typically each time a new variant or generalization of the SMP or a different optimality criterion is considered, a new algorithm is developed; see e.g. (Gusfield 1987; Irving *et al.* 1987; McDermid and Irving 2012), or (Manlove 2013) for an overview. In this paper, we propose to use ASP as a general vehicle for modeling a large class of extensions and variations of the SMP. We show how ASP encodings can be used to compute stable matchings, and how this encoding can be extended to compute optimal stable matchings. Although the SMP has been widely investigated, and efficient approximation or exact algorithms are available for several of its variants (Iwama *et al.* 2010; McDermid and Irving 2012), to the best of our knowledge, our encoding offers the first exact implementation to find sex-equal, minimum regret, egalitarian or maximum cardinality stable matchings for SMP instances with unacceptable partners and ties (see below).

In this paper, we will consider two well-known adaptions of the SMP. First, we will consider problem instances in which every person can specify a set of unacceptable partners. The second alteration consists of allowing ties in the preferences, i.e. one can be indifferent between some possible partners. In the literature, the SMP variant with unacceptable partners – or, equivalently, with incomplete preference lists – is abbreviated as SMI. The variant with ties is denoted as stable matching problem with ties (SMT) and the variant which allows both extensions as stable matching problem with ties and incomplete preference lists (SMTI). Note that, the original SMP is a special case of the SMTI, i.e. the set of unacceptable partners is empty for each man and woman, and there are no ties. Therefore, our paper focusses on the SMTI variant, as it is the most general one.

Another way to generalize the SMP is by introducing optimality criteria for stable matchings. This is motivated by the fact that, if multiple stable matchings exist, some may be more interesting than others. In this paper, we focus on sex-equality, minimum regret, egalitarity and maximum cardinality of the stable matchings, as these are commonly investigated optimality criteria in the matching literature. Note, however, that there exist several other optimality criteria in the context of matchings, such as *popularity* (Gärdenfors 1975), *Pareto optimality* (Gale and Sotomayor 1985; Roth and Sotomayor 1990) or profile-based notions such as rank-maximum (Irving 2003), greedy maximum and generous maximum (Irving 2003). As this list is non-exhaustive, we refer the interested reader to (Manlove 2013) for an overview.

https://doi.org/10.1017/S147106841600003X Published online by Cambridge University Press

The structure of the paper is as follows. First we give some background about the SMP and ASP in Section 2. Then, we introduce our encoding of the SMTI with ASP and prove its correctness in Section 3.1. To illustrate the flexibility of the approach, we show how it can be used to tackle three-dimensional stable matching problems in Section 3.2. In Section 4, we first discuss several optimality criteria and then extend the encoding from Section 3.1, enabling us to find optimal stable matchings. We show how optimal stable matchings of an SMTI instance can be found by solving the corresponding induced disjunctive ASP program and prove the soundness of our approach. This paper is an extended version of De Clercq *et al.* (2013) and additionally provides detailed examples, complete correctness proofs and an ASP encoding of the three-dimensional stable matching problem. The three-dimensional stable matching problem is very important for practical applications, such as the kidney exchange program (Biró and McDermid 2010).

2 Background

2.1 The stable marriage problem

To solve the original SMP, Gale and Shapley (1962) constructed an iterative algorithm—known as the Gale-Shapley algorithm, G-S algorithm or deferred-acceptance algorithm—to compute a particular stable matching 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 among 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 (Gale and Shapley 1962). It should be noted, however, that only one of the potentially exponentially many stable matchings is found in this way.

The classical SMP can be generalized by (i) allowing men and women to point out unacceptable partners, i.e. exclude them from their preference list and (ii) dropping the restriction that the number of men n equals the number of women p. In this variant, men and women can remain single in a stable matching. Intuitively, one prefers remaining single over being matched with an unacceptable partner. This variant is also referred to as the SMP with incomplete preference lists, abbreviated as SMI. A stable matching for an SMI instance always exists and can be found in polynomial time (Roth and Sotomayor 1990) by a slightly modified G-S algorithm. As we focus on an extension of the SMI, we refer to the online appendix for the formal definitions of the classical SMP and the SMI.

The SMI variant can further be generalized by additionally allowing ties in the preference lists. For this variant (SMTI) there are several ways to define stability,

https://doi.org/10.1017/S147106841600003X Published online by Cambridge University Press

but we will use the notion of weak stability (Irving 1994). We denote a set of men as $M = \{m_1, \ldots, m_n\}$ and a set of women $W = \{w_1, \ldots, w_p\}$. A set of marriages or a matching is a collection of man-woman pairs and singles (persons paired to themselves) such that every man and every woman occurs in just one pair.

Definition 1 (*SMTI*)
An instance of the SMTI is a pair (S_M, S_W) with $S_M = \{\sigma_M^1, \ldots, \sigma_M^n\}$ and $S_W = \{\sigma_W^1, \ldots, \sigma_W^p\}$. 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 and we denote the length of the list σ_M^i as $|\sigma_M^i|$. If $k \in \sigma_M^i(j)$, woman w_k is in man m_i's jth 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 similarly when there is no l such that $j \in \sigma_W^i(l)$. For every k in the set $\sigma_M^i(|\sigma_M^i|)$, 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 strictly prefers being single to being paired to his partner in S. A weakly stable matching is a set of marriages without blocking pairs or individuals.

A weakly stable matching always exists for an instance of the SMTI and it can be found in polynomial time by arbitrarily breaking the ties (Iwama and Miyazaki 2008). However, as opposed to the SMI variant, the number of matched persons is no longer constant for every stable matching in this variant.

We introduce the following 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 $\preccurlyeq_M^{m_i}$ on $\{w_j \mid j \in acceptable_M^i\} \cup \{m_i\}$ as $x \preccurlyeq_M^{m_i} y$ iff m_i prefers person x at least as much as person y. Note that m_i is included in its own preference ordering to encode the possibility of staying single. The strict ordering $\prec_M^{m_i}$ is defined in the obvious way and analogous notations are used for σ_W^j.

Example 1
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 $w_1 \prec_M^{m_1} m_1$, 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 $unacceptable_M^1 = \{2\}$.

https://doi.org/10.1017/S147106841600003X Published online by Cambridge University Press

2.2 Answer set programming

ASP is a form of declarative programming (Brewka *et al.* 2011). 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

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

where $A_1, \ldots, A_k, B_1, \ldots, B_m, C_1, \ldots, C_n$ are predicates, possibly negated by $\neg$, and *not* is the negation-as-failure operator, whose meaning is explained below. 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. (Brewka *et al.* 2011) 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, denoted with $\neg$. 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 $\mathscr{L}$, determined by a fixed set $\mathscr{A}$ 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 $c_1, \ldots, c_n$ is 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 $\mathscr{P}$ 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 $\mathscr{P}$ as a subset of $\mathscr{L}$. An interpretation I *satisfies* a simple disjunctive rule $a_1 \vee \ldots \vee a_k \leftarrow b_1, \ldots, b_m$ when $a_1 \in I$ or $a_2 \in I$ or … or $a_k \in I$ or $\{b_1, \ldots, b_m\} \nsubseteq I$. 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 $\mathscr{P}$ iff it is a minimal model of $\mathscr{P}$, i.e. no strict subset of I is a model of $\mathscr{P}$ (Gelfond and Lifschitz 1988). The *reduct* $\mathscr{P}^I$ of a disjunctive ASP program $\mathscr{P}$ w.r.t. an interpretation I is defined as the simple disjunctive ASP program $\mathscr{P}^I = \{a_1 \vee \ldots \vee a_k \leftarrow b_1, \ldots, b_m \mid (a_1 \vee \ldots \vee a_k \leftarrow b_1, \ldots, b_m, not\ c_1, \ldots, not\ c_n) \in \mathscr{P}, \{c_1, \ldots, c_n\} \cap I = \emptyset\}$. An interpretation I of a disjunctive ASP program $\mathscr{P}$ is an answer set of $\mathscr{P}$ iff I is an answer set of $\mathscr{P}^I$.

https://doi.org/10.1017/S147106841600003X Published online by Cambridge University Press

Example 2
Let $\mathscr{P}$ be the ASP program with the following four rules:

$$man(john) \leftarrow, \quad person(john) \leftarrow, \quad person(fiona) \leftarrow$$
$$woman(X) \vee child(X) \leftarrow person(X), not\ man(X)$$

The first three 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. The latter rule is grounded to two rules in which X is resp. replaced by *john* and *fiona*. We check that the interpretation $I = \{man(john), woman(fiona), person(john), person(fiona)\}$ is an answer set of the ground version of $\mathscr{P}$ by computing the reduct $\mathscr{P}^I$. As the grounded rule with $X = john$ is deleted since $man(john)$ is in I, $\mathscr{P}^I$ is:

$$man(john) \leftarrow, \quad person(john) \leftarrow, \quad person(fiona) \leftarrow$$
$$woman(fiona) \vee child(fiona) \leftarrow person(fiona)$$

It is clear that I is a minimal model of this simple program, so I is an answer set of $\mathscr{P}$. 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 (`www.dlvsystem.com`), due to its ability to handle predicates, disjunction and numeric values, with built-in aggregate functions (Faber *et al.* 2008). The numeric values are only used for grounding.

3 Modeling the stable marriage problem in ASP

3.1 Modeling the SMTI in ASP

In this section, we model the SMTI, using ASP. A few proposals of using non-monotonic reasoning for modeling the SMP have already been described in the literature. For instance, in Marek *et al.* (1990) 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 Dung (1995) an abductive program is used to find a stable matching 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 matchings of SMTI instances allows to easily adjust the encoding to variants of the SMP, such as the stable roommate problem (Gale and Shapley 1962), in which matches need to be found within one group instead of between two groups, or the three-dimensional stable matching problem (Ng and Hirschberg 1991), which matches triples between three groups instead of pairs between two.

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 formalization to find the stable matchings.

https://doi.org/10.1017/S147106841600003X Published online by Cambridge University Press

Definition 2 (*ASP program induced by SMTI*)
The ASP program $\mathscr{P}$ induced by an instance $(\{\sigma_M^1,\ldots,\sigma_M^n\},\{\sigma_W^1,\ldots,\sigma_W^p\})$ of the SMTI is the program containing for every $i \in \{1,\ldots,n\}, j \in \{1,\ldots,p\}$ the following rules:

$$accept(m_i,w_j) \leftarrow manpropose(m_i,w_j), womanpropose(m_i,w_j) \quad (1)$$

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

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

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

$$manpropose(m_i,w_j) \leftarrow \{not\ accept(m_i,x) \mid x \leqslant_M^{m_i} w_j \text{ and } w_j \neq x\} \quad (4)$$

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

$$womanpropose(m_i,w_j) \leftarrow \{not\ accept(x,w_j) \mid x \leqslant_W^{w_j} m_i \text{ and } m_i \neq x\} \quad (5)$$

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 explicit rules are stated about the number of persons someone can propose to or accept but as we will see below, in Proposition 1, this is unnecessary. Note that, for $k = \max(n,p)$, the number of grounded rules in the induced ASP program is $\mathcal{O}(k^2)$. We now illustrate our approach with an example.

Example 3
Consider the following instance (S_M,S_W) of the SMTI. Let $M = \{m_1,m_2\}$ and $W = \{w_1,w_2,w_3\}$. Furthermore:

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

The ASP program induced by this SMTI instance is:

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

https://doi.org/10.1017/S147106841600003X Published online by Cambridge University Press

$$
\begin{aligned}
womanpropose(m_2, w_1) &\leftarrow not\ accept(m_1, w_1)\\
accept(w_1, w_1) &\leftarrow not\ accept(m_1, w_1), not\ accept(m_2, w_1)\\
womanpropose(m_1, w_2) &\leftarrow\\
accept(w_2, w_2) &\leftarrow not\ accept(m_1, w_2)\\
womanpropose(m_2, w_3) &\leftarrow\\
womanpropose(m_1, w_3) &\leftarrow not\ accept(m_2, w_3)\\
accept(w_3, w_3) &\leftarrow not\ accept(m_1, w_3), not\ accept(m_2, w_3)
\end{aligned}
$$

If we run the 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 answer sets correspond to the three weakly stable matching of marriages of this SMTI 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 $\{(m_1, w_1), (m_2, m_2), (w_2, w_2), (w_3, w_3)\}$.

The following proposition states that our ASP encoding is sound, i.e. that there is a bijective correspondence between the answer sets of the induced program and the weak stable matchings of the SMTI. The complete proof is provided in the online appendix.

Proposition 1
Let (S_M, S_W) be an instance of the SMTI and let $\mathscr{P}$ be the corresponding ASP program. If I is an answer set of $\mathscr{P}$, then a weakly stable matching for (S_M, S_W) is given by $\{(x, y) \mid accept(x, y) \in I\}$. Conversely, if $\{(x_1, y_1), \ldots, (x_k, y_k)\}$ is a weakly stable matching for (S_M, S_W) then $\mathscr{P}$ has the following answer set I:

$$
\begin{aligned}
&\{manpropose(x_i, y) \mid i \in \{1, \ldots, k\}, x_i \in M, y <_M^{x_i} y_i \vee y = y_i \neq x_i\}\\
\cup&\{womanpropose(x, y_i) \mid i \in \{1, \ldots, k\}, y_i \in W, x <_W^{y_i} x_i \vee x = x_i \neq y_i\}\\
\cup&\{accept(x_i, y_i) \mid i \in \{1, \ldots, k\}\}
\end{aligned}
$$

A pair (m, w) is *stable* if there exists a stable matching that contains (m, w). In Manlove *et al.* (2002) it is shown that the decision problem 'is the pair (m, w) stable?' for a given SMTI instance is an NP-complete problem, even in the absence of unacceptability. 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 $\mathscr{P}$ which contains the literal $accept(m, w)$?' (i.e. brave reasoning), which is known to be an NP-complete problem (Baral 2003). Thus our model forms a suitable framework for these kind of decision problems concerning the SMTI.

3.2 Modeling the three-dimensional SMTI in ASP

To illustrate further the flexibility of our ASP approach, we consider a variant of the SMP and show how small adaptations of the ASP encoding can solve this variant.

https://doi.org/10.1017/S147106841600003X Published online by Cambridge University Press

Extending the SMP by adding another dimension to the problem was first proposed in Knuth (1976). We work out the three-dimensional SMTI, where n men are to be matched with p women and r children. Definition 1 can straightforwardly be generalized to a three-dimensional instance (S_M, S_W, S_C), in which preference lists of the men are defined over the set of woman-child pairs and similarly the women have preferences over man-child pairs and the children have preferences over man-woman pairs. In the three-dimensional case, σ_M^i becomes a list of disjoint subsets of $\{1,\ldots,p\} \times \{1,\ldots,r\}$, and analogously for σ_W^j and σ_C^k. Similarly as before, we can define the notions of $acceptable_M^i$ etc. Note that the orderings $\leqslant_M^{m_i}$, $\leqslant_W^{w_j}$ and $\leqslant_C^{c_k}$ are resp. defined on pairs in $W \times C$, $M \times C$ and $M \times W$. A stable matching is now defined as a set of man-woman-child triples and singles, with the properties that no man, woman and child can be found such that each of them prefers the pair formed by the others above their current mates in the matching and no person prefers being single to being matched with its current mates. The practical relevance of this problem is pointed out in Biró and McDermid (2010).

Extending the ASP program from Definition 2, we can write an ASP program induced by an instance of the three-dimensional matching problem.

Definition 3 (*ASP program induced by 3D SMTI*)
The ASP program $\mathscr{P}$ induced by an instance $(\{\sigma_M^1,\ldots,\sigma_M^n\}, \{\sigma_W^1,\ldots,\sigma_W^p\}, \{\sigma_C^1, \ldots, \sigma_C^r\})$ of the three-dimensional SMTI is the program containing the following rules for every $i \in \{1,\ldots,n\}, j \in \{1,\ldots,p\}$ and $k \in \{1,\ldots,r\}$:

$$\begin{aligned} accept(m_i, w_j, c_k) &\leftarrow manprop(m_i, w_j, c_k), womprop(m_i, w_j, c_k), \\ &\qquad childprop(m_i, w_j, c_k) \\ accept(m_i, m_i, m_i) &\leftarrow \{not\ accept(m_i, w_u, c_v) \mid (u,v) \in acceptable_M^i\} \\ accept(w_j, w_j, w_j) &\leftarrow \{not\ accept(m_u, w_j, c_v) \mid (u,v) \in acceptable_W^j\} \\ accept(c_k, c_k, c_k) &\leftarrow \{not\ accept(m_u, w_v, c_k) \mid (u,v) \in acceptable_C^k\} \end{aligned}$$

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

$$manprop(m_i, w_j, c_k) \leftarrow \{not\ accept(m_i, x, y) \mid (x,y) \leqslant_M^{m_i} (w_j, c_k); (w_j, c_k) \neq (x,y)\}$$

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

$$womprop(m_i, w_j, c_k) \leftarrow \{not\ accept(x, w_j, y) \mid (x,y) \leqslant_W^{w_j} (m_i, c_k); (m_i, c_k) \neq (x,y)\}$$

and for every $k \in \{1,\ldots,r\}$ and $(i,j) \in acceptable_C^j$:

$$childprop(m_i, w_j, c_k) \leftarrow \{not\ accept(x, u, c_k) \mid (x,y) \leqslant_C^{c_k} (m_i, w_j); (m_i, w_j) \neq (x,y)\}$$

Ng and Hirschberg (1991) proved that deciding whether a stable matching exists for the three-dimensional problem – in the absence of unacceptability and ties – is an NP-complete problem. Completely analogously as for the two-sided SMP, one can prove that this encoding yields a bijective correspondence between the answer sets of the ASP program and the stable matchings of the three-dimensional matching problem. Note that, for $k = \max(n,p,r)$, the number of grounded rules in the induced ASP program is $\mathcal{O}(k^3)$.

https://doi.org/10.1017/S147106841600003X Published online by Cambridge University Press

4 Selecting optimal stable matchings

4.1 Notions of optimality of stable matchings

When several stable matchings can be found for an instance of the SMP, some may be more interesting than others. The stable matching found by the G-S algorithm is *M-optimal* (Roth and Sotomayor 1990), i.e. every man likes this set at least as well as any other stable matching. Exchanging the roles of men and women in the G-S algorithm yields a *W-optimal* stable matching (Gale and Shapley 1962), optimal from the women's point of view.

While some applications may require us to favor 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 $c_x(y)$ of being matched with y for an individual x, where $c_x(y) = k$ if y is x's kth preferred partner. More precisely, for a man m_i, we define $c_{m_i}(y) = |\{z : z <_M^{m_i} y\}| + 1$ for every $y \in acceptable_M^i$; for a woman w_j, c_{w_j} is defined analogously. So in case of ties we assign the same list position to equally preferred partners, as illustrated in Example 4.

Example 4
Let $x = m_1$ be a man with preference list $\sigma_M^1 = (\{1\}, \{2,3\}, \{4\})$ then $c_x(w_1) = 1$, $c_x(w_2) = c_x(w_3) = 2$ and w_4 yields $c_x(w_4) = 4$. The cost for being single would be 4, i.e. $c_x(m_1) = 4$, 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 4 (*Optimal Stable Matchings*)
Let S be a set of marriages and let $S(x)$ denote the partner of x in S.

- The sex-equality cost of S is $c_{sexeq}(S) = |\sum_{x \in M} c_x(S(x)) - \sum_{x \in W} c_x(S(x))|$,
- the egalitarian cost of S is $c_{weight}(S) = \sum_{x \in M \cup W} c_x(S(x))$,
- the regret cost of S is $c_{regret}(S) = \max_{x \in M \cup W} c_x(S(x))$, and
- the cardinality cost of S is $c_{singles}(S) = |\{z : (z,z) \in S\}|$.

S is a sex-equal stable matching iff S is a stable matching with minimal sex-equality cost. Similarly, S is an egalitarian (resp. minimum regret, maximum cardinality) stable matching iff S is a stable matching with minimal egalitarian (resp. regret or cardinality) cost.

A *sex-equal stable matching* assigns an equal importance to the preferences of the men and women, i.e. the men are as pleased with the matching as the women. An *egalitarian stable matching* is a stable matching in which the preferences of every individual are considered to be equally important, i.e. it minimizes the difference in happiness of all the men and women. In Xu and Li (2011) the use of an egalitarian stable matching 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 matching* is optimal for the person who is worst off, i.e. there does not exist a stable matching such that the person who is most displeased with the matching is happier than the most displeased person in the minimum regret stable

https://doi.org/10.1017/S147106841600003X Published online by Cambridge University Press

matching. A *maximal or minimal cardinality stable matching* is a stable matching with resp. as few or as many singles as possible. Examples of practical applications include an efficient kidney exchange program (Roth *et al.* 2005), which matches donors of incompatible pairs to recipients of other incompatible pairs and vice versa, and the National Resident Matching Program (www.nrmp.org) (Manlove *et al.* 2002), which matches healthcare professionals to graduate medical education and advanced training programs. Maximizing cardinality guarantees that as many recipients as possible will get a compatible donor and as many healthcare professionals as possible will get a position.

Remark 1
It might be somewhat confusing that the term *utilitarian* is more frequently used in sociological and economical contexts for an optimization of the overall happiness (here called egalitarity), while *egalitarity* is more used for an optimization which minimizes the unhappiness of the individuals (tending more to minimum regret). However, we will use the cost terms as defined above, since these are standard in the context of the SMP.

For an overview of literature results concerning the computational complexity of finding optimal stable matchings in the SMP, SMI, SMT and SMTI, we refer to the online appendix.

Note that, other notions of preferred matchings have been described in the literature, such as *popularity* (Gärdenfors 1975), *Pareto optimality* (Gale and Sotomayor 1985; Roth and Sotomayor 1990) or profile-based notions such as rank-maximum (Irving 2003), greedy maximum and generous maximum (Irving 2003). For more details on these and other optimality criteria, we refer the interested reader to Manlove (2013) for an overview.

4.2 Finding stable matchings using disjunctive Naf-free ASP

As we discuss in Section 4.3, we can extend our ASP encoding of the SMTI such that the optimal stable matchings correspond to the answer sets of an associated ASP program. In particular, we use the saturation technique (Eiter and Gottlob 1995) to filter non-optimal answer sets. Intuitively, the idea is to create a program with three 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, however, known that the presence of negation-as-failure can cause problems when applying saturation. This is due to the fact that rules containing naf-literals can be altered in the reduct. To address this issue, we use saturation in combination with a disjunctive naf-free ASP program instead of the ASP program in Definition 2. To this end, we use a SAT encoding (Janhunen 2004) of the ASP program in Definition 2 and define a disjunctive naf-free ASP program in Definition 5 which selects particular models of the SAT problem.

https://doi.org/10.1017/S147106841600003X Published online by Cambridge University Press

Note that, our original normal program is absolutely tight, i.e. there is no infinite 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 (Erdem and Lifschitz 2003). We use the completion to derive an ASP encoding for finding optimal stable matchings. 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, where $body'_i$ is the conjunction of literals derived from $body_i$ by replacing every occurrence of '*not*' with '$\neg$'. If an atom a of the program does not occur in any rule head, then $a \equiv \bot$ is in the completion of the program. Similarly the completion of the program contains the propositional formula $\bot \equiv body'_1 \vee \ldots \vee body'_l$ where the disjunction extends over all the program constraints $\leftarrow body_i$ ($i \in \{1, \ldots, l\}$). Because our program is absolutely tight, we know that every propositional model of the completion will correspond to an answer set of the original program and vice versa (Erdem and Lifschitz 2003). When applied to the induced normal ASP program in Definition 2, the completion contains the following formulas, for all $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, p\}$:

$$accept(m_i, w_j) \equiv manpropose(m_i, w_j) \wedge womanpropose(m_i, w_j), \hat{E}$$

$$accept(m_i, m_i) \equiv \bigwedge_{k \in acceptable^i_M} \neg accept(m_i, w_k),$$

$$accept(w_j, w_j) \equiv \bigwedge_{k \in acceptable^j_W} \neg accept(m_k, w_j),$$

and for all $i \in \{1, \ldots, n\}$ and $j \in acceptable^i_M$:

$$manpropose(m_i, w_j) \equiv \bigwedge_{x \leqslant^{m_i}_M w_j, x \neq w_j} \neg accept(m_i, x),$$

and similarly for all $j \in \{1, \ldots, p\}$ and $i \in acceptable^j_W$:

$$womanpropose(m_i, w_j) \equiv \bigwedge_{x \leqslant^{w_j}_W m_i, x \neq m_i} \neg accept(x, w_j),$$

and for all $i \in \{1, \ldots, n\}$ and $j \in unacceptable^i_M$:

$$manpropose(m_i, w_j) \equiv \bot,$$

and similarly for all $j \in \{1, \ldots, p\}$ and $i \in unacceptable^j_W$:

$$womanpropose(m_i, w_j) \equiv \bot .$$

Using these formulas, which form the completion of the normal ASP program from Definition 2, we can define an equivalent disjunctive ASP program without negation-as-failure.

https://doi.org/10.1017/S147106841600003X Published online by Cambridge University Press

Definition 5 (*Induced disj. naf-free ASP program*)
The disjunctive naf-free ASP program $\mathscr{P}_{disj}$ induced by an SMTI instance (S_M, S_W) contains the following rules for every $i \in \{1,\ldots,n\}, j \in \{1,\ldots,p\}$:

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

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

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

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

Note that, for $k = \max(n, p)$, the number of grounded rules in the induced naf-free program is $\mathcal{O}(k^3)$. The following lemma follows from the fact that the completion corresponds to the original program (Erdem and Lifschitz 2003).

Lemma 1
Let $\mathscr{P}$ be the normal ASP program from Definition 2 and $\mathscr{P}_{disj}$ the disjunctive ASP program from Definition 5. It holds that for any answer set I of $\mathscr{P}$ there exists an answer set I_{disj} of $\mathscr{P}_{disj}$ such that the atoms of I and I_{disj} coincide. Conversely for any answer set I_{disj} of $\mathscr{P}_{disj}$ there exists an answer set I of $\mathscr{P}$ 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 SMTI instance with $S_M = \{\sigma^1_M,\ldots,\sigma^n_M\}$ and $S_W = \{\sigma^1_W,\ldots,\sigma^p_W\}$, and let $\mathscr{P}_{norm}$ be the induced normal ASP program from Definition 2. Our technique for extending this program to a program that can respectively optimize for the sex-equality, egalitarity, minimum regret and maximum cardinality criterion is in each case very similar. We start by explaining it for the case of sex-equality. Our first step is to add a set of rules that compute the sex-equality cost of a set of marriages. For every man m_i and every $x \in acceptable^i_M \cup \{m_i\}$ we use the following rule to determine the cost for m_i:

$$mancost(i, c_{m_i}(x)) \leftarrow accept(m_i, x) \tag{6}$$

and similarly for every w_j and every $x \in acceptable^j_W \cup \{w_j\}$:

$$womancost(j, c_{w_j}(x)) \leftarrow accept(x, w_j) \tag{7}$$

https://doi.org/10.1017/S147106841600003X Published online by Cambridge University Press

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

$$manweight(Z) \leftarrow \#sum\{B, A : mancost(A, B)\} = Z, \#int(Z) \quad (8)$$

$$womanweight(Z) \leftarrow \#sum\{B, A : womancost(A, B)\} = Z, \#int(Z) \quad (9)$$

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

$$sexeq(Z) \leftarrow manweight(X), womanweight(Y), Z = Y - X \quad (10)$$

Note that $\#sum$, $\#max$, $\#int$ and $\#count$ are DLV aggregate functions (Faber *et al.* 2008). 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). Rule (8) determines the sum of the male costs and similarly (9) determines the sum of the female costs. According to Definition 4 the absolute difference of these values yields the sex-equality cost, as determined by rules (10). Since numeric variables are restricted to positive integers in DLV, we omit conditions as '$X \geqslant Y$' or '$X < Y$'. The program $\mathscr{P}_{norm}$ extended with rules (6)–(10) is denoted $\mathscr{P}_{ext}^{sexeq}$.

We construct a program $\mathscr{P}_{sexeq}$, composed by subprograms, that selects optimal solutions. Let $\mathscr{P}'_{disj}$ be the disjunctive naf-free ASP program, induced by the same SMTI instance, in which a prime symbol is added to all literal names (e.g. *accept* becomes *accept*′). Define a new program $\mathscr{P}'^{sexeq}_{ext}$ with all the rules of $\mathscr{P}'_{disj}$ in 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 $\mathscr{P}'^{sexeq}_{ext}$, add the following rule to exclude non-consistent solutions:

$$sat \leftarrow atom, natom \quad (11)$$

For instance, the rule $sat \leftarrow accept'(m_1, w_1), naccept'(m_1, w_1)$ is added. Finally add rules (6)–(10) with prime symbols to the literal names to $\mathscr{P}'^{sexeq}_{ext}$ but replace rule (8) and rule (9) by:

$$\begin{aligned}
mansum'(n, X) &\leftarrow mancost'(n, X)\\
mansum'(J, Z) &\leftarrow mansum'(I, X), mancost'(J, Y), Z = X + Y,\ \#succ(J, I)\\
manweight'(Z) &\leftarrow mansum'(1, Z)\\
womansum'(p, X) &\leftarrow womancost'(p, X)\\
womansum'(J, Z) &\leftarrow womansum'(I, X), womancost'(J, Y), Z = X + Y,\ \#succ(J, I)\\
womanweight'(Z) &\leftarrow womansum'(1, Z) \quad (12)
\end{aligned}$$

The DLV aggregate function $\#succ(J, I)$ is true whenever $J + 1 = I$. We replace the rules with the aggregate function $\#max$ by these rules to make sure the saturation happens correctly. When saturation is used, the DLV aggregate functions $\#max$, $\#sum$ and $\#count$ would not yield the right criterion values. Moreover, DLV does not accept these aggregate functions in saturation because of the cyclic dependency of literals within the aggregate functions created by the rules for saturation. These adjusted rules, however, will not pose any problems because of the successive way they compute the criterion values. This becomes more clear in the proof of

https://doi.org/10.1017/S147106841600003X Published online by Cambridge University Press

Proposition 2. We define the ASP program $\mathscr{P}_{sexeq}$ as the union of $\mathscr{P}_{ext}^{sexeq}$, $\mathscr{P}_{ext}^{\prime sexeq}$ and $\mathscr{P}_{sat}$. The ASP program $\mathscr{P}_{sat}$ contains the following rules to select minimal solutions based on sex-equality:

$$sat \leftarrow sexeq(X), sexeq'(Y), X \leqslant Y \tag{13}$$

$$\leftarrow not\ sat \tag{14}$$

$$\begin{aligned} mancost'(X,Y) &\leftarrow sat, manargcost'_1(X), manargcost'_2(Y) \\ womancost'(X,Y) &\leftarrow sat, womanargcost'_1(X), womanargcost'_2(Y) \\ manpropose'(X,Y) &\leftarrow sat, man(X), woman(Y) \\ womanpropose'(X,Y) &\leftarrow sat, man(X), woman(Y) \\ accept'(X,X) &\leftarrow sat, man(X) \\ accept'(X,X) &\leftarrow sat, woman(X) \\ accept'(X,Y) &\leftarrow sat, man(X), woman(Y) \end{aligned} \tag{15, 16}$$

and analogous to (16) a set of rules with prefix '*n*' for the head predicates. Finally, we add the facts $manargcost'_1(1..n) \leftarrow$, $manargcost'_2(1..(p+1)) \leftarrow$, $womanargcost'_1(1..p) \leftarrow$, $womanargcost'_2(1..(n+1)) \leftarrow$, $man(x) \leftarrow$ for every man x and $woman(x) \leftarrow$ for every woman x to $\mathscr{P}_{sat}$. The rule $manargcost'_1(1..n) \leftarrow$ is DLV syntax for the n facts $manargcost'_1(1) \leftarrow, \ldots, manargcost'_1(n) \leftarrow$. Intuitively the rules of $\mathscr{P}_{sat}$ express the key idea of saturation. First every answer set is forced to contain the atom *sat* by rule (14). Then the rules (15)–(16) and the facts make sure that any answer set should contain all possible literals with a prime symbol that occur in $\mathscr{P}_{sexeq}$. Rule (13) will establish that only optimal solutions will correspond to minimal models and thus lead to answer sets. For any non-optimal solution, the corresponding interpretation containing *sat* will never be a minimal model of the reduct. It is formally proven in Proposition 2 below that $\mathscr{P}_{sexeq}$ produces exactly the stable matchings with minimal sex-equality cost.

Furthermore, only small adjustments to $\mathscr{P}_{sexeq}$ are needed to create programs $\mathscr{P}_{weight}$, $\mathscr{P}_{regret}$, and $\mathscr{P}_{singles}$ that respectively produce egalitarian, minimum regret and maximum cardinality stable matchings. Indeed, the ASP program $\mathscr{P}_{weight}$ can easily be defined as $\mathscr{P}_{sexeq}$ in which the predicates *sexeq* and *sexeq*′ are respectively replaced by *weight* and *weight*′ and the rules (10) are replaced by (17), determining the egalitarian cost of Definition 4 as the sum of the male and female costs:

$$weight(Z) \leftarrow manweight(X), womanweight(Y), Z = X + Y \tag{17}$$

Similarly the ASP program $\mathscr{P}_{regret}$ is defined as $\mathscr{P}_{sexeq}$ in which the predicates *sexeq* and *sexeq*′ are resp. replaced by *regret* and *regret*′ and rules (8)–(10) are replaced by the following rules:

$$manregret(Z) \leftarrow \#max\{B : mancost(A,B)\} = Z, \#int(Z) \tag{18}$$

$$womanregret(Z) \leftarrow \#max\{B : womancost(A,B)\} = Z, \#int(Z) \tag{19}$$

$$\begin{aligned} regret(X) &\leftarrow manregret(X), womanregret(Y), X > Y \\ regret(Y) &\leftarrow manregret(X), womanregret(Y), X \leqslant Y \end{aligned} \tag{20}$$

https://doi.org/10.1017/S147106841600003X Published online by Cambridge University Press

Rule (18) determines the regret cost but only for the men. Similarly (19) determines the regret cost for the women. The regret cost as defined in Definition 4 is the maximum of these two values, determined by the rules in (20). Again we adjust rules (18) and (19) for the program part $\mathscr{P}_{ext}^{'regret}$ by replacing them with a variant based on the successor function:

$$\begin{aligned}
manmax'(n,X) &\leftarrow mancost'(n,X)\\
manmax'(J,X) &\leftarrow manmax'(I,X), mancost'(J,Y), X \geqslant Y, \#succ(J,I)\\
manmax'(J,Y) &\leftarrow manmax'(I,X), mancost'(J,Y), X < Y, \#succ(J,I)\\
manregret'(Z) &\leftarrow manmax'(1,Z)\\
womanmax'(p,X) &\leftarrow womancost'(p,X)\\
womanmax'(J,X) &\leftarrow womanmax'(I,X), womancost'(J,Y), X \geqslant Y, \#succ(J,I)\\
womanmax'(J,Y) &\leftarrow womanmax'(I,X), womancost'(J,Y), X < Y, \#succ(J,I)\\
womanregret'(Z) &\leftarrow womanmax'(1,Z)
\end{aligned} \tag{21}$$

Finally, we define the ASP program $\mathscr{P}_{singles}$ as $\mathscr{P}_{sexeq}$ in which the predicates *sexeq* and *sexeq*′ are resp. replaced by *singles* and *singles*′. Furthermore, we replace rules (6)–(10) by (22), determining the number of singles:

$$singles(Z) \leftarrow \#count\{B : accept(B,B)\} = Z, \#int(Z) \tag{22}$$

This time we adjust rule (22) for the program part $\mathscr{P}_{ext}^{'singles}$ as follows:

$$\begin{aligned}
single'(p+i,1) &\leftarrow accept'(m_i,m_i), \qquad single'(p+i,0) \leftarrow naccept'(m_i,m_i)\\
single'(j,1) &\leftarrow accept'(w_j,w_j), \qquad single'(j,0) \leftarrow naccept'(w_j,w_j)\\
singlesum'(n+p,X) &\leftarrow single'(n+p,X)\\
singlesum'(J,Z) &\leftarrow singlesum'(I,X), single'(J,Y), Z = X+Y, \#succ(J,I)\\
singles'(Z) &\leftarrow singlesum'(1,Z)
\end{aligned} \tag{23}$$

Note that, for $k = \max(n,p)$, the number of grounded rules in the induced ASP program is $\mathcal{O}(k^3)$ for minimum regret and maximum cardinality, but $\mathcal{O}(k^4)$ for sex-equalness and egalitarity. The latter programs have a higher number of grounded rules because of how the weights are counted in the first and second program part. We illustrate our method with an example.

Example 5

We reconsider Example 3. This SMTI instance had three stable matchings 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 ASP program selecting this minimum regret stable matching is the program consisting of the rules in Example 3 in addition to:

https://doi.org/10.1017/S147106841600003X Published online by Cambridge University Press

$$man(m_1) \leftarrow, \quad man(m_2) \leftarrow,$$

$$woman(w_1) \leftarrow, \quad woman(w_2) \leftarrow, \quad woman(w_3) \leftarrow$$

$$\begin{aligned}
mancost(1,1) &\leftarrow accept(m_1,w_1), & womancost(1,1) &\leftarrow accept(m_1,w_1)\\
mancost(1,2) &\leftarrow accept(m_1,w_2), & womancost(1,1) &\leftarrow accept(m_2,w_1)\\
mancost(1,2) &\leftarrow accept(m_1,w_3), & womancost(1,2) &\leftarrow accept(w_1,w_1)\\
mancost(1,4) &\leftarrow accept(m_1,m_1), & womancost(2,1) &\leftarrow accept(m_1,w_2)\\
mancost(2,2) &\leftarrow accept(m_2,w_1), & womancost(2,2) &\leftarrow accept(w_2,w_2)\\
mancost(2,1) &\leftarrow accept(m_2,w_2), & womancost(3,2) &\leftarrow accept(m_1,w_3)\\
mancost(2,2) &\leftarrow accept(m_2,m_2), & womancost(3,1) &\leftarrow accept(m_2,w_3)\\
& & womancost(3,3) &\leftarrow accept(w_3,w_3)
\end{aligned}$$

$$manregret(Z) \leftarrow \#max\{B : mancost(A,B)\} = Z, \#int(Z)$$

$$womanregret(Z) \leftarrow \#max\{B : womancost(A,B)\} = Z, \#int(Z)$$

$$regret(X) \leftarrow manregret(X), womanregret(Y), X > Y$$

$$regret(Y) \leftarrow manregret(X), womanregret(Y), X <= Y$$

$$naccept'(M,W) \vee manpropose'(M,W) \leftarrow man(M), woman(W)$$

$$naccept'(M,W) \vee womanpropose'(M,W) \leftarrow man(M), woman(W)$$

$$accept'(M,W) \vee nmanpropose'(M,W) \vee nwomanpropose'(M,W) \leftarrow man(M), woman(W)$$

$$accept'(m_1,w_1) \vee accept'(m_1,w_2) \vee accept'(m_1,w_3) \vee accept'(m_1,m_1) \leftarrow$$

$$accept'(m_2,w_1) \vee accept'(m_2,w_2) \vee accept'(m_2,m_2) \leftarrow$$

$$naccept'(m_1,m_1) \vee naccept'(m_1,w_1) \leftarrow$$

$$naccept'(m_1,m_1) \vee naccept'(m_1,w_2) \leftarrow$$

$$naccept'(m_1,m_1) \vee naccept'(m_1,w_3) \leftarrow$$

$$naccept'(m_2,m_2) \vee naccept'(m_2,w_1) \leftarrow$$

$$naccept'(m_2,m_2) \vee naccept'(m_2,w_2) \leftarrow$$

$$accept'(m_1,w_1) \vee accept'(m_2,w_1) \vee accept'(w_1,w_1) \leftarrow$$

$$accept'(m_1,w_2) \vee accept'(w_2,w_2) \leftarrow$$

$$accept'(m_1,w_3) \vee accept'(m_2,w_3) \vee accept'(w_3,w_3) \leftarrow$$

$$naccept'(w_1,w_1) \vee naccept'(m_1,w_1) \leftarrow$$

$$naccept'(w_1,w_1) \vee naccept'(m_2,w_1) \leftarrow$$

$$naccept'(w_2,w_2) \vee naccept'(m_1,w_2) \leftarrow$$

$$naccept'(w_3,w_3) \vee naccept'(m_1,w_3) \leftarrow$$

$$naccept'(w_3,w_3) \vee naccept'(m_2,w_3) \leftarrow$$

$$nmanpropose'(m_1,w_2) \vee naccept'(m_1,w_1) \leftarrow$$

$$nmanpropose'(m_1,w_2) \vee naccept'(m_1,w_3) \leftarrow$$

https://doi.org/10.1017/S147106841600003X Published online by Cambridge University Press

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

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

$$mancost'(1,1) \leftarrow accept'(m_1, w_1), \qquad womancost'(1,1) \leftarrow accept'(m_1, w_1)$$
$$mancost'(1,2) \leftarrow accept'(m_1, w_2), \qquad womancost'(1,1) \leftarrow accept'(m_2, w_1)$$
$$mancost'(1,2) \leftarrow accept'(m_1, w_3), \qquad womancost'(1,2) \leftarrow accept'(w_1, w_1)$$
$$mancost'(1,4) \leftarrow accept'(m_1, m_1), \qquad womancost'(2,1) \leftarrow accept'(m_1, w_2)$$
$$mancost'(2,2) \leftarrow accept'(m_2, w_1), \qquad womancost'(2,2) \leftarrow accept'(w_2, w_2)$$
$$mancost'(2,1) \leftarrow accept'(m_2, w_2), \qquad womancost'(3,2) \leftarrow accept'(m_1, w_3)$$
$$mancost'(2,2) \leftarrow accept'(m_2, m_2), \qquad womancost'(3,1) \leftarrow accept'(m_2, w_3)$$
$$womancost'(3,3) \leftarrow accept'(w_3, w_3)$$

$$manmax'(2, X) \leftarrow mancost'(2, X)$$
$$manmax'(J, X) \leftarrow manmax'(I, X), mancost'(J, Y), X >= Y, \#succ(J, I)$$

https://doi.org/10.1017/S147106841600003X Published online by Cambridge University Press

$$
\begin{aligned}
manmax'(J,X) &\leftarrow manmax'(I,X), mancost'(J,Y), X >= Y, \#succ(J,I)\\
manregret'(Z) &\leftarrow manmax'(1,Z)\\
womanmax'(3,X) &\leftarrow womancost'(3,X)\\
womanmax'(J,X) &\leftarrow womanmax'(I,X), womancost'(J,Y), X >= Y,\\
&\quad \#succ(J,I)\\
womanmax'(J,X) &\leftarrow womanmax'(I,X), womancost'(J,Y), X >= Y,\\
&\quad \#succ(J,I)\\
womanregret'(Z) &\leftarrow womanmax'(1,Z)\\
regret'(X) &\leftarrow manregret'(X), womanregret'(Y), X > Y\\
regret'(Y) &\leftarrow manregret'(X), womanregret'(Y), X <= Y\\
sat &\leftarrow regret(X), regret'(Y), X <= Y\\
&\leftarrow not\ sat
\end{aligned}
$$

$$
\begin{aligned}
manargcost'_1(1..2) &\leftarrow, \qquad womanargcost'_1(1..3) \leftarrow\\
manargcost'_2(1..4) &\leftarrow, \qquad womanargcost'_2(1..3) \leftarrow
\end{aligned}
$$

$$
\begin{aligned}
mancost'(X,Y) &\leftarrow sat, manargcost'_1(X), manargcost'_2(Y)\\
womancost'(X,Y) &\leftarrow sat, womanargcost'_1(X), womanargcost'_2(Y)\\
manpropose'(X,Y) &\leftarrow sat, man(X), woman(Y)\\
nmanpropose'(X,Y) &\leftarrow sat, man(X), woman(Y)\\
womanpropose'(X,Y) &\leftarrow sat, man(X), woman(Y)\\
nwomanpropose'(X,Y) &\leftarrow sat, man(X), woman(Y)\\
accept'(X,Y) &\leftarrow sat, man(X), woman(Y)\\
accept'(X,X) &\leftarrow sat, man(X)\\
accept'(X,X) &\leftarrow sat, woman(X)\\
naccept'(X,Y) &\leftarrow sat, man(X), woman(Y)\\
naccept'(X,X) &\leftarrow sat, man(X)\\
naccept'(X,X) &\leftarrow sat, woman(X)
\end{aligned}
$$

Computing the unique answer set of this disjunctive ASP program with DLV 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 matching S_1 of the SMTI instance and the corresponding regret cost.

We prove that there exists a bijective correspondence between the answer sets of the induced disjunctive ASP program and the optimal stable matchings of the SMTI (see the online appendix).

Proposition 2
Let the criterion *crit* be an element of $\{sexeq, weight, regret, singles\}$. For every answer set I of the program $\mathscr{P}_{crit}$ induced by an SMTI instance the set $S_I = \{(m,w) \mid accept(m,w) \in I\}$ forms an optimal stable matching of marriages w.r.t. criterion *crit* and the optimal criterion value is given by the unique value v_I for which $crit(v_I) \in I$.

https://doi.org/10.1017/S147106841600003X Published online by Cambridge University Press

Conversely for every optimal stable matching $S = \{(x_1, y_1), \ldots, (x_k, y_k)\}$ with optimal criterion value v there exists an answer set I of $\mathscr{P}_{crit}$ 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 $crit(v) \in I$.

Remark 2
If we remove from $\mathscr{P}_{sexeq}$ the rules (9)–(10) and replace rule (13) by the rule $sat \leftarrow manweight(X), manweight'(Y), X \leqslant Y$, then we obtain the M-optimal stable matchings. Analogously we can obtain the W-optimal stable matchings.

If a criterion is to be maximized, the symbol $\leqslant$ in rule (13) is simply replaced by $\geqslant$. E.g. for $crit = singles$ we will get minimum cardinality stable matchings.

5 Conclusion

We have shown how ASP programs can be used to encode a number of variations and generalizations of the SMP. Apart from the availability of efficient ASP solvers, the main advantage of our approach is its flexibility, allowing us to find solutions for a wide range of stable matching problems. We can, for instance, compute stable matchings of variants such as the three-dimensional stable matching problem, as well as select stable matchings based on optimality criteria, even for problems with unacceptable partners and ties. We have illustrated our method for sex-equality, egalitarity, minimum regret and maximum cardinality, but the approach can readily be adapted to other optimality criteria (e.g. popular matchings) or to different matching problems (e.g. the roommate problem). To the best of our knowledge, no other exact algorithms exist to find an optimal stable matching for an SMP instance with ties, regardless of the presence of unacceptability and regardless of whether the optimality notion is sex-equality, egalitarity, minimum regret or maximum cardinality. Therefore, our encoding offers the first exact implementation for solving the aforementioned problems.

Supplementary materials

For supplementary material for this article, please visit http://dx.doi.org/10.1017/S147106841600003X

References

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

Biró, P. and McDermid, E. 2010. Three-sided stable matchings with cyclic preferences. *Algorithmica 58,* 1, 5–18.

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

De Clercq, S., Schockaert, S., De Cock, M. and Nowé, A. 2013. Modeling stable matching problems with answer set programming. In *Proc. RuleML 2013*. L. Morgenstern, P. Stefaneas, F. Lévy, A. Wyner and A. Paschke, Eds. Lecture Notes in Computer Science, Springer Berlin Heidelberg, Berlin, Heidelberg, vol. 8035. 68–83.

https://doi.org/10.1017/S147106841600003X Published online by Cambridge University Press

Dung, P. 1995. An argumentation-theoretic foundation for logic programming. *The Journal of Logic Programming 22,* 2, 151–177.

Eiter, T. and Gottlob, G. 1995. On the computational cost of disjunctive logic programming: Propositional case. *Annals of Mathematics and Artificial Intelligence 15,* 3-4, 289–323.

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

Faber, W., Pfeifer, G., Leone, N., Dell'Armi, T. and Ielpa, G. 2008. Design and implementation of aggregate functions in the DLV system. *Theory and Practice of Logic Programming 8,* 5-6, 545–580.

Gale, D. and Shapley, L. 1962. College admissions and the stability of marriage. *The American Mathematical Monthly 69,* 1, 9–15.

Gale, D. and Sotomayor, M. 1985. Some remarks on the stable matching problem. *Discrete Applied Mathematics 11*, 223–232.

Gärdenfors, P. 1975. Match making: Assignments based on bilateral preferences. *Behavioral Science 20,* 3, 166–173.

Gelfond, M. and Lifschitz, V. 1988. The stable model semantics for logic programming. In *Proc. ICLP/SLP*, R. Kowalski and K. Bowen, Eds. MIT Press, Cambridge, vol. 88. 1070–1080.

Gusfield, D. 1987. Three fast algorithms for four problems in stable marriage. *SIAM Journal on Computing 16,* 1, 111–128.

Irving, R. 1994. Stable marriage and indifference. *Discrete Applied Mathematics 48,* 3, 261–272.

Irving, R. 2003. Greedy matchings. Tech. Rep. TR-2003-136, Dept. of Computing Science, University of Glasgow. April.

Irving, R. 2007. The cycle roommates problem: A hard case of kidney exchange. *Information Processing Letters 103,* 1, 1–4.

Irving, R., Leather, P. and Gusfield, D. 1987. An efficient algorithm for the "optimal" stable marriage. *Journal of the ACM 34,* 3, 532–543.

Iwama, K. and Miyazaki, S. 2008. A survey of the stable marriage problem and its variants. In *Proc. of the Interntional Conference on Informatics Education and Research for Knowledge-Circulating Society*. S. Kurohashi, Y. Sumi, Y. Nakamura and K. Tanaka, Eds. ICKS'08. IEEE Computer Society, IEEE Computer Society Washington, DC, USA, 131–136.

Iwama, K., Miyazaki, S. and Yanagisawa, H. 2010. Approximation algorithms for the sex-equal stable marriage problem. *ACM Transactions on Algorithms (TALG) 7,* 1, 2.

Janhunen, T. 2004. Representing normal programs with clauses. In *Proc. of the 16th European Conference on Artificial Intelligence*. R. López De Mántaras, and L. Saitta, Eds. IOS Press, Amsterdam, 358–362.

Knuth, D. 1976. *Mariages Stables.* Les Presses de L'Université de Montréal, Montréal.

Manlove, D. 2013. *Algorithmics of Matching under Preferences.* Series on Theoretical Computer Science, vol. 2. World Scientific Publishing, Singapore.

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

Marek, V., Nerode, A. and Remmel, J. 1990. A theory of nonmonotonic rule systems I. *Annals of Mathematics and Artificial Intelligence 1*, 241–273.

McDermid, E. and Irving, R. 2014. Sex-equal stable matchings: Complexity and exact algorithms. *Algorithmica 68,* 3, 545–570.

Ng, C. and Hirschberg, D. 1991. Three-dimensional stable matching problems. *SIAM Journal on Discrete Mathematics 4*, 245–252.

https://doi.org/10.1017/S147106841600003X Published online by Cambridge University Press

Roth, A., Sömnez, T. and Ünver, M. 2005. Pairwise kidney exchange. *Journal of Economic Theory 125,* 2, 151–188.

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

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

https://doi.org/10.1017/S147106841600003X Published online by Cambridge University Press