Stable Matching Problems with Soft Constraints

(Extended Abstract)

Maria Silvia Pini
University of Padova, Italy
p.ini@dei.unipd.it

Francesca Rossi
University of Padova
Padova, Italy
frossi@math.unipd.it

Kristen Brent Venable
Tulane University & IHMC,
USA
kvenabl@tulane.edu

ABSTRACT

We consider stable matching problems where the agents express their preferences compactly via soft constraints. We study the impact of this choice on the computational complexity of finding stable matching, with particular attention to fuzzy constraints.

Categories and Subject Descriptors
I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence—Multiagent systems; E.2 [Theory of Computation]: Analysis of Algorithms and Problem Complexity

General Terms
Theory, Algorithms

Keywords
Stable Matching, Preferences, Soft Constraints

1. INTRODUCTION

The stable matching (SM) problem is a well-known problem with many practical applications. It has to do with two sets of agents, often called men and women, that should be matched in such a way that no man and woman, who are not married to each other, both prefer each other to their current partner [5]. This property is called stability. Problems of this kind arise in many real-life situations, such as assigning junior doctors to hospitals, children to schools, students to campus housing, kidney transplant patients to donors, and so on. The most well-known and used algorithm to find a stable matching is the GS algorithm [4], that runs in polynomial time. This algorithm assumes that both men and women express their preferences over all members of the opposite gender. However, this can be unfeasible, since the number of men and women can be very large. In addition, eliciting the preferences may be a costly and time-consuming process. Fortunately, the sets of men and women may have a combinatorial structure, which allows for expressing preferences in a compact way by referring to features rather than entire men or women.

Our challenge is to understand how to adapt the GS algorithm to work with such preference statements over features, expressed via soft constraints, and to study the impact of this approach over the computational properties of the algorithm. The main operations performed by the GS algorithm are the following ones: men need to exploit their preferences over women to find their most preferred woman, and possibly also their next most preferred woman (several times), while women need to compare two men according to their preferences over men. We study what it means to perform such operations in a soft constraint setting. Moreover, soft constraints induce a total order over men and women, possibly with ties. The GS algorithm requires a strict total order (that is, no ties) over men and women. We consider three ways to break ties, with particular attention to fuzzy constraints.

A similar study was done using CP-nets instead of soft constraints [6]. However, CP-nets [1] have very different properties from soft constraints, both in terms of expressiveness and of computational complexity of reasoning with them.

2. STABLE MATCHING PROBLEMS

The stable marriage problem (SM) [5] is a well-known problem of matching \( n \) men to \( n \) women to achieve a certain type of ‘stability’. Given \( n \) men and \( n \) women, where each person expresses a strict total ordering over the members of the opposite sex, the problem is to match the men to the women such that no two people of the opposite sex, who are not married to each other, both prefer each other to their current partners. If there are no such pairs, called blocking pairs, every marriage is stable.

In [4] Gale and Shapley (GS) provided an \( O(n^2) \) time algorithm for finding a stable marriage. The GS algorithm consists of a number of rounds in which each un-engaged man proposes to his most preferred woman to whom he has not yet proposed. Each woman receiving a proposal becomes “engaged”, provisionally accepting the proposal from her most preferred man. In subsequent rounds, an already engaged woman can “trade up”, becoming engaged to a more preferred man and rejecting a previous proposal, or, if she prefers him, she can stick with her current partner. Given a matching \( M \), we will denote with \( M(w) \) (resp., \( M(m) \)) the man (resp., woman) associated to the woman \( w \) (resp., man \( m \)) in \( M \). Also, \( \text{pref}(x) \) denotes the preference list of a man or a woman \( x \). The GS algorithm includes the following operations: \( \text{Opt}(\text{pref}(m)) \): Computes the optimal woman for \( m \) (i.e., \( m \)'s first proposal); \( \text{Next}(\text{pref}(m), w) \): computes the next best woman after \( w \) for man \( m \) (i.e., a new proposal for \( m \)); \( \text{Compare}(\text{pref}(w), m, m') \): returns true if woman \( w \) prefers man \( m \) to \( m' \). This is needed when woman \( w \), currently matched with \( m' \), must decide whether to accept or decline a proposal from \( m \).

3. SOFT CONSTRAINTS

A soft constraint [7] involves a set of variables and associates a value from a (partially ordered) set to each instantiation of its variables. Such a value is taken from a c-semiring which is defined...
by \(\{A, +, \times, 0, 1\}\), where \(A\) is the set of preference values, \(+\) induces an ordering over \(A\) (where \(a \leq b\) if \(a + b = b\)), \(\times\) is used to combine preference values, and 0 and 1 are respectively the worst and best element. A Soft Constraint Satisfaction Problem (SCSP) is a tuple \((V, D, C, A)\) where \(V\) is a set of variables, \(D\) is the domain of the variables, \(C\) is a set of soft constraints (each one involving a subset of \(V\)), and \(A\) is the set of preference values. An instance of the SCSP framework is obtained by choosing a specific c-semiring. Choosing \(S_{\text{FCSP}} = \{0, 1\}\), \(\max\), and \(\min\) means that preferences are in \([0, 1]\) and we want to maximize the minimum preference. This is the setting of fuzzy CSPs (FCSPs) that we consider in the paper.

An optimal solution of an SCSP is a complete assignment with an undominated preference. Finding an optimal solution is an NP-hard problem, unless certain restrictions are imposed, such as a tree-shaped constraint graph. Constraint propagation may help the search for an optimal solution. Given a variable ordering \(\sigma\), an FCSP is direction-arc-consistent (DAC) if, for any two variables \(x\) and \(y\) linked by a fuzzy constraint, such that \(x\) precedes \(y\) in the ordering \(\sigma\), we have that, for each \(a\) in the domain of \(x\), \(f_x(a) = \max_{b \in D(y)}(\min\{f_x(a), f_y(a, b), f_y(b)\})\), where \(f_x\), \(f_y\), and \(f_{xy}\) are the preference functions of \(c_x\), \(c_y\), and \(c_{xy}\). This definition can be generalized to any instance of the SCSP approach by replacing \(\max\) with \(+\) and \(\min\) with \(\times\). DAC is enough to find the preference level of an optimal solution when the problem has a tree-shaped constraint graph and the variable ordering is compatible with the father-child relation of the tree [7], since the optimum preference level is the best preference level in the domain of the root variable.

4. STABLE MATCHING WITH SOFT CONSTRAINTS

We consider a stable marriage problem with \(n\) men and \(n\) women, where each man and each woman specify their preferences over the members of the other gender via a set of soft constraints. Each man and woman is described by a set of features, that are represented by the variables of the soft constraint problems. If each variable has \(d\) possible values, the number of variables, say \(f\), of each soft constraint problem is \(O(log_d n)\).

GS operations. In the GS algorithm, men make proposals, starting from their most preferred woman and going down in their ordering, while women receive proposals and compare these against the man to whom they are currently engaged. We will now model the GS operations.

\(\text{Opt}(\text{pref}(m))\) must return the optimal solution of an SCSP defining the preferences of man \(m\) over the women. In general, finding the optimal solution of an SCSP is a computationally difficult problem. However, if the SCSP has a tree-like shape, or a bounded tree-width, it can be done in polynomial time [7].

\(\text{Compare}(\text{pref}(w), m_1, m_2)\) compares two complete assignments \(m_1\) and \(m_2\) and check if \(m_1\) is strictly more preferred to \(m_2\). In SCSPs, this is computationally easy to do, if the combination operator is polynomially computable and there is a polynomial number of constraints. In fact, \(m_1\) is strictly preferred to \(m_2\) when the preference of \(m_1\) for \(w\) is strictly greater than that of \(m_2\) for \(w\). Notice that women need only to perform \(\text{Compare}\) operations. Thus we do not need any restriction on the shape of the constraint graph for women’s preferences to make \(\text{Compare}\) polynomial.

For the \(\text{Next}(\text{pref}(m), w)\) operation, we need to understand how to linearize the solution ordering of an SCSP. In fact, this operation is used to find the next most preferred woman in a man’s preference ordering, so when two or more women are tied, we need to put an order over them to understand who to propose first.