Brief Announcement: Applying Predicate Detection to the Stable Marriage Problem*

Vijay K. Garg[1]

1 **The University of Texas at Austin,**
Austin, TX 78712, garg@ece.utexas.edu

Abstract

We show that many techniques developed in the context of predicate detection are applicable to the stable marriage problem. The standard Gale-Shapley algorithm can be derived as a special case of detecting linear predicates. We also show that techniques in computation slicing can be used to represent the set of all constrained stable matchings.

1998 ACM Subject Classification C.2.4 Distributed Systems http://www.acm.org/about/class/ccs98-html

Keywords and phrases Stable Matching Linear Predicates Distributive Lattices

Digital Object Identifier 10.4230/LIPIcs.DISC.2017.52

1 Introduction

The Stable Matching Problem (SMP) [2] has wide applications in economics, distributed computing, and resource allocation. In this paper, we show that techniques for detecting global predicates can be used to derive solutions to a more general problem than SMP, called *constrained SMP*. In our formulation, in addition to men and women preferences, there may be a set of constraints on the set of marriages consistent with men's preferences. For example, we may state that Peter's regret [4] should be less than that of Paul where the *regret* of a man in a matching is the choice number he is assigned.

To solve a constrained SMP, we define a distributed computation such that every assignment of women to men corresponds to a global state of the distributed computation. The set of global states form a finite distributive lattice under the natural order on the global states. The problem of finding a stable matching reduces to that of finding a consistent global state that satisfies the boolean predicate B for the constrained stable matching. We show that B satisfies the linearity property introduced in predicate detection [1]. Consequently, we can use the algorithm that can detect a linear predicate in [1] to find a constrained stable matching. The stable matching found by this algorithm is man-optimal [2]. The Gale-Shapley algorithm is a special case of the linear predicate detection algorithm when the set of external constraints is null. While there always exists a stable matching for the SMP problem, there may not exist a stable matching in the constrained SMP. Our algorithm is guaranteed to find one whenever it exists. In addition, the stable matching it finds is man-optimal.

We then consider the constrained SMP in a distributed setting where men and women know only those constraints that either they choose (such as their preference lists) or the external constraints in which they participate. We present a distributed algorithm based on diffusing computation that solves the constrained SMP. When external constraints are null,

* This work was partially supported by NSF CSR-1563544.

the algorithm reduces to a distributed version of the Gale-Shapley Algorithm [2]. To our knowledge, this paper is the first one to propose sequential and distributed algorithms for constrained SMP.

In the full version of the paper [3], we also consider the problem of computing all constrained stable matchings. Since the number of stable matchings may be exponential in the number of men, instead of keeping all matchings in an explicit form, we would like a concise representation of polynomial size that can be used to enumerate all constrained stable matchings. In SMP literature, rotation posets are used to capture all stable matchings. We use the notion of computation slicing introduced in [5] for this purpose. In particular, we give an efficient algorithm to compute the slice for the constrained SMP computation. A rotation poset [4] is a special case of the slice when the set of external constraints is empty.

2 Modeling Stable Matching Problem (SMP) as a Distributed Computation

Let E be the set of proposals made by men to women. We also call these proposals *events* which are executed by n processes corresponding to n men denoted by $\{P_1 \ldots P_n\}$. Each of the events can be characterized by a tuple (i, j) that corresponds to the proposal made by man i to woman j. We impose a partial order $\rightarrow_p$ on this set of events to model the order in which these proposals can be made. In the standard SMP, every man P_i has its preference list mpref$[i]$ such that mpref$[i][k]$ gives the k^{th} most preferred woman for P_i. We model mpref using $\rightarrow_p$; if P_i prefers woman j to woman k, then there is an edge from the event (i, j) to the event (i, k). As in SMP, we assume that every man gives a total order on all women.

In the standard stable matching problem, there are no constraints on the order of proposals made by different men, and $\rightarrow_p$ can be visualized as a partial order $(E, \rightarrow_p)$ with n disjoint chains. We generalize the SMP problem to include external constraints on the set of proposals. In the constrained SMP problem, $\rightarrow_p$ can relate proposals made by different men and therefore $\rightarrow_p$ forms a general poset $(E, \rightarrow_p)$.

A global state $G \subseteq E$ is simply the subset of events executed in the computation such that it preserves the order of events within each P_i. A global state G is *consistent* if it preserves the $\rightarrow_p$ order. We will deal only with consistent global states from now on. We let $G[i]$ denote the last proposal made by P_i. Initially, $G[i]$ is null for all men. If P_i has made $k > 0$ proposals, then mpref$[i][k]$ gives the identity of the woman last proposed by P_i. We model women preferences using edges on the computation graph as follows. If an event e corresponds to a proposal by P_i to woman q and she prefers P_j, then we add a $\rightarrow_w$ edge from e to the event f that corresponds to P_j proposing to woman q. The set E along with $\rightarrow_w$ edges also forms a partial order $(E, \rightarrow_w)$ where $e \rightarrow_w f$ iff both proposals are to the same woman and that woman prefers the proposal f to e.

The above discussion motivates the following definition.

▶ **Definition 1** (Constrained SMP Graph). Let $E = \{(i, j) | i \in [1..n] \text{ and } j \in [1..n]\}$. A Constrained SMP Graph $((E, \rightarrow_p), \rightarrow_w)$ is a directed graph on E with two sets of edges $\rightarrow_p$ and $\rightarrow_w$ with the following properties: (1) $(E, \rightarrow_p)$ is a poset such that the set $P_i = \{(i, j) | j \in [1..n]\}$ is a chain for all i, and (2) $(E, \rightarrow_w)$ is a poset such that the set $Q_j = \{(i, j) | i \in [1..n]\}$ is a chain for all j and there is no $\rightarrow_w$ edge between proposals to different women, i.e., for all $i, j, k, l : (i, j) \rightarrow_w (k, l) \Rightarrow (j = l)$.

We define the *frontier* of a global state G as the set of maximal events executed by any process in G. It includes only the last event executed by P_i (if any). We call the events in G

that are not in $frontier(G)$ as pre-frontier events. A consistent global state G is *admissible* if $\forall e, f \in frontier(G) : \forall g \in G : (e \rightarrow_w g) \Rightarrow \neg((g \rightarrow_p f) \vee (g = f))$. We now have the following lemma.

▶ **Lemma 2.** *Let* $((E, \rightarrow_p), \rightarrow_w)$ *be a constrained SMP graph. A consistent global state* G *such that* $G \cap P_i \neq \emptyset$ *for all* i *is admissible iff the assignment by* G *corresponds to a constrained stable matching.*

Therefore, the problem of finding a stable matching is the same as finding a consistent global state that satisfies the predicate *admissible* which is defined purely in graph-theoretic terms on the constrained SMP graph.

We now show that admissibility satisfies linearity introduced in [1]. Any linear predicate can be detected efficiently. A key concept in deriving an efficient predicate detection algorithm is that of a *forbidden* state. Let L be the lattice of all global states of a poset $(E, \rightarrow_p)$. Given a predicate B, and a global state $G \in L$, a state $G[i]$ is called forbidden if its inclusion in any global state $H \in L$, where $G \subseteq H$, implies that B is false for H. Formally, $\mathrm{forbidden}(G, i, B) \equiv \forall H \in L : G \subseteq H : (G[i] \neq H[i]) \vee \neg B(H)$.

A predicate B is linear with respect to the poset $(E, \rightarrow_p)$ if for any global state G in the poset, B is false in G implies that G contains a *forbidden state*. Formally, a boolean predicate B is *linear* with respect to a poset $(E, \rightarrow_p)$ iff $\forall G \in L : \neg B(G) \Rightarrow \exists i : \mathrm{forbidden}(G, i, B)$. We now have

▶ **Lemma 3.** *For any global state* G *that is not a constrained stable matching, there exists an* i *such that* $forbidden(G, i, admissible)$*.*

We now discuss detection of linear global predicates. On account of linearity of B, if B is evaluated to be false in some global state G, then we can determine i such that $forbidden(G, i, B)$. We can then simply advance on any i such that $forbidden(G, i, B)$ holds.

We now consider the constrained SMP in a distributed system setting. We assume that each man and woman knows only his or her preference lists. In addition, each man is given a list of prerequisite proposals for each of the women that he can propose to. In terms of the constrained-SMP graph, this corresponds to every man knowing the incoming $\rightarrow_p$ edges for the chain that corresponds to that man in the graph. The full paper [3] presents a diffusing computation for solving the constrained SMP problem.

Acknowledgements. I thank Rohan Garg for many discussions on this topic and anonymous reviewers for their comments.

References

1. Craig M Chase and Vijay K Garg. Detection of global predicates: Techniques and their limitations. *Distributed Computing*, 11(4):191–201, 1998.
2. David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.
3. Vijay K. Garg. Applying predicate detection to the stable marriage problem. Technical Report TR-PDS-2017-001, The University of Texas at Austin, May 2017. http://users.ece.utexas.edu/~garg/TechReports/2017/TR-PDS-2017-001.pdf.
4. Dan Gusfield and Robert W Irving. *The stable marriage problem: structure and algorithms.* MIT press, 1989.
5. Neeraj Mittal and Vijay K Garg. Computation slicing: Techniques and theory. In *International Symposium on Distributed Computing*, pages 78–92. Springer, 2001.