

CORE MANY-TO-ONE MATCHINGS BY FIXED-POINT METHODS¹

FEDERICO ECHENIQUE

Humanities and Social Sciences MC 228-77
California Institute of Technology
Pasadena CA 91125, USA
fede@hss.caltech.edu

JORGE OVIEDO

Instituto de Matemática Aplicada
Universidad Nacional de San Luis
Ejército de los Andes 950
5700 San Luis, Argentina.
joviedo@unsl.edu.ar

Abstract

We characterize the core many-to-one matchings as fixed points of a map. Our characterization gives an algorithm for finding core allocations; the algorithm is efficient and simple to implement. Our characterization does not require substitutable preferences, so it is separate from the structure needed for the non-emptiness of the core. When preferences are substitutable, our characterization gives a simple proof of the lattice structure of core matchings, and it gives a method for computing the join and meet of two core matchings.

Journal of Economic Literature Classification Number: C78

Keywords: Matching, Core, Lattice, Stability, Algorithm, Complexity, Substitutability, Tarski's Fixed Point Theorem.

Running title: "Core Matchings"

¹We are grateful to Jimpeng Ma, Ruth Martínez, and Alejandro Neme for their helpful comments.

1 Introduction

This paper deals with the many-to-one matchings that are in the core. Given a set of workers and a set of firms, a many-to-one matching (a matching, for short) is an assignment of a group of workers to each firm. The problem is interesting because each worker has a preference relation over firms, and each firm has a preference relation over groups of workers. Given a matching, firms who are unhappy with their current group of workers, and workers who are unhappy with their current employers, may recontract in a mutually beneficial way, thus destroying the proposed matching. Arguably, the natural formalization of matchings that are robust to recontracting is the core (see Roth and Sotomayor [12], RS in the rest of the paper, for definitions).

The standard approach to the core (RS is a classic) is to impose some structure on firms' preferences that both simplifies the core and ensures that it is non-empty; the structure used is to assume that firms' preferences are "substitutable" (see RS, or our Definition 5 below). When preferences are substitutable, a matching is in the core if and only if it is "stable"—a stable matching is robust to recontracting by worker-firm pairs, but not more complicated coalitions. Further, there is an algorithm, the "deferred acceptance algorithm" (DAA) which finds certain distinguished core matchings: the DAA finds a core matching that is better for the firms than all other core matchings, and a core matching that is better for the workers than all other core matchings. Kelso and Crawford [7] introduced the notion of substitutability (their "gross substitutes" condition) and first used it to prove—via the DAA—that the core is non-empty in a class of models that, under substitutability, includes the model studied in our paper.

Our approach is different. We characterize the core as the set of fixed points of a map T . We do not add structure to the model, so the core (and the fixed points of T) may be empty. Our characterization is useful because it allows us to:

1. Construct a simple algorithm that consists in iterating T , we shall call it the T -algorithm. The T -algorithm either finds a core matching or cycles in a recognizable way. Unlike the DAA, the T -algorithm does not stop at a non-core matching. We present an example where the DAA stops at a non-core matching, while the T -algorithm succeeds in finding a core matching.
2. We prove that, when preferences are substitutable, T has at least one fixed point—hence that the core is non-empty—and that the T -algorithm finds the two distinguished core matchings: the matching that is best for the firms and the matching that is best for the workers.

We can bound the computational complexity of the T -algorithm. To the best of our knowledge, the computational complexity of the DAA for the many-to-one

case is unknown. ¹

3. Blair [3] proves that, when preferences are substitutable, the core matchings have a lattice structure.
 - (a) Our characterization gives a very simple proof of Blair’s result; it follows almost immediately from Tarski’s fixed point theorem, as T is a monotone map under an appropriate order.
 - (b) There is no formula for computing Blair’s lattice operations (join and meet). We show how the T -algorithm *computes* the join and meet of any two core matchings.

Our results on the core build a bridge to the literature on supermodular games (see [15]); in fact, our results on the T -algorithm are adapted from similar results on computing Nash equilibria in supermodular games. We believe that this bridge is useful for understanding matching problems better, and may deliver further results in future research.

Fixed-point methods have been used in the matching literature before. Tarski’s fixed-point theorem was applied by Roth and Sotomayor [11] to the assignment game, Adachi [1] to one-to-one stable matchings, and Fleiner [5] to many-to-one stable matchings. Fleiner’s is the paper closest to our paper; he proves a version of Blair’s lattice result for a general graph-theoretic model. Fleiner’s model includes the many-to-one matching model with substitutable preferences as a special case. Fleiner’s paper is a paper in graph theory, though, and his results are very hard to translate into the language of matching models. For that reason, we believe that our result in (3.a) above is valuable, beyond its auxiliary role in our other results.

We should clarify at this point that “core” is RS’s “weak core.” We do not have results for RS’s “strong core,” hence we opted, in the interest of exposition economy, for omitting all “weak” qualifiers from the paper. Nothing is really known about the strong core, so we feel that our omission is justified. See RS, and Sotomayor [13], for a discussion of the different core definitions.

In Section 2 we define the model, and prove some preliminary results on the core many-to-one matchings. In Section 3 we introduce our fixed-point approach to the core. In Section 4 we introduce the T -algorithm. In Section 5 we study the lattice structure of the core many-to-one matchings. In Section 6 we analyze the T -algorithm for substitutable preferences.

¹Gusfield and Irving [6] prove that the DAA is $O(n^2)$ for one-to-one matchings.

2 Preliminary definitions and results

2.1 The Model

There are two disjoint sets of agents, the set of n **firms** F and the set of m **workers** W . Each firm $f \in F$ has a strict, transitive, and complete preference relation $P(f)$ over the set of all subsets of W , and each worker has a strict, transitive, and complete preference relation $P(w)$ over $F \cup \emptyset$. **Preferences profiles** are $(n + m)$ -tuples of preference relations; we denote them by $P = (P(f_1), \dots, P(f_n); P(w_1), \dots, P(w_m))$.

Given a firm's preference relation, $P(f)$, the sets of workers that f prefers to the empty set are called acceptable; thus we allow that firm f may prefer not hiring any worker rather than hiring unacceptable subsets of workers. Similarly, given a preference relation of a worker $P(w)$ the firms preferred by w to the empty set are called acceptable; in this case we are allowing that worker w may prefer to remain unemployed rather than working for an unacceptable firm. It turns out that only acceptable partners matter, so we shall write preference relation concisely as lists of acceptable partners. For example, $P(f_i) = \{w_1, w_3\}, \{w_2\}, \{w_1\}, \{w_3\}$ indicates that $\{w_1, w_3\} P(f_i) \{w_2\} P(f_i) \{w_1\} P(f_i) \{w_3\} P(f_i) \emptyset$. And $P(w_j) = f_1, f_3$ indicates that $f_1 P(w_j) f_3 P(w_j) \emptyset$.

We denote by R the weak orders associated with P . So, $f_i R(w) f_j$ if $f_i = f_j$ or $f_i P(w) f_j$. Similarly for $R(f)$.

We shall study assignments of workers to firms as matchings of firms with groups of workers. The assignment problem consists of matching workers with firms maintaining the bilateral nature of their relationship, and allowing for the possibility that both firms and workers may remain unmatched. Formally,

Definition 1 A **matching** μ is a mapping from the set $F \cup W$ into the set of all subsets of $F \cup W$ such that for all $w \in W$ and $f \in F$: a) Either $|\mu(w)| = 1$ and $\mu(w) \subseteq F$ or else $\mu(w) = \emptyset$. b) $\mu(f) \in 2^W$. c) $\mu(w) = \{f\}$ ² if and only if $w \in \mu(f)$.

Given (F, W, P) we denote the set of matchings by \mathcal{M} .

Let P be a preference profile. Given a set $S \subseteq W$, let $Ch(S, P(f))$ denote firm f 's most-preferred subset of S according to its preference ordering $P(f)$. We call $Ch(S, P(f))$ the **choice set** of S according to $P(f)$. That is, $A = Ch(S, P(f))$ if and only if $A \subseteq S$ and $A P(f) B$ for all $B \subseteq S$ with $A \neq B$.

²We shall often abuse notation by omitting the brackets to denote a set with a unique element; here we write $\mu(w) = f$ instead of $\mu(w) = \{f\}$.

2.2 The core, stable matchings, and stable* matchings

A matching μ is stable if no worker-firm pair can benefit by deviating from μ . RS prove that, if firms' preferences are substitutable, the set of stable matchings equals the core.

A matching is stable* if no *set* of workers can, jointly with one firm, benefit from deviating from μ . We prove that the set of stable* matchings equals the core; the result does not require additional assumptions on preferences. We use the notion of stable* matching instead of stable matching because our fixed-point approach allows us to characterize the set of stable* matchings, and thus the core. It should be clear from our proofs and examples just how useful the stable* definition is.

Let P be a preference profile and let μ be a matching. Say that μ is **individually rational** if $\mu(w)R(w)\emptyset$ for all $w \in W$, and if $\mu(f) = Ch(\mu(f), P(f))$, for all $f \in F$. Intuitively, μ is individually rational if no agent can unilaterally improve over its assignment by μ —workers by choosing to remain unemployed and firms by firing some of its workers.

A worker-firm pair (w, f) **blocks** μ if $w \notin \mu(f)$, $w \in Ch(\mu(f) \cup \{w\}, P(f))$, and $fP(w)\mu(w)$; that is, if w and f are not matched through μ , firm f wants to hire w —possibly after firing some of its current workers in $\mu(f)$ —and worker w prefers firm f over her current match $\mu(w)$.

Let P be a preference profile and let μ be a matching. A pair $(B, f) \in 2^W \times F$, with $B \neq \emptyset$, **blocks*** μ if $fP(w)\mu(w)$, for all $w \in B$, and there is $A \subseteq \mu(f)$ such that $[A \cup B]P(f)\mu(f)$. In words, (B, f) blocks* μ if firm f is willing to hire the workers in B —possibly after firing some of its current workers in $\mu(f)$ —and all workers w in B prefer f over their current match $\mu(w)$.

Definition 2 A matching μ is **stable** if it is individually rational and there is no worker-firm pair that blocks μ . A matching μ is **stable*** if it is individually rational and there is no pair that blocks* μ .

Notation. Given a preference profile P , we denote the set of stable matchings by $S(P)$, and the set of stable* matchings by $S^*(P)$.

Lemma 1 $S^*(P) \subseteq S(P)$

Proof. Let $\mu \in S^*(P)$, and assume that $\mu \notin S(P)$. Since μ is individually rational, and $\mu \notin S(P)$, there exist $(w, f) \in W \times F$ such that $fP(w)\mu(w)$ and $w \in Ch(\mu(f) \cup \{w\}, P(f))$. This implies $Ch(\mu(f) \cup \{w\}, P(f))P(f)\mu(f)$. Let

$B = \{w\}$, and $A = Ch(\mu(f) \cup \{w\}, P(f)) \cap \mu(f)$. Then (B, f) blocks* μ . Thus, $\mu \notin S^*(P)$. \blacksquare

We prove below that $S^*(P)$ equals the core, so it follows from standard results that $S^*(P)$ and $S(P)$ may be different. We present an example as “supplementary material” in <http://www.nyu.edu/jet/supplementary.html> and in the working-paper version ([4]) of the paper.

Definition 3 *Let P be a preference profile. The **core** is the set of matchings μ for which there is no $F' \subseteq F$, $W' \subseteq W$ with $F' \cup W' \neq \emptyset$, and $\hat{\mu} \in \mathcal{M}$ that satisfy a), b) and c): a) For all $w \in W'$ and $f \in F'$ $\hat{\mu}(w) \subseteq F'$, and $\hat{\mu}(f) \subseteq W'$. b) For all $w \in W'$ and $f \in F'$, $\hat{\mu}(w)R(w)\mu(w)$ and $\hat{\mu}(f)R(f)\mu(f)$. c) There is $s \in W' \cup F'$ with $\hat{\mu}(s)P(s)\mu(s)$. We shall denote the core by $C_w(P)$.*

Theorem 2 $S^*(P) = C_w(P)$.

Proof. First we shall prove that $S^*(P) \subseteq C_w(P)$. Let $\mu \in S^*(P)$, and suppose that $\mu \notin C_w(P)$. Let $F' \subseteq F$, $W' \subseteq W$ with $F' \cup W' \neq \emptyset$, and let $\hat{\mu} \in \mathcal{M}$ such that, for all $w \in W'$, and for all $f \in F'$, $\hat{\mu}(w) \subseteq F'$, and $\hat{\mu}(f) \subseteq W'$, $\hat{\mu}(w)R(w)\mu(w)$, $\hat{\mu}(f)R(f)\mu(f)$, and $\hat{\mu}(s)P(s)\mu(s)$ for at least one $s \in W' \cup F'$. We shall need the following

CLAIM. There exists $f \in F'$, such that $\hat{\mu}(f)P(f)\mu(f)$ if and only if there is $w \in W'$ such that $\hat{\mu}(w)P(w)\mu(w)$.

PROOF OF THE CLAIM. Let $\hat{\mu}(f)P(f)\mu(f)$. Because μ is individually rational, we have that $\hat{\mu}(f) \not\subseteq \mu(f)$, so let $\bar{w} \in \hat{\mu}(f) \setminus \mu(f)$. By definition of $\hat{\mu}$ we have that $\bar{w} \in \hat{\mu}(f) \subseteq W'$; so $\bar{w} \notin \mu(f)$ and the definition of $\hat{\mu}$ implies that $\hat{\mu}(\bar{w})P(\bar{w})\mu(\bar{w})$. Now suppose there exists $w \in W'$ such that $\hat{\mu}(w)P(w)\mu(w)$. Let $\bar{f} = \hat{\mu}(w)$. Then $\bar{f} \neq \mu(w)$, so $w \notin \mu(\bar{f})$. Thus $\hat{\mu}(\bar{f}) \neq \mu(\bar{f})$, and the definition of $\hat{\mu}$ implies that $\hat{\mu}(\bar{f})P(\bar{f})\mu(\bar{f})$. This proves the claim.

By the claim, we can assume that there exists $f \in F'$ such that $\hat{\mu}(f) \neq \mu(f)$. Let $B = \hat{\mu}(f) \setminus \mu(f)$ and $A = \hat{\mu}(f) \cap \mu(f)$ then $A \cup B = \hat{\mu}(f)P(f)\mu(f)$. Now, $B \subseteq \hat{\mu}(f) \subseteq W'$, and $B \cap \mu(f) = \emptyset$. Then (B, f) blocks* μ , this contradicts that $\mu \in S^*(P)$.

Now we shall prove that $C_w(P) \subseteq S^*(P)$. Let $\mu \in C_w(P)$, and suppose that $\mu \notin S^*(P)$. First we shall show that μ is individually rational. Assume that there exists $f \in F$ such that $\mu(f) \neq Ch(\mu(f), P(f))$. Let $\hat{\mu}(f) = Ch(\mu(f), P(f))$, $F' = \{f\} \subseteq F$, $W' = \hat{\mu}(f) \subseteq W$. Thus W' , F' , and $\hat{\mu}$ imply $\mu \notin C_w(P)$. Thus, $\mu(f) = Ch(\mu(f), P(f))$, for all $f \in F$.

Now suppose there is $w \in W$ such that $\emptyset P(w) \mu(w)$. Let $\hat{\mu}(w) = \emptyset$, $W' = \{w\}$, $F' = \emptyset$. As before, $\mu \notin C_w(P)$ because $\hat{\mu}(w)P(w)\mu(w)$. Thus, $\mu(w) R(w) \emptyset$, for all $w \in W$. Thus μ is individual rational.

Assume that there exist $(B, f) \in 2^W \times F$ with $B \neq \emptyset$, such that: for all $w \in B$, $fP(w)\mu(w)$, and there exists $A \subseteq \mu(f)$ such that $[A \cup B] P(f) \mu(f)$. Letting $\hat{\mu}(f) = A \cup B$, $F' = \{f\}$ and $W' = B$, it follows that $\mu \notin C_w(P)$. A contradiction. ■

2.3 Lattices and chains

Let X be a set, and R a partial order—a transitive, reflexive, antisymmetric binary relation—on X . For any subset A of X , $\inf_R A$ ($\sup_R A$) is the greatest (lowest) lower (upper) bound on A , in the order R . We say that the pair (X, R) is a **lattice** if whenever $x, y \in X$, both $x \wedge_R y = \inf_R \{x, y\}$ and $x \vee_R y = \sup_R \{x, y\}$ exist in X . We say that a lattice (X, R) is **complete** if for every nonempty subset A of X , $\inf_R A$ and $\sup_R A$ exist in X . Note that any finite lattice is complete. A lattice (X, R) is a **chain** if, for any $x, y \in X$, xRy or yRx , or both, are true.

3 The core as a set of fixed points

We shall construct a map T on a superset of \mathcal{M} such that the set of fixed points of T is the core.

3.1 Pre-matchings

Say that a pair $\nu = (\nu_F, \nu_W)$, with $\nu_F : F \rightarrow 2^W$ and $\nu_W : W \rightarrow F \cup \{\emptyset\}$, is a **pre-matching**. Let \mathcal{V}_W (\mathcal{V}_F) denote the set of all ν_W (ν_F) functions. Thus, $\mathcal{V} = \mathcal{V}_F \times \mathcal{V}_W$ denotes the set of pre-matchings $\nu = (\nu_F, \nu_W)$. The superset of \mathcal{M} we need turns out to be the set of pre-matchings \mathcal{V} (pre-matchings were first used in [1]).

A pre-matching $\nu = (\nu_F, \nu_W)$, and a matching μ are equivalent if $\mu(f) = \nu_F(f)$ for all f , and $\mu(w) = \nu_W(w)$ for all w . A pre-matching $\nu = (\nu_F, \nu_W)$ is equivalent to some matching if and only if ν is such that $\nu_W(w) = f$ if and only if $w \in \nu_F(f)$.

Notation. We make two conventions. First, we shall identify matchings and pre-matchings when they are equivalent, so if $\nu \in \mathcal{V}$ is equivalent to some matching μ we shall say that ν is a matching. Hence \mathcal{M} is a subset of \mathcal{V} . Second, if $\nu = (\nu_F, \nu_W)$ is a pre-matching, we shall some times write $\nu(f)$ for $\nu_F(f)$ and $\nu(w)$ for $\nu_W(w)$.

3.2 T and the core

Let ν be a pre-matching, and let $U(f, \nu) = \{w \in W : fR(w)\nu(w)\}$, and $V(w, \nu) = \{f \in F : w \in Ch(\nu(f) \cup \{w\}), P(f)\} \cup \{\emptyset\}$. The set $U(f, \nu)$ is the set of workers w that are willing to give up $\nu(w)$ in exchange for f . The set $V(w, \mu)$ is the set of firms f that are willing to hire w , possibly after firing some of the workers it was assigned by ν .

Now, define $T : \mathcal{V} \rightarrow \mathcal{V}$ by $(T\nu)(s) = Ch(U(s, \nu), P(s))$ if $s \in F$, and $(T\nu)(s) = \max_{P(s)} \{V(s, \nu)\}$ if $s \in W$. The map T has a simple interpretation: $(T\nu)(f)$ is firm f 's optimal team of workers, among those willing to work for f , and $(T\nu)(w)$ is the firm preferred by w among the firms that are willing to hire w . We shall denote by \mathcal{E} the set of fixed points of T , so $\mathcal{E} = \{\nu \in \mathcal{V} : \nu = T\nu\}$.

Theorem 3 $\mathcal{E} = S^*(P)$.

Before we prove Theorem 3, note that Theorems 2 and 3 imply:

Corollary 4 $\mathcal{E} = C_w(P)$.

Proof of Theorem 3. First we shall prove that if $\nu \in \mathcal{E}$, then ν is a matching and ν is individually rational. Let $\nu = (\nu_F, \nu_W) \in \mathcal{E}$. Fix $w \in \nu_F(f)$, we shall prove that $f = \nu_W(w)$. $\nu \in \mathcal{E}$ implies that $w \in \nu_F(f) = (T\nu)(f) = Ch(U(f, \nu), P(f))$. Thus $w \in U(f, \nu)$.

The definition of $U(f, \nu)$ implies $fR(w)\nu_W(w)$. Now, $\nu_F(f) \cup \{w\} = \nu_F(f)$ and $\nu \in \mathcal{E}$ imply that $\nu_F(f) = (T\nu)(f) = Ch(U(f, \nu), P(f))$ (\star). So

$$\begin{aligned} Ch(\nu_F(f), P(f)) &\stackrel{(1)}{=} Ch(Ch(U(f, \nu), P(f)), P(f)) \\ &\stackrel{(2)}{=} Ch(U(f, \nu), P(f)) \\ &\stackrel{(3)}{=} \nu_F(f). \end{aligned}$$

Equalities (1) and (3) follow from (\star). Equality (2) is a simple property of choice sets. Hence $\nu_F(f) = Ch(\nu_F(f), P(f))$.

Now, $w \in \nu_F(f)$ implies $Ch(\nu_F(f), P(f)) = Ch(\nu_F(f) \cup \{w\}, P(f))$. So we have $f \in V(w, \nu)$. But $\nu_W(w) = (T\nu)(w) = \max_{P(w)} \{V(w, \nu)\}$, so $\nu_W(w)R(w)f$. But $fR(w)\nu_W(w)$, so $f = \nu_W(w)$, as $R(w)$ is antisymmetric.

Let $f = \nu_W(w)$, we shall prove that $w \in \nu_F(f)$. First, note that $f = \nu_W(w)$ implies that $w \in U(f, \nu)$. Second, note that $f = \nu_W(w) = (T\nu)(w) = \max_{P(w)} \{V(w, \nu)\}$, because $\nu \in \mathcal{E}$. So $\nu_W(w)R(w)\emptyset$, and since $\nu_F(f) = Ch(\nu_F(f), P(f))$, ν is individually rational.

Now, $f \in V(w, \nu)$, so we must have $w \in Ch(\nu_F(f) \cup \{w\}, P(f))$ by the definition of $V(w, \nu)$. That is,

$$w \in Ch(\nu_F(f) \cup \{w\}, P(f))R(f)\nu_F(f), \quad (1)$$

Now, $\nu \in \mathcal{E}$ implies that

$$\nu_F(f) = (T\nu)(f) = Ch(U(f, \nu), P(f)). \quad (2)$$

The definition of choice set implies $\nu_F(f) \subseteq U(f, \nu)$, and $\nu_F(f)R(f)U(f, \nu)$. But $w \in U(f, \nu)$ implies $U(f, \nu) \supseteq Ch(\nu_F(f) \cup \{w\}, P(f))$.

The definition of choice set implies $\nu_F(f)R(f)Ch(\nu_F(f) \cup \{w\}, P(f))$. Together with (1), this implies $w \in \nu_F(f)$.

We have shown that $w \in \nu_F(f)$ if and only if $f = v_W(w)$. So, ν is a matching. Also, ν is individually rational.

Let $\mu \in \mathcal{E}$. Fix $f \in F$, $B \subseteq W$ such that $B \neq \emptyset$. We assume that, for all $w \in B$, $fP(w)\mu(w)$. By the definition of $U(f, \mu)$, we have that $B \subseteq U(f, \mu)$. Let $A \subseteq \mu(f)$. Because μ is a matching, we have that for all $w \in A$, $f = \mu(w)$. So the definition of $U(f, \mu)$ implies that

$$A \subseteq U(f, \mu). \quad (3)$$

Now, $\mu \in \mathcal{E}$ so $\mu(f) = (T\mu)(f) = Ch(U(f, \mu), P(f))$; $B \subseteq U(f, \mu)$ and statement (3) imply then

$$\mu(f)R(f)Ch(A \cup B, P(f))R(f)A \cup B. \quad (4)$$

So there is no (B, f) that blocks* μ . Thus $\mu \in S^*(P)$.

Let $\mu \in S^*(P)$ and assume that $\mu \neq T\mu$. First assume that there exist $f \in F$ such that

$$\mu(f) \neq (T\mu)(f) = Ch(U(f, \mu), P(f)) = C \subseteq U(f, \mu).$$

Let $A = C \cap \mu(f)$, and $B = C \setminus \mu(f)$. Because μ is an individually rational matching we have that $\mu(f) \subseteq U(f, \mu)$ and $B \neq \emptyset$. Then (B, f) blocks* μ , which contradicts that $\mu \in S^*(P)$.

Hence, for all $f \in F$,

$$\mu(f) = (T\mu)(f). \quad (5)$$

Assume that there exists $w \in W$ such that

$$\mu(w) \neq (T\mu)(w) = \max_{P(w)} \{V(w, \mu)\} = \bar{f} \in V(w, \mu),$$

so $w \in Ch(\mu(\bar{f}) \cup \{w\}, P(\bar{f}))$, by the definition on $V(w, \mu)$.

But μ is a matching, so we have that $w \notin \mu(\bar{f})$ and $\mu(w) \in V(w, \mu)$. This implies that $\bar{f}P(w)\mu(w)$, and $w \in Ch(\mu(f) \cup \{w\}, P(f)) = C$. Let $A = C \cap \mu(f) = C - \{w\}$, and $B = \{w\}$. Then (B, f) blocks* μ . Hence, for all $w \in W$, $\mu(w) = (T\mu)(w)$. This and statement (5) imply $\mu = T\mu$. Hence $\mu \in \mathcal{E}$. ■

4 The T -algorithm

The T -algorithm is very simple: start at some $\nu \in \mathcal{V}$ and iterate $T\nu$ until two iterations are identical. When two iterations are identical, stop.

We prove that, when the T -algorithm stops, it must be at a core matching. Further, we show that once the T -algorithm hits on a matching μ (not just a pre-matching), μ must be in the core. Hence, the T -algorithm “wanders” around pre-matchings until it finds a matching—it will stop at the matching it finds, and it will be a core matching. It follows, of course, that if the core is empty the algorithm must cycle, and that the cycle will only involve pre-matchings that are not matchings.

The algorithm is very simple, but we include a detailed description to be as clear as possible, and to introduce some notation that we shall use in the proofs below:

Algorithm 1 (T -algorithm)

1. Set $\nu^0 = \nu$. Set $\nu^1 = T\nu^0$ and $k = 1$.
2. While $\nu^k \neq \nu^{k-1}$, do:
 - (a) Set $k = k + 1$,
 - (b) set $\nu^k = T\nu^{k-1}$.
3. Set $\mu = \nu^k$. Stop.

Proposition 5 *If the T -algorithm stops at $\mu \in \mathcal{V}$, then μ is a matching ($\mu \in \mathcal{M}$) and μ is in the core. If ν^k is in the core, for some iteration k of the T -algorithm, then the algorithm stops at $\mu = \nu^k$.*

Proof. If the algorithm stops at μ then $\nu^k = \nu^{k-1} = \mu$. Then, $\mu = T\nu^{k-1} = T\mu$, so $\mu \in \mathcal{E}$. By Corollary 4, $\mu \in C_w(P)$. On the other hand, if ν^k is in the core, then ν^k is a fixed point of T by Corollary 4. Then the T -algorithm stops at $\mu = \nu^k$. ■

Proposition 6 *If ν^k is a matching, for some iteration k in the T -algorithm, then the algorithm stops at $\mu = \nu^k$, and thus ν^k is a core matching.*

We shall present the proof of Proposition 6 as two lemmata:

Lemma 7 *Let $\nu \in \mathcal{V}$ be a pre-matching. If $\nu^1 = T\nu$ is a matching, then ν^1 is individually rational.*

Proof. For all $f \in F$,

$$\nu^1(f) = (T\nu)(f) = Ch(U(f, \nu), P(f)).$$

So

$$\begin{aligned} Ch(\nu^1(f), P(f)) &\stackrel{(1)}{=} Ch(Ch(U(f, \nu), P(f)), P(f)) \\ &\stackrel{(2)}{=} Ch(U(f, \nu), P(f)) \\ &\stackrel{(3)}{=} \nu^1(f). \end{aligned}$$

Equalities (1) and (3) follow because $\nu^1 = T\nu$. Equality (2) is a property of choice sets. So, for all $f \in F$,

$$\nu^1(f) = Ch(\nu^1(f), P(f)). \quad (6)$$

For all $w \in W$,

$$\nu^1(w) = (T\nu)(w) = \max_{P(w)} \{V(w, \nu)\} \cup \{\emptyset\}.$$

So,

$$\nu^1(w) R(w) \emptyset. \quad (7)$$

Statements (6) and (7) show that ν^1 is individually rational. ■

Lemma 8 *If $T\nu$ is a matching, for some $\nu \in \mathcal{V}$, then $T\nu$ is a core matching. Thus if $\nu^k = T\nu^{k-1}$ is a matching, then the T -algorithm must stop at ν^k , and ν^k is a core matching.*

Proof. We shall prove that, if $\nu \in \mathcal{V}$ is a pre-matching, and $\nu^1 = T\nu$ is a matching, then $\nu^1 \in S^*(P)$. The lemma follows from Proposition 5, as $S^*(P) = \mathcal{E} = C_w(P)$.

Assume that $\nu^1 \notin S^*(P)$. By Lemma 7, ν^1 is individually rational, so there must exist a (B, f) that blocks ν^1 . Hence, for all $w \in B$ ($B \neq \emptyset$),

$$f P(w) \nu^1(w), \quad (8)$$

and there must exist $A \subseteq \nu^1(f)$ such that

$$[A \cup B] P(f) \nu^1(f). \quad (9)$$

But ν^1 is a matching, and $\nu^1 = T\nu$, so we have that for all $w \in B$, $w \in \nu^1(\nu^1(w))$. But $\nu^1(\nu^1(w)) = (T\nu)(\nu^1(w)) = Ch(U(\nu^1(w), \nu), P(\nu^1(w)))$, so that, $w \in U(\nu^1(w), \nu)$, i.e.

$$\nu^1(w) R(w) \nu(w). \quad (10)$$

Statements (8) and (10) imply that, for all $w \in B$, $fR(w) \nu(w)$, i.e.

$$B \subseteq U(f, \nu). \quad (11)$$

Now, $A \subseteq \nu^1(f)$ and $\nu^1(f) = (T\nu)(f) = Ch(U(f, \nu), P(f))$, so

$$A \subseteq U(f, \nu). \quad (12)$$

Statements (9), (11), and (12) contradict that $\nu^1(f) = (T\nu)(f) = Ch(U(f, \nu), P(f))$. Thus $\nu^1 \in S^*(P)$. ■

Remarks.

- Proposition 6, and Example 9 below, imply that the T -algorithm and the DAA are different algorithms; we have not simply written down the DAA using different language.
- Unlike the DAA, the T -algorithm only stops at a core matchings. Example 9 below shows how the T -algorithm succeeds in finding a core matching, while the DAA stops at a non-core matching.
- When the core is empty, the T -algorithm must cycle. We present an example as “supplementary material” in <http://www.nyu.edu/jet/supplementary.html> and in the working-paper version ([4]) of the paper.

Example 9 Let $F = \{f_1, f_2\}$, $W = \{w_1, w_2, w_3, w_4\}$, and the preference profile P be defined by: $P(f_1) = \{w_1, w_2\}, \{w_3, w_4\}, \{w_1, w_3\}, \{w_2, w_4\}, \{w_1\}, \{w_2\}, \{w_3\}, \{w_4\}$, $P(f_2) = \{w_1, w_2\}, \{w_1, w_3\}, \{w_2, w_4\}, \{w_3, w_4\}, \{w_1\}, \{w_2\}, \{w_3\}, \{w_4\}$, $P(w_1) = f_1, f_2$, $P(w_2) = f_2, f_1$, $P(w_3) = f_1, f_2$, and $P(w_4) = f_1, f_2$.

Note that P is not substitutable, as $w_1 \in Ch(\{w_1, w_2, w_3, w_4\}, P(f_1)) = \{w_1, w_2\}$ while $w_1 \notin Ch(\{w_1, w_3, w_4\}, P(f_1)) = \{w_3, w_4\}$. The T -algorithm starting at

$$\nu^0 = \begin{array}{cccccc} & f_1 & f_2 & w_1 & w_2 & w_3 & w_4 \\ \nu^0 = & \{w_1, w_2\} & \{w_1, w_2\} & \emptyset & \emptyset & \emptyset & \emptyset \end{array}$$

does:

	f_1	f_2	w_1	w_2	w_3	w_4
$U(f, \nu^0)/V(w, \nu^0) =$	$\{w_1, w_2, w_3, w_4\}$	$\{w_1, w_2, w_3, w_4\}$	f_1, f_2	f_1, f_2	\emptyset	\emptyset
$\nu^1 =$	$\{w_1, w_2\}$	$\{w_1, w_2\}$	f_1	f_2	\emptyset	\emptyset
$U(f, \nu^1)/V(w, \nu^1) =$	$\{w_1, w_3, w_4\}$	$\{w_2, w_3, w_4\}$	f_1, f_2	f_1, f_2	\emptyset	\emptyset
$\nu^2 =$	$\{w_3, w_4\}$	$\{w_2, w_4\}$	f_1	f_2	\emptyset	\emptyset
$U(f, \nu^2)/V(w, \nu^2) =$	$\{w_1, w_3, w_4\}$	$\{w_2, w_3, w_4\}$	f_2	f_2	f_1	f_1, f_2
$\nu^3 =$	$\{w_3, w_4\}$	$\{w_2, w_4\}$	f_2	f_2	f_1	f_1
$U(f, \nu^3)/V(w, \nu^3) =$	$\{w_1, w_3, w_4\}$	$\{w_1, w_2\}$	f_2	f_2	f_1	f_1, f_2
$\nu^4 =$	$\{w_3, w_4\}$	$\{w_1, w_2\}$	f_2	f_2	f_1	f_1

Note that ν^i for $i = 0, 1, 2, 3$ are not matchings; ν^4 is a matching so, by Proposition 6, ν^4 is a core matching.

On the other hand, the DAA stops at a matching that is not in the core. For the DAA, the equivalent to starting the T -algorithm at ν^0 is, as we shall see in Section 5, the DAA with firms proposing. The DAA with firms proposing stops at the matching

$$\mu = \begin{array}{cccccc} f_1 & f_2 & w_1 & w_2 & w_3 & w_4 \\ \{w_3, w_4\} & \{w_2\} & \emptyset & f_2 & f_1 & f_1. \end{array}$$

But μ is not in the core, as $f_2 P(w_1) \mu(w_1) = \emptyset$, $f_2 R(w_2) \mu(w_2) = f_2$, and $\{w_1, w_2\} P(f_2) \{w_2\}$ implies that $(\{w_1, w_2\}, f_2)$ blocks* μ .

5 The lattice structure of the core

We shall introduce a partial order on \mathcal{V} such that, if preferences are substitutable, T is a monotone increasing map. Tarski's fixed point theorem then delivers a lattice structure on \mathcal{E} , and thus on $C_w(P)$.

Definition 4 Define the following partial orders on \mathcal{V}_F , \mathcal{V}_W and \mathcal{V} :

1. $P(F)$ on \mathcal{V}_F by $\nu_F P(F) \nu'_F$ if and only if $\nu'_F \neq \nu_F$ and, for all f in F , $\nu_F(f) = \nu'_F(f)$ or

$$\nu_F(f) = Ch(\nu_F(f) \cup \nu'_F(f), P(f)).$$

2. $P(W)$ on \mathcal{V}_W by $\nu_W P(W) \nu'_W$ if and only if $\nu'_W \neq \nu_W$ and

$$\nu_W(w)R(w) \nu'_W(w) \text{ for all } w \in W.$$

3. The weak partial orders associated to $P(F)$ and $P(W)$ are denoted $R(F)$ and $R(W)$, defined as: $\nu_F R(F) \nu'_F$ if $\nu_F = \nu'_F$ or $\nu_F P(F) \nu'_F$, and $\nu_W R(W) \nu'_W$ if $\nu_W = \nu'_W$ or $\nu_W P(W) \nu'_W$.

4. $P(F)$ on \mathcal{V} by $\nu P(F) \nu'$ iff $\nu_F P(F) \nu'_F$ and $\nu'_W R(W) \nu_W$, or $\nu_F R(F) \nu'_F$ and $\nu'_W P(W) \nu_W$.

5. $P(W)$ on \mathcal{V} by $\nu P(W) \nu'$ iff $\nu_W P(W) \nu'_W$ and $\nu'_F R(F) \nu_F$, or $\nu_W R(W) \nu'_W$ and $\nu'_F P(F) \nu_F$.

Definition 5 A firm f 's preference ordering $P(f)$ satisfies **substitutability** if for any set $S \subseteq W$ containing workers w and \bar{w} ($w \neq \bar{w}$), if $w \in Ch(S, P(f))$ then $w \in Ch(S \setminus \{\bar{w}\}, P(f))$. A preference profile P is **substitutable** if, for each firm f , the preference ordering $P(f)$ satisfies substitutability.

Theorem 10 Let P be substitutable. Then $(C_w(P), P(F))$ and $(C_w(P), P(W))$ are non-empty complete lattices, and $\inf_{P(F)} C_w(P) = \sup_{P(W)} C_w(P)$, and $\sup_{P(F)} C_w(P) = \inf_{P(W)} C_w(P)$.

Theorem 10 says that there are two distinguished core matchings, μ_W and μ_F such that: For firms, $\mu_W = \inf_{P(F)} C_w(P)$ is worse, and $\mu_F = \sup_{P(F)} C_w(P)$ is better, than any other core matchings. For workers $\mu_W = \sup_{P(W)} C_w(P)$ is better, and $\mu_F = \inf_{P(W)} C_w(P)$ is worse, than any other core matching.

Proof of Theorem 10. First, $(\mathcal{V}', P(F))$ is a complete lattice—due to space restrictions, we refer to the working paper version of the paper ([4]) for the proof. We shall present the proof of the Theorem as 2 simple lemmata. We assume that P is substitutable in the rest of the section.

Notation. Let $\mathcal{V}' = \{\nu \in \mathcal{V} : \nu(s)R(s)\emptyset, \text{ for all } s \in F \cup W\}$.

Remark. For all $\nu \in \mathcal{V}$, $T\nu \in \mathcal{V}'$

Lemma 11 *Let μ and μ' be pre-matchings. If $\mu'R(F)\mu$ then, for all $w \in W$ and $f \in F$, $U(f, \mu) \subseteq U(f, \mu')$, and $V(w, \mu) \supseteq V(w, \mu')$.*

Proof. Let $w \in U(f, \mu)$. We have that $fR(w)\mu(w)$, but $\mu(w)R(w)\mu'(w)$ so $w \in U(f, \mu')$. This proves $U(f, \mu) \subseteq U(f, \mu')$.

Now we shall prove that $V(w, \mu) \supseteq V(w, \mu')$. First, if $V(w, \mu') = \{\emptyset\}$, then there is nothing to prove, as $\{\emptyset\} = V(w, \mu') \subseteq V(w, \mu)$. Suppose that $V(w, \mu') \neq \{\emptyset\}$, and let $f \in V(w, \mu')$. Then, $w \in Ch(\mu'(f) \cup \{w\}, P(f))$.

But $\mu'R(F)\mu$, so the definition of $R(F)$ implies that, for all $f \in F$, either $\mu'(f) = \mu(f)$ so $w \in Ch(\mu(f) \cup \{w\}, P(f))$, or $\mu'(f) = Ch(\mu'(f) \cup \mu(f), P(f))$. Then $w \in Ch(\mu'(f) \cup \{w\}, P(f))$ implies that

$$\begin{aligned} w &\in Ch(\mu'(f) \cup \{w\}, P(f)) \\ &= Ch(Ch(\mu'(f) \cup \mu(f), P(f)) \cup \{w\}, P(f)) \\ &\stackrel{(1)}{=} Ch(\mu'(f) \cup \mu(f) \cup \{w\}, P(f)). \end{aligned}$$

Equality (1) is from [3, Proposition 2.3]. Substitutability of P implies that $w \in Ch(\mu(f) \cup \{w\}, P(f))$. Then $f \in V(w, \mu)$, and thus $V(w, \mu) \supseteq V(w, \mu')$. \blacksquare

Lemma 12 *\mathcal{E} is a nonempty and complete lattice.*

Proof. First we show that $T|_{\mathcal{V}}$ is monotone increasing. That is, whenever $\mu'R(F)\mu$, we have $(T\mu')R(F)(T\mu)$.

Let $\mu'R(F)\mu$, and fix $f \in F$ and $w \in W$. Lemma 11 says that $U(f, \mu) \subseteq U(f, \mu')$. We first show that

$$Ch(U(f, \mu'), P(f)) = Ch([Ch(U(f, \mu'), P(f)) \cup Ch(U(f, \mu), P(f))], P(f)). \quad (13)$$

To see this, let $S \subseteq Ch(U(f, \mu'), P(f)) \cup Ch(U(f, \mu), P(f))$. Then $S \subseteq U(f, \mu) \cup U(f, \mu') = U(f, \mu')$, so $Ch(U(f, \mu'), P(f))R(f)S$. But, $Ch(U(f, \mu'), P(f)) \subseteq Ch(U(f, \mu'), P(f)) \cup Ch(U(f, \mu), P(f))$, so we have established statement 13.

Now, $(T\mu')(f) = Ch(U(f, \mu'), P(f))$ and $(T\mu)(f) = Ch(U(f, \mu), P(f))$, so statement 13 implies that

$$(T\mu')(f) = Ch([(T\mu')(f) \cup (T\mu)(f)], P(f)). \quad (14)$$

We now show that $(T\mu)(w)R(w)(T\mu')(w)$. Lemma 11 says that $V(w, \mu) \subseteq V(w, \mu')$. So,

$$(T\mu)(w) = \max_{P(f)} \{V(w, \mu)\} R(w) \max_{P(f)} \{V(w, \mu')\} = (T\mu')(w). \quad (15)$$

Statements 14 and 15 imply that $(T\mu')R(F)(T\mu)$, as $f \in F$ and $w \in W$ were arbitrary.

Finally, $T(\mathcal{V}) \subseteq \mathcal{V}'$ so $\mathcal{E} \subseteq \mathcal{V}'$, and \mathcal{E} equals the set of fixed points of $T|_{\mathcal{V}'}$. $T(\mathcal{V}) \subseteq \mathcal{V}'$ also implies that the restricted map $T|_{\mathcal{V}'}$ has range in \mathcal{V}' .

Now we have that $T|_{\mathcal{V}'} : \mathcal{V}' \rightarrow \mathcal{V}'$ is monotone increasing, and \mathcal{V}' is a complete lattice. So Tarski’s fixed point theorem implies that $(\mathcal{E}, P(F))$ is a non-empty complete lattice. ■

Corollary 4 and Lemma 12 finish the proof of Theorem 10. See [4] for full details.

6 The T -algorithm and substitutability.

When preferences are substitutable, the T -algorithm

1. finds the worker-optimal core matching ($\inf_{P(F)} C_w(P)$) and the firm-optimal core matching ($\sup_{P(F)} C_w(P)$),
2. is computationally efficient—we can give a bound on its computational complexity and show that the bound is likely to be quite small,
3. calculates the lattice operations “join” and “meet,” on the $C_w(P)$ lattice.

Result 1 is important because it shows that the T -algorithm does the job of the DAA. Further, by result 2 we have an idea of the T -algorithm’s computational complexity, which is an open problem for the DAA in the many-to-one case. In a very simple example, we show that, while the “size” of the problem is 30,000 iterations, the T -algorithm needs at most 9 iterations to find a core matching. Our result on the complexity of the T -algorithm is similar to Topkis’s [14] (see also [15]) results on algorithms for finding equilibria in supermodular games.

Result 3: A number of papers have sought to improve on Blair’s lattice operations ([8, 2]) by restricting firms’ preferences. We show that, with only substitutability, the T -algorithm can *compute* the lattice operations—thus allowing for a practical solution to the problem of Blair’s lattice operations.

6.1 The firm- and worker-optimal core matchings

Proposition 13 *If P is substitutable, the T -algorithm starting at $\inf_{P(F)} \mathcal{V}'$ stops at the worker-optimal core matching $\inf_{P(F)} C_w(P)$; the T -algorithm starting at $\sup_{P(F)} \mathcal{V}'$ stops at the firm-optimal core matching $\sup_{P(F)} C_w(P)$.*

Proof. When preferences are substitutable, T is a monotone map on the complete lattice \mathcal{V}' . Proposition 13 follows then from standard results, see [14, 15]. ■

Remarks.

1. The pre-matching $\inf_{P(F)} \mathcal{V}'$ is $\nu \in \mathcal{V}$ with $\nu(f) = \{\emptyset\}$ and $\nu(w) = \max_{P(w)} F$. The pre-matching $\sup_{P(F)} \mathcal{V}'$ is ν with $\nu(f) = Ch(W, P(f))$ and $\nu(w) = \{\emptyset\}$.
2. The T -algorithm starting at points that are not $\inf_{P(F)} \mathcal{V}'$ or $\sup_{P(F)} \mathcal{V}'$ may not converge. For example, one can show that the T -algorithm starting at the pre-matching $\nu = \left((\sup_{P(F)} \mathcal{V}')_F, (\inf_{P(F)} \mathcal{V}')_W \right)$ cycles.

6.2 Complexity of the T -algorithm

We shall now give a bound on the computational complexity of the T -algorithm, but we need some auxiliary definitions first.

For each f , let $X_f = \{A \subseteq W : AR(f)\emptyset\}$, and let \geq_f be the partial order on X_f defined by $A \geq B$ iff $A = Ch(A \cup B, P(f))$ or $A = B$. Let q_f be the size of the longest \geq_f -chain in X_f . For each w , let $X_w = \{f \in F : fR(w)\emptyset\}$, and let q_w be the number of elements in X_w . We present an example as “supplementary material” in <http://www.nyu.edu/jet/supplementary.html> and in the working-paper version ([4]) of the paper.

Proposition 14 *If P is substitutable, the T -algorithm starting at $\inf_{P(F)} \mathcal{V}'$ ($\sup_{P(F)} \mathcal{V}'$) stops at the worker-optimal (firm-optimal) core matching in less than*

$$\sum_{s \in F \cup W} (q_s - 1)$$

iterations.

Proof. Let $\nu^0, \nu^1, \dots, \nu^k$ be the sequence generated by the T -algorithm before it stops at a core matching ν^k . The number of iterations before the algorithm stops at ν^k is k , we shall prove that $k \leq \sum_{s \in F \cup W} (q_s - 1)$.

Assume—without loss of generality—that we started the T -algorithm at $\nu^0 = \inf_{P(F)} \mathcal{V}'$. Since $\nu^1 = (T\nu^0)P(F)\nu^0$ and T is monotone increasing (see the proof of Theorem 10),

$$\nu^k P(F) \nu^{k-1} P(F) \dots \nu^1 P(F) \nu^0.$$

For each l , $1 \leq l \leq k$, it must be that $\nu^l(f) = \nu^{l-1}(f)$ or that $\nu^l(f) = Ch(\nu^{l-1}(f) \cup \nu^{l-1}(f), P(f))$ for all f , and $\nu^{l-1}(w)R(w)\nu^l(w)$ for all w . Also, for each l , $1 \leq l \leq k$,

either $\nu^l(f) \neq \nu^{l-1}(f)$ and $\nu^l(f) = Ch(\nu^l(f) \cup \nu^{l-1}(f), P(f))$ for some f or $\nu^{l-1}(w)P(w)\nu^l(w)$ for some w —recall that $P(F)$ is a strict order, see Definition 4. Say that l is an f -step if $\nu^l(f) \neq \nu^{l-1}(f)$ and $\nu^l(f) = Ch(\nu^l(f) \cup \nu^{l-1}(f), P(f))$ for firm f . Say that l is a w -step if $\nu^{l-1}(w)P(w)\nu^l(w)$ for worker w . There cannot be more than $q_s - 1$ s -steps, as a number of, say r steps involves a chain of $r + 1$ elements in X_s . Now k is smaller than the sum over all $s \in W \cup F$ of all s -steps, so

$$k \leq \sum_{s \in F \cup W} (q_s - 1).$$

But k is the number of iterations of the T -algorithm before it finds ν^k . ■

Remark. Recall that $|W| = m$ and $|F| = n$, so the number of pre-matchings is

$$2^{nm}(n+1)^m,$$

a number which is normally orders of magnitude larger than the bound in Proposition 14: In an example we develop as “supplementary material” in <http://www.nyu.edu/jet/supplementary.html> (see also [4]) Proposition 14 implies that the T -algorithm converges before

$$\sum_{s \in F \cup W} (q_s - 1) = 9$$

iterations. Compare with the number of pre-matchings: 32768. The number of matchings is smaller than the number of pre-matchings, but still substantially larger than 9. It may be unfair to compare the complexity of the T -algorithm to the number of pre-matchings, after all the T -algorithm will not involve pre-matchings where agents are matched to unacceptable partners—but still, the number of pre-matchings that only involve acceptable partners is 9261.

6.3 Computing the lattice operations

Let $\mu^1, \mu^2 \in C_w(P)$. We shall look at the problem of finding $\mu^1 \vee_{P(F)} \mu^2$, the join (least upper bound) of μ^1 and μ^2 in the $(C_w(P), P(F))$ -lattice. One can solve the problem of finding $\mu^1 \wedge_{P(F)} \mu^2$ in the $(C_w(P), P(F))$ -lattice in a similar way.

Martínez, Massó, Neme and Oviedo ([8, 9]) prove that, when preferences satisfy substitutability and an additional assumption (which they call q -separability) there is a simple formula for $\mu^1 \vee_{P(F)} \mu^2$ (see [2] for a similar result). Martínez, Massó, Neme and Oviedo prove that $\mu^1 \vee_{P(F)} \mu^2$ is

$$\bar{\nu}(s) = \begin{cases} Ch(\mu^1(s) \cup \mu^2(s), P(s)) & \text{if } s \in F \\ \min_{P(s)} \{\mu^1(s), \mu^2(s)\} & \text{if } s \in W. \end{cases} \quad (16)$$

In general, though, when preferences are only substitutable, there is no formula for $\mu^1 \vee_{P(F)} \mu^2$. In fact, $\bar{\nu}$ defined in (16) is a pre-matching but not necessarily a matching. We prove that, if one starts the T -algorithm at $\bar{\nu}$, it will stop at $\mu^1 \vee_{P(F)} \mu^2$. Hence we give a practical solution to Blair's problem of calculating $\mu^1 \vee_{P(F)} \mu^2$.³

Theorem 15 *Let P be substitutable, let μ^1 and μ^2 be core matchings, and $\bar{\nu}$ be defined as in (16). The T -algorithm started at $\bar{\nu}$ stops at $\mu^1 \vee_{P(F)} \mu^2$.*

Proof. Let $\hat{\mathcal{V}} = \{\nu \in V : \nu R(F) \bar{\nu}\}$. First we prove (1) that $T\hat{\mathcal{V}} \subseteq \hat{\mathcal{V}}$. Then we prove (2) that T -algorithm starting at $\bar{\nu}$ stops at $\mu^1 \vee_{P(F)} \mu^2$.

(1) First we shall prove that $\bar{\nu}$ is the least upper bound on $\{\mu^1, \mu^2\}$ in \mathcal{V} . By the definition of $\bar{\nu}$, $\bar{\nu} R(F) \mu^1$ and $\bar{\nu} R(F) \mu^2$.

Let ν be an upper bound on $\{\mu^1, \mu^2\}$ in \mathcal{V} , we shall prove that $\nu R(F) \bar{\nu}$. For any $f \in F$:

$$\begin{aligned} \nu(f) R(f) \mu^1(f) &\iff \nu(f) \stackrel{(1)}{=} Ch(\nu(f) \cup \mu^1(f), P(f)) \\ \nu(f) R(f) \mu^2(f) &\iff \nu(f) \stackrel{(2)}{=} Ch(\nu(f) \cup \mu^2(f), P(f)). \end{aligned}$$

We have that

$$\begin{aligned} Ch(\nu(f) \cup \bar{\nu}(f), P(f)) &\stackrel{(3)}{=} Ch(\nu(f) \cup Ch(\mu^1(f) \cup \mu^2(f), P(f)), P(f)) \\ &\stackrel{(4)}{=} Ch(\nu(f) \cup \mu^1(f) \cup \mu^2(f), P(f)) \\ &\stackrel{(5)}{=} Ch(Ch(\nu(f) \cup \mu^1(f), P(f)) \cup \mu^2(f), P(f)) \\ &\stackrel{(6)}{=} Ch(\nu(f) \cup \mu^2(f), P(f)) \\ &\stackrel{(7)}{=} \nu(f). \end{aligned}$$

Where equality (3) follows from the definition of $\bar{\nu}$. Equalities (4) and (5) follow from substitutability and [3, Proposition 2.3]. Equality (6) follows from equality (1). Equality (7) follows from equality (2). Thus, $\nu_F R(F) \bar{\nu}_F$. Now, $\nu R(F) \mu^1$ and $\nu R(F) \mu^2$ imply that, for any $w \in W$, $\mu^1(w) R(w) \nu(w)$, and $\mu^2(w) R(w) \nu(w)$. But then,

$$\bar{\nu}(w) = \min_{P(w)} \{\mu^1(w), \mu^2(w)\} R(w) \nu(w).$$

This and $\nu_F R(F) \bar{\nu}_F$ implies that $\nu R(F) \bar{\nu}$. Hence $\bar{\nu}$ is the least upper bound on $\{\mu^1, \mu^2\}$ in \mathcal{V} .

Now we shall prove that $T\hat{\mathcal{V}} \subseteq \hat{\mathcal{V}}$. Let $\nu \in \hat{\mathcal{V}}$, then $\nu R(F) \bar{\nu} R(F) \mu^1$ and $\nu R(F) \bar{\nu} R(F) \mu^2$. But then $T\nu R(F) T\mu^1$ and $T\nu R(F) T\mu^2$, as T is monotone. Now, μ^1 and μ^2 are core

³Blair's proof of the existence of $\mu^1 \vee_{P(F)} \mu^2$ can also be expressed as an algorithm for calculating $\mu^1 \vee_{P(F)} \mu^2$.

matchings, so $\mu^1 = T\mu^1$ and $\mu^2 = T\mu^2$; then $T\nu$ is an upper bound on $\{\mu^1, \mu^2\}$ in \mathcal{V} . But $\bar{\nu}$ is the least upper bound, so $T\nu R(F)\bar{\nu}$ and thus $T\nu \in \hat{\mathcal{V}}$.

(2) We know that $T\hat{\mathcal{V}} \subseteq \hat{\mathcal{V}}$, so $T|_{\hat{\mathcal{V}}} : \hat{\mathcal{V}} \rightarrow \hat{\mathcal{V}}$. T is monotone and $\bar{\nu}$ is the smallest pre-matching in $\hat{\mathcal{V}}$. By Topkis's [14] results, the T -algorithm stops at the $R(F)$ -smallest fixed point of $T|_{\hat{\mathcal{V}}}$; but this is the $R(F)$ -smallest fixed point of T , of those fixed points that are $R(F)$ -larger than both μ^1 and μ^2 . Then the T -algorithm stops at $\mu^1 \vee_{P(F)} \mu^2$. ■

Remarks.

1. A key part of the proof of Theorem 15 is that $\bar{\nu}$ is the join of μ^1 and μ^2 in \mathcal{V} . By a similar argument, the T -algorithm calculates the meet, $\mu^1 \wedge_{P(F)} \mu^2$, of any two core matchings μ^1 and μ^2 . One only needs to start the algorithm at the meet of μ^1 and μ^2 in \mathcal{V} instead of the join.⁴
2. The bound presented in Proposition 14 also bounds the number of T -iterations necessary to calculate $\mu^1 \vee_{P(F)} \mu^2$.

We now illustrate how the algorithm finds the join of two core matchings:

Example 16 ([10]). Let $F = \{f_1, f_2, f_3, f_4, f_5\}$ be the set of firms and

$W = \{w_1, w_2, w_3, w_4, w_5, w_6\}$ be the set of workers. As in [10], we specify only preferences among the choices we shall need. $P(f_1) = \{w_4\}, \{w_1\}, \{w_2, w_3, w_5, w_6\}, \dots, \{w_5\}$, $P(f_2) = \{w_2\}, \{w_1, w_3\}$, $P(f_3) = \{w_3\}, \{w_2\}$, $P(f_4) = \{w_5\}, \{w_4, w_6\}$, $P(f_5) = \{w_6\}, \{w_5\}$, $P(w_1) = f_2, f_1$, $P(w_2) = f_1, f_3, f_2$, $P(w_3) = f_1, f_2, f_3$, $P(w_4) = f_4, f_1$, $P(w_5) = f_1, f_5, f_4$, $P(w_6) = f_1, f_4, f_5$, Consider the following two core matchings

$$\mu_1 = \begin{array}{ccccc} f_1 & f_2 & f_3 & f_4 & f_5 \\ \{w_1\} & \{w_2\} & \{w_3\} & \{w_4, w_6\} & \{w_5\} \end{array} \text{ and}$$

$$\mu_2 = \begin{array}{ccccc} f_1 & f_2 & f_3 & f_4 & f_5 \\ \{w_4\} & \{w_1, w_3\} & \{w_2\} & \{w_5\} & \{w_6\} \end{array}.$$

$$\begin{array}{cccccccccc} & f_1 & f_2 & f_3 & f_4 & f_5 & w_1 & w_2 & w_3 & w_4 & w_5 & w_6 \\ \nu^1 = & w_4 & w_2 & w_3 & w_5 & w_6 & f_1 & f_2 & f_3 & f_1 & f_4 & f_5 \\ U(f, \nu^1)/V(w, \nu^1) = & w_1, w_4 & w_2 & w_3 & w_5 & w_6 & \emptyset & f_2 & f_3 & f_1 & f_4 & f_5 \\ \nu^2 = & w_4 & w_2 & w_3 & w_5 & w_6 & \emptyset & f_2 & f_3 & f_1 & f_4 & f_5. \end{array}$$

ν^2 is a matching and $\nu^2 = \mu_1 \vee_{P(F)} \mu_2$.

⁴See [2] on how to calculate the meet of μ^1 and μ^2 in \mathcal{V} .

References

- [1] H. Adachi, On a Characterization of Stable Matchings, *Economic Letters*. **68** (2000), 43-49.
- [2] A. Alkan, “A Class of Multipartner Matching Markets with a Strong Lattice Structure,” *Econ. Theory* **19** (2002), 737–746.
- [3] C. Blair, The Lattice Structure of the Set of Stable Matchings with Multiple Partners, *Mathematics of Operations Research* **13** (1988), 619–628.
- [4] F. Echenique and J. Oviedo, Core Many-to-one Matchings by Fixed-Point Methods, *Caltech Social Science Working Paper* 1140, (2002).
- [5] T. Fleiner, “A Fixed-point Approach to Stable Matchings and Some Applications”, Technical Report, Egervary Research Group on Combinatorial Optimization, 2001.
- [6] D. Gusfield and R. Irving. *The Stable Marriage Problem: Structure and Algorithms*, MIT Press, Cambridge MA, 1989.
- [7] A. Kelso and V. Crawford, Job Matching, Coalition Formation, and Gross Substitutes, *Econometrica*. **50** (1982), 1483–1504.
- [8] R. Martínez, J. Massó, A. Neme, and J. Oviedo, Single Agents and the Set of Many-to-one Stable Matchings, *J. of Econ. Theory* **91** (2000), 91-105.
- [9] R. Martínez, J. Massó, A. Neme, and J. Oviedo, On the Lattice Structure of the Set of Stable Matchings for a Many-to-one Model, *Optimization* **50** (2001), 439-457.
- [10] A. Roth, Conflict and Coincidence of Interest in Job Matching: Some New Results and Open Questions, *Mathematics of Operations Research* **10** (1985), 379-389.
- [11] A. Roth and M. Sotomayor, Interior Points in the Core of Two-Sided Matching Markets, *J. Econ. Theory* **45** (1988), 85-101.

- [12] A. Roth and M. Sotomayor, "Two-sided Matching: A Study in Game-Theoretic Modelling and Analysis," Econometric Society Monographs No. 18, Cambridge University Press, Cambridge England, 1990.
- [13] M. Sotomayor, Three Remarks on the Many-to-many Stable Matching Problem, *Math. Soc. Sci.***38** (1999), 55-70.
- [14] D. M. Topkis, Equilibrium Points in Non-zero sum n -person submodular games, *SIAM Journal on Control and Optimization* **17** (1979), 773-787.
- [15] D. M. Topkis, "Supermodularity and Complementarity," Princeton University Press, Princeton NJ 1998.

List of Symbols

Symbol	Explanation	L ^A T _E Xcode
μ	Greek letter “mu.”	<code>\mu</code>
ν	Greek letter “nu.”	<code>\nu</code>
\mathcal{V}	Calligraphic letter “V.”	<code>\mathcal{V}</code>
\mathcal{M}	Calligraphic letter “M.”	<code>\mathcal{M}</code>
\mathcal{E}	Calligraphic letter “E.”	<code>\mathcal{E}</code>
Ch	Letters “C” and “h.”	<code>\$Ch\$</code>
\in	Belongs to set.	<code>\in</code>
\subseteq	Contained in set.	<code>\subseteq</code>
\cup	Union of sets	<code>\cup</code>
\cap	Union of sets	<code>\cap</code>
\vee	Lattice operation “join.”	<code>\vee</code>
\wedge	Lattice operation “meet.”	<code>\wedge</code>