A Matching Theory Approach to the Time Minimization Assignment Problem.

Mir Adnan Mahmood *

Department of Economics, The Ohio State University

May 29, 2019

Abstract

We apply a mechanism design approach to the time minimization assignment problem studied in the operations research literature. A group of workers is to be assigned to tasks. Workers have preferences over tasks as well as scores that determine their compatibility with the tasks. Tasks have a priority schedule that is dependent on the workers' scores. We introduce a notion of time taken to complete a task based on the weakest link principle and use the metric of time minimization as a means of comparison. We look at existing matching mechanisms and compare how they perform in terms of notions of stability and time minimization. We find inconclusive evidence of a particular mechanism outperforming the others in this regard.

Keywords: Matching; Stability; Time Minimization; Strategy-Proof

JEL Classification: C78; D82;

*I would like to thank PJ Healy, Yaron Azrieli and John Rehbeck for their feedback.

1 Introduction

The Time Minimization Assignment Problem has been an integral part of the Operations Research literature. The standard problem consists of assigning n workers to m tasks, with $n \leq m$ (see for example Arora and Puri, 1998; Chauvet et al., 2000). Each worker-task pair has an associated "cost" that indicates the time taken to complete the task. The objective is to assign workers to tasks such that all tasks have exactly one worker assigned to them and the total cost (time) is minimized and is usually solved by way of Linear Programming. Recent developments in this literature revolve around utilizing a 2 stage approach (Sonia and Puri, 2008; Jain et al., 2018).

While the Time Minimization Assignment Problem has a plethora of practical applications, there is one glaring omission in the standard setup. The problem does not explicitly account for the fact that workers may have preferences over tasks. One could argue that the preferences are somewhat subsumed in the cost indices. However, the issue with this interpretation is that one cannot effectively disentangle the possibility of a workers' preferences not being correlated with their performance in a given task. Given this oversight, we develop a two-sided matching theory framework to tackle the time minimization assignment problem in which we explicitly allow workers to have preferences over tasks. This framework allows us to not only focus on positive aspects of task assignments (in this case time to completion) but also allows us to answer questions pertaining to normative aspects (incentive compatibility, fairness, stability etc.).

Ever since the introduction of the Gale-Shapley deferred acceptance algorithm (Gale and Shapley, 1962), matching theory has seen a surge in the literature, particularly due to its real world applications. Matching Theory has been useful in studying and improving upon real world scenarios like the School Choice Problems (Abdulkadiroğlu and Sönmez, 2003; Abdulkadiroğlu, 2005), college admissions (Balinski and Sönmez, 1999), object allocation (Ergin, 2002) and Kidney Exchange (Roth et al., 2004). A full survey of the theoretical advancements and applications for the standard theory is provided by Abdulkadiroğlu and Sönmez (2013).

Recently, however, much of the focus in matching theory has shifted towards incorporating distributional constraints. Distributional constraints are an integral part of many of the real world applications of matching theory. Some common examples are minimum quotas for schools (Fragiadakis et al., 2016), affirmative action policies in school choice (Abdulkadiroğlu, 2005; Kojima, 2012) and regional cap policies in the Japanese medical residency matching (Kamada and Kojima, 2015, 2018). Since distributional constraints often cause inconsistencies with the notion of stability, attempts have been made to develop matching mechanisms that satisfy notions of stability even under constraints. Kojima (2017) and Kamada and Kojima (2017) provide a brief survey of the developments.

Our framework comprises of a two-sided matching environment similar to the object allocation settings of Balinski and Sönmez (1999) and Ergin (2002). Workers have preferences over tasks and tasks have a priority schedule over the workers. This priority schedule stems from workers' compatibility with the tasks. We allow for the possibility that a workers' preferences are not dependent on her compatibility scores. We incorporate distributional constraints in a flavor similar to that in Fragiadakis et al. (2016) by introducing feasibility constraints – every task has to be assigned *at least one* worker. We then introduce our notion of "Time Minimization." Informally, a matching mechanism is time minimizing if it induces a matching that results in the lowest time taken to collectively complete all the tasks. The measure of time taken stems from workers' compatibility scores – higher compatibility yields a faster time to completion.

There are a couple of differences between our setting and the standard setup in the operations research literature. We focus on cases where the number of workers is at least as large as the number of tasks. In this case, there is a possibility of multiple workers being assigned to the same task. As a result, we allow the time taken to complete a task to depend *only* on the score of the "weakest link" – the minimum of the compatibility scores of the workers assigned to a task. This provides a somewhat more "natural" link to time to completion. To the best of our knowledge, this is the first approach to modelling the time minimization assignment problem using a mechanism design/matching environment.

The paper is structured as follows. Section 2 describes the model and some failures of standard mechanisms, in particular the Gale-Shapley deferred acceptance algorithm (Gale and Shapley, 1962) and introduces weaker notions of fairness and non-wastefulness. Section 3 discusses the details of two feasible mechanisms and their properties. Section 4 discusses a simple case of one-to-one matching. Section 5 provides an example that showcases the difficulties in a general setting. Section

6 concludes.

2 Model

The model is similar to the the one presented in Balinski and Sönmez (1999) and the priority based model of object allocation in Ergin (2002). There is a (finite) set of workers I and a set of tasks T with $|I| \geq |T|$. Worker $i \in I$ has preferences $\succ_i$ over the tasks T. Let $\succ = (\succ_1, \succ_2, \ldots, \succ_{|I|})$ denote the vector of preference profiles of all workers and let $\mathcal{P}^{|I|}$ be the set of all preference profiles. For the moment we assume that preferences are strict (for matching examples allowing for indifferences, see Erdil and Ergin 2017).

Each task $t \in T$ has a capacity limit over the number of workers assigned to task t. Denote this capacity by q_t. Let $Q = (q_1, q_2, \ldots, q_T)$ denote the vector of capacities of all tasks. Let $C = \{c_1, c_2, \ldots, c_N\}$ be a set of N compatibility categories. Each task t has a (single) category assigned to it. Denote this assignment by the function $f : T \to C$ where $f(t) = c \in C$. Each worker i has a score assigned to each compatibility characteristic. Let worker i's vector of characteristic scores be $s^i = (s^i(c_1), s^i(c_2), \ldots, s^i(c_N))$ and the vector of all workers' scores $s = (s^1, s^2, \ldots, s^I)$.

The compatibility scores of the workers and the mapping f induce a priority schedule of the tasks over the workers. For each task $t \in T$, worker i has a higher priority than worker i' if and only if $s^i(f(t)) > s^{i'}(f(t))$. Furthermore, the compatibility scores determine how "good" a worker is at a particular task, relative to the other workers. In particular, worker i is faster at completing task t than worker i' if and only if $s^i(f(t)) > s^{i'}(f(t))$. Thus the priority schedule is in line with how quickly workers can finish the assigned task.[1]

Definition 1. A matching μ is a mapping $\mu : I \to T$ that assigns a worker i to a task t such that $\forall t \in T, |\mu(t)| \leq q_t$.[2]

Worker i prefers his match under μ to his match under δ if and only if $\mu(i) \succ_i \delta(i)$.

[1]We assume that ties are broken arbitrarily to allow for only strict priorities.

[2]We slightly abuse notation here and set $\mu(t) = \{i \in I : \mu(i) = t\}$ to be the set of workers matched to task t.

2.1 Properties of Matching Mechanisms

2.1.1 Stability

We now review some of the main normative axioms that have been the center of matching theory:

Definition 2. A matching μ is **fair** if for all $i, i' \in I$ with $t' = \mu(i')$ we have that $t' \succ_i \mu(i)$ implies that $s^{i'}(f(t')) > s^i(f(t'))$ i.e if worker i prefers worker i' task to $\mu(i)$ then it must be the case that worker i' has a higher priority in task $t' = \mu(i')$ than worker i.

Definition 3. A matching μ is **non-wasteful** if $\forall t \in T$ and $\forall i \in I$, $t \succ_i \mu(i)$ implies that $|\mu(t)| = q_t$ i.e if a worker prefers a task to his assigned task, it must be the case that the task's capacity is filled.

Definition 4. A matching μ is **stable** if it is both fair and non-wasteful.[3]

The worker proposed deferred acceptance algorithm (henceforth WPDA) satisfies certain nice properties such as fairness, non-wastefulness and is strategy-proof. In fact, it is the only matching mechanism that satisfies these properties:

Theorem 1. *(Balinski and Sönmez, 1999) The WPDA is the only matching mechanism that is stable and strategy-proof.*

2.1.2 Feasibility

Since each task is integral to the project's completion, we require a feasibility constraint in the sense that each task has to have at least one worker assigned to it. More formally:

Definition 5. A matching μ is **feasible** if $\forall t \in T$, $1 \leq |\mu(t)| \leq q_t$.[4]

As we show later on, feasibility may cause problems with stability. In particular, as outlined in the literature (see for example Fragiadakis et al., 2016 and Goto et al., 2017), a stable matching may not exist. Fairness and non-wastefulness may not be satisfied simultaneously with feasibility.

[3]In the setting of Balinski and Sönmez (1999), stability requires an additional condition, individual rationality (all workers prefer being matched to not being matched). This is primarily due to the fact that in their setting, students have preferences over not being matched. Our setting is a special case in which we abstract from such preferences. Thus workers trivially satisfy individual rationality.

[4]This is a special case of the feasibility constraint in Fragiadakis et al. (2016). They generalize this to allow for varying minimum quotas.

2.1.3 Time Minimization

We introduce a new positive property for matching mechanisms, one appropriate for the time minimization assignment problem. In particular, we look at the time to completion of the tasks, as a collective whole, as a means to compare how "efficient" the matching mechanisms are. To do this, we first introduce a bit of notation. Under a matching μ, let $\underline{s}_t(\mu)$ be the minimum score of all the workers matched to task t, in category $f(t)$. For each task t, let $\tau_t : \mathbb{R}_+ \to \mathbb{R}_+$ be a strictly decreasing function that determines the time taken to complete the task. We assume that the time taken only depends on the minimum score $\underline{s}_t(\mu)$, to be in line with the notion of the "weakest" link. In the case that under a matching mechanism μ, $\mu(t) = \varnothing$ for some task t, then we set $\tau_t = \infty$.

We now introduce the concept of "Time Minimizing." Informally, a matching μ is time minimizing if it ensures that the tasks are completed (collectively) in the lowest time possible. Formally:

Definition 6. A matching μ is **time minimizing** if for any other matching δ, we have:

$$\sum_{t\in T} \tau_t(\underline{s}_t(\mu)) \leq \sum_{t\in T} \tau_t(\underline{s}_t(\delta))$$

This specification yields an immediate implication: Feasibility is a necessary condition for time minimization.

A matching mechanism φ is a function $\varphi : \mathcal{P}^{|I|} \to \mathcal{M}$, where $\mathcal{M}$ is the set of matchings. A matching mechanism is **feasible** if it always induces a feasible matching. A matching mechanism is **fair** (**non-wasteful**) if it always induces a fair (non-wasteful) matching. The definition of a matching mechanism being time minimizing follows similarly. We also introduce standard notions of a mechanism being strategy-proof.

2.1.4 Strategy-Proof

Definition 7. A matching mechanism φ is **strategy-proof** if for all workers $i \in I$, $\varphi_i(\succ) \succ_i \varphi_i(\succ_i', \succ_{-i})$ for all $\succ \in \mathcal{P}^{|I|}$ and for all $\succ_i'$.

In other words, a matching mechanism is strategy-proof if no worker has an incentive to lie about their preferences.

We follow Balinski and Sönmez (1999) and look at the *associated* two-sided matching problem $(I, T, \succ_I, \succ_T, Q)$ where $\succ_I$ is the preference profile of the workers as above and $\succ_T$ is a (fixed) priority schedule generated from the compatibility scores as follows: for all $t \in T$, $i \succ_t i'$ if and only if $s^i(f(t)) > s^{i'}(f(t))$. This priority schedule can be interpreted as a tasks' "preferences" over the workers. We apply our mechanisms to this associated problem. All the standard definitions of fairness, non-wastefulness and stability in a two-sided framework follow readily from above.

2.2 The impossibility of a Time Minimizing and Stable Matching Mechanism

The inclusion of a feasibility constraint leads to problems in obtaining a stable matching. Distributional constraints (regional quotas, minimum quotas etc.) tend to induce a conflict between non-wastefulness and fairness (for examples, see Fragiadakis et al., 2016 and Goto et al., 2017). We show, by means of an example that a stable matching may not always be feasible.

Example 2.1. Consider the following setup. $I = \{A, B, C\}$, $T = \{x, y, z\}$, $Q = \{2, 1, 1\}$. For simplicity, we consider an associated two-sided problem with the following preference profile and priority schedule:

A	B	C
x	x	z
z	y	x
y	z	y

x	y	z
A	A	C
B	C	A
C	B	B

Both the WPDA and the task proposed deferred acceptance (henceforth TPDA) algorithms yield the following (unique) stable matching:

$$\begin{pmatrix} x & y & z \\ \{A, B\} & \emptyset & C \end{pmatrix}$$

The above matching, while stable, is not feasible as there is no worker assigned to task y.

Example 2.1 shows that (in general) there are cases in which the *unique* stable matching may not always be feasible. This leads us to the following impossibility result:

Theorem 2. *There is no matching mechanism that is stable and feasible.*

Due to feasibility being a necessary condition for time minimization, we get our second impossibility result:

Corollary 1. *There is no matching mechanism that is stable and time minimizing.*

2.3 Weaker Notions of Fairness and Non-Wastefulness

Seeing the negative result highlighted in Theorem 2 and Corollary 1, we introduce weaker notions of fairness and non-wastefulness in line with the notion of feasibility:

2.3.1 Feasible Elimination of Justified Envy

Consider an associated two-sided matching problem $(I, T, \succ_I, \succ_T, Q)$ and a matching μ. We define a pair (i, t) to be a **feasible blocking pair** as follows:

Definition 8. A pair $(i, t) \in I \times T$ is a **feasible blocking pair** for matching μ if:

i) $t \succ_i \mu(i)$ and for some $i' \in \mu(t)$, $i \succ_t i'$ and

ii) The matching μ' such that $\mu'(i) = t$ and $\mu'(i') = \mu(i')$ for all $i' \neq i$ is feasible.

Definition 9. A matching μ **feasibly eliminates justified envy** if there exists no feasible blocking pair. A matching mechanism φ feasibly eliminates justified envy if it always results in a matching that does so.

Note that fairness implies feasible elimination of justified envy.

The following result shows that we can always (weakly) improve a matching that fails to feasibly eliminate justified envy:

Proposition 1. *For any feasible matching μ that fails to feasibly eliminate justified envy, there exists another feasible (and fairer) matching δ that (weakly) reduces the collective time to completion i.e. $\sum_{t \in T} \tau_t(\underline{s}_t(\delta)) \leq \sum_{t \in T} \tau_t(\underline{s}_t(\mu))$.*

The proof is given in the appendix.

Proposition 1 yields an immediate result. Feasible elimination of justified envy is a *necessary* condition for time minimization:

Corollary 2. *If a matching μ is time minimizing then μ feasibly eliminates justified envy. Similarly, if a matching mechanism φ is time minimizing, then φ feasibly eliminates justified envy.*

2.3.2 Feasible Non-Wastefulness

We define Feasible Non-Wastefulness below:

Definition 10. For any matching μ, a worker i can **claim a seat** in task t if $t \succ_i \mu(i)$ and $|\mu(t)| < q_t$.

Definition 11. A matching μ is **feasibly non-wasteful** if for any worker i that claims a seat in task t, it must be the case that $|\mu(\mu(i))| = 1$.

This definition implies that if worker i prefers task t and if task t still has excess capacity **and** if task $\mu(i)$ has more than one worker assigned to it then i should be moved to t (Fragiadakis et al., 2016). Clearly if a matching μ is non-wasteful, it is feasibly non-wasteful. Interestingly enough, if a matching μ is feasibly non-wasteful then it also feasibly eliminates justified envy:

Proposition 2. *Consider an arbitrary matching μ. If μ is feasibly non-wasteful then μ feasibly eliminates justified envy.*

The proof is given in the Appendix.

3 Two Feasible Mechanisms

Given the conflict between feasibility and stability, the idea is to see if we can apply mechanisms that yield second best outcomes and see how they perform in terms of time minimizing. Many mechanisms have been designed in an attempt to satisfy distributional constraints and stability as much as possible. The Artificial Caps Deferred Acceptance (ACDA) algorithm imposes artificial bounds on the capacities of the tasks in such a way that every task gets assigned a worker and applies the WPDA on the new problem with the new caps. The problem is that the ACDA is known to be highly wasteful, yet is fair (see Fragiadakis et al., 2016 and Goto et al., 2017 for a discussion). As it turns out, the mechanisms in question seem to sacrifice either fairness or non-wastefulness in order to satisfy feasibility. We consider two such mechanisms, the Extended Seat Deferred Acceptance

Algorithm (ESDA) and the Multi-Stage Deferred Acceptance Algorithm (MSDA), both developed by Fragiadakis et al. (2016).

We attempt to determine whether the ESDA or the MSDA performs better when it comes to the time taken to complete the tasks. In particular, we look at whether or not the ESDA always results in a matching that yields a faster time to completion than the MSDA. Intuitively, this would seem to be the case, seeing how the MSDA is not fair, which implies that there is at least one task and worker pair such that the task is more preferred by the worker to his current matching and that the worker has a higher priority for the task (which implies that they weakly perform the task at a faster rate) whereas the ESDA always leads to a fair matching. We attempt to see if this conjecture is true. In regards to the notion of time minimization, since the ESDA and the MSDA are derivatives of the WPDA (in fact they are equal to the WPDA when minimum quotas are removed), they are not time minimizing (Example 2.1 shows this).

3.1 The Extended Seat Deferred Acceptance Algorithm

The ESDA works by first dividing the capacities of the tasks (schools in the original setting of Fragiadakis et al., 2016) into two groups. Regular seats, which is equal to the minimum quotas of the tasks (in this case 1 for each task) and extended seats which is equal to the difference between the minimum and maximum quotas. The algorithm then proceeds to apply the WPDA on this "extended market."

3.1.1 The Algorithm

We simplify the analysis by considering only the associated problem $(I, T, \succ_I, \succ_T, Q, P)$[5] and assume that the priority schedules were generated by some compatibility scores. We follow Fragiadakis et al. (2016) in the exposition. Notation has been changed in accordance to our setting.

Consider the associated 2-sided matching problem $(I, T, \succ_I, \succ_T, Q, P)$ and define the "extended problem" $(I, \tilde{T}, \tilde{\succ}_I, \tilde{\succ}_{\tilde{T}}, \tilde{Q})$ as follows:

- The set of workers remains the same.

[5]Here, $P = (1, 1, \ldots, 1)$ is the vector of minimum quotas.

- For each task t, divide it into two "smaller" tasks. The *standard task* t with maximum quota $\tilde{q}_t = p_t$ and the *extended task* t^* with maximum quota $\tilde{q}_{t^*} = q_t - p_t$. The priority schedule for the standard and extended tasks are the same as the original task: $\succ_t = \tilde{\succ}_t = \tilde{\succ}_{t^*}$. The set of tasks $\tilde{T} = T \cup T^*$ and the vector of maximum quotas follows similarly.

- Worker preferences are extended as follows: Place task t_j^* immediately after task t_j i.e if worker i has preferences $t_j \succ_i t_k \succ_i t_m$, the extended preferences are $t_j \tilde{\succ}_i t_j^* \tilde{\succ}_i t_k \tilde{\succ}_i t_k^* \dots$

- Let $e = |I| - \sum_{t \in T} p_t$ be the number of workers in excess of the minimum requirement to satisfy all minimum quotas.

The algorithm proceeds as follows:

For each extended task t^* fix $\overline{q}_{t^*}$ such that $\overline{q}_{t^*} \leq \tilde{q}_{t^*}$ and $\sum_{t^* \in T^*} \overline{q}_{t^*} \leq e$. Fix an ordering of the extended tasks. Without loss of generality, suppose this ordering is $\{t_1^*, t_2^*, \dots, t_T^*\}$. Let $\tilde{\mu}$ be the matching obtained in the extended problem.

1) For all $i \in I$, let $\tilde{\mu}(i) = \emptyset$.

2) Choose a worker i who is currently not tentatively assigned to any task. If no such worker exists, terminate the algorithm.

3) Let worker i apply to their most preferred task $\tilde{t} \in \tilde{T}$ according to $\tilde{\succ}_i$ that has not rejected him/her.

   a) If $\tilde{t}$ is a standard task, let task $\tilde{t}$ choose the top $\tilde{q}_{\tilde{t}}$ workers according to its priority list $\tilde{\succ}_{\tilde{t}}$ among the workers who have applied to $\tilde{t}$ and have not been rejected by it. Reject all remaining workers and return to Step 2.

   b) If $\tilde{t}$ is an extended task, proceed to step 4.

4) For *all* extended tasks t^*, define I_{t^*} to be the set of workers tentatively assigned to task t^* (i.e. workers who applied to task t^* but have not yet been rejected by it). Let each task t^* choose the top $\overline{q}_{t^*}$ workers in I_{t^*} based on its priority schedule $\tilde{\succ}_{t^*}$. Define this set of workers to be I'_{t^*}. Set $j = 1$:

a) If either the number of workers assigned across all extended tasks is equal to e or if for all extended tasks t^*, the number of workers chosen thus far equals $\min\{\tilde{q}_{t^*}, |I_{t^*}|\}$, then reject all remaining workers not chosen by any extended school and return to step 2.

b) Otherwise, let t_j^* choose its most prioritized worker in I_t^* that has yet to be chosen as long as the number of workers chosen so far is strictly less than the actual capacity $\tilde{q}_{t^*}$. If $j \leq |T|$, increase j by 1. If $j = |T|$, set $j = 1$. Go to step 4a).

The algorithm results in a matching $\tilde{\mu}$. The resulting matching μ for the original problem is obtained as follows: If $\tilde{\mu}(i) = t$ or $\tilde{\mu}(i) = t^*$ then set $\mu(i) = t$.

3.1.2 Properties of ESDA

The ESDA has been shown to satisfy certain nice properties. Fragiadakis et al. (2016) show that the ESDA is strategy-proof and fair but fails to satisfy non-wastefulness.

Theorem 3. *(Fragiadakis et al., 2016, Theorem 3.1) The ESDA mechanism is:*

(i) Strategy-proof;

(ii) Fair

Furthermore, since ESDA is fair, it also feasibly eliminates justified envy:

Corollary 3. *ESDA feasibly eliminates justified envy.*

3.2 The Multi-Stage Deferred Acceptance Algorithm

The Multi-stage Deferred Acceptance Algorithm (MSDA) works by first reserving a set number of workers, enough to fill any minimum quotas. The remaining workers are assigned using the WPDA and then the minimum quota seats remaining are calculated. The algorithm ends when all workers are assigned tasks.

3.2.1 The Algorithm

We follow Fragiadakis et al. (2016). The notation and the steps have been modified to suit our setting. Consider the associated 2-sided matching problem $(I, T, \succ_I, \succ_T, Q, P)$. The algorithm relies upon a **precedence list** $\succ_{PL}$ set exogenously over the workers. Without loss of generality, suppose $i_1 \succ_{PL} i_2 \succ_{PL} \ldots \succ_{PL} i_I$. Initialize the algorithm by setting $R^0 = I$, $p_t^1 = p_t$, $q_t^1 = q$ for all $t \in T$. Define $r^1 = \sum_{t \in T} p_t$ to be the number of reserved workers in R^1. The algorithm proceeds as follows:

Stage $n \geq 1$

1) Set $R^n = \{i_{I-r^n+1}, i_{I-r^n+2}, \ldots i_I\}$ i.e R^n is the set of r^n workers with the lowest ranking in the precedence list.

   (a) If $R^{n-1} \setminus R^n \neq \emptyset$ then apply the WPDA on the workers in $R^{n-1} \setminus R^n \neq \emptyset$ with maximum quotas for the tasks given by q_t^n for all $t \in T$.

   (b) Otherwise, apply the worker proposed on workers in R^n with maximum quotas given by p_t^n.

2) Define μ^n to be the matching induced by 1). Remove all assigned workers. If all workers have been assigned a task, end the algorithm. Otherwise, proceed to step 3.

3) Update the quotas for the tasks as follows:

   i) $q_t^{n+1} = q_t^n - |\mu^n(t)|$.

   ii) $p_t^{n+1} = \max\{p_t^n - |\mu^n(t)|, 0\}$

   iii) $r^{n+1} = \sum_{t \in T} p_t^{n+1}$

   Then proceed to stage $n + 1$.

Suppose the algorithm terminates after step $\tilde{N}$. The final matching is given by $\mu(t) = \cup_{n=1}^{\tilde{N}} \mu^n(t)$ for all tasks t and $\mu(i) = \mu^{n^i}(i)$ where n^i is the stage at which worker i was assigned a task t.

3.2.2 Properties of MSDA

The MSDA is strategy-proof and feasibly non-wasteful. It is not fair. However, it is shown to satisfy a weaker notion of fairness.

Theorem 4. *(Fragiadakis et al., 2016, Theorem 4.2)*

(i) MSDA is strategy-proof

(ii) *MSDA is (feasibly) non-wasteful*

Proof. See Fragiadakis et al. (2016). *Q.E.D*

Corollary 4. *MSDA feasibly eliminates justified envy.*

The proof is in the appendix.

4 A Special Case

Consider the setting with $|I| = |T|$ and $q_t = 1$ for all tasks $t \in T$. Essentially, the number of workers is equal to the number of tasks and all tasks can accommodate at most one worker each. Then the *associated* problem $(I, T, \succ_I, \succ_T, Q)$ is just the time minimization assignment problem analog to the stable marriage problem (Gale and Shapley, 1962). We show two results that pertain to this simple setting:

Proposition 3. *Consider the setting with $|I| = |T|$ and $q_t = 1$ for all tasks $t \in T$. Let μ^{ESDA}, μ^{MSDA} and μ^{WPDA} be the matchings obtained under the ESDA, the MSDA and the WPDA respectively. Then $\mu^{ESDA} = \mu^{MSDA} = \mu^{WPDA}$.*

Proposition 4 shows that the ESDA, MSDA and WPDA all yield the same matching and thus are stable (i.e fair and non-wasteful)[6] and feasible.

Interestingly enough, the TPDA reigns supreme when it comes to time to completion.

Proposition 4. *Consider the setting with $|I| = |T|$ and $q_t = 1$ for all tasks $t \in T$. Let μ^{TPDA} be the matching obtained by the Task Proposed Deferred Acceptance algorithm and let δ be any other stable matching. Then $\sum_{t \in T} \tau_t(\underline{s}_t(\mu^{TPDA})) \leq \sum_{t \in T} \tau_t(\underline{s}_t(\delta))$.*

Since the WPDA (and by Proposition 4 the ESDA and MSDA) are stable, we get the following result.

Corollary 5. *Consider the setting with $|I| = |T|$ and $q_t = 1$ for all tasks $t \in T$. Then the TPDA yields a matching that results in a (weakly) faster time to completion as compared to the ESDA and the MSDA.*

However, the drawback of the TPDA is that it is **not** strategy-proof.

[6] In this special case, stability just collapses to fairness. A matching μ is stable if and only if there exists no blocking pair (i, t).

5 The General Case

While we show positive results for the simple case with one-to-one matching, these results do not extend to a more general setting. While the TPDA appears to be a primary candidate for a time minimizing matching mechanism, it turns out that it may not even be feasible. The following example illustrates some key failures of standard theory, as well as some drawbacks of using the aforementioned feasible matching mechanisms. Details on the procedure to obtain matchings under ESDA and MSDA are relegated to the appendix.

5.1 An Illustrative Example

Consider the following time minimization assignment problem with 5 workers $I = \{A, B, C, D, E\}$ and 3 tasks $T = \{x, y, z\}$. Consider 3 possible characteristics $\{c_x, c_y, c_z\}$ with $f(x) = c_x$, $f(y) = c_y$ and $f(z) = c_z$. The preference profile of the workers, their characteristics scores, the induced priority schedule of the tasks as well and their capacities are given below:

	A	B	C	D	E
	x	x	y	y	y
	z	y	x	z	x
	y	z	z	x	z
$s(c_x)$	5	4	3	2	1
$s(c_y)$	5	3	4	1	2
$s(c_z)$	2	1	5	3	4

	x	y	z
	A	A	C
	B	C	E
	C	B	D
	D	E	A
	E	D	B
q	2	3	1

5.1.1 Failure of Standard Mechanisms

The matchings obtained from the WPDA and TPDA algorithms are:

$$\mu^{TPDA} = \mu^{WPDA} = \begin{pmatrix} x & y & z \\ \{A, B\} & \{C, D, E\} & \emptyset \end{pmatrix}$$

Note that both mechanisms yield the *unique* stable matching. However this matching is infeasible since task z has no worker assigned to it. Since feasibility is a necessary condition for time minimization, neither the WPDA or the TPDA are time minimizing.

5.1.2 Comparison of MSDA and ESDA

The ESDA and the MSDA result in the following matchings:

$$\hat{\mu}^{ESDA} = \begin{pmatrix} x & y & z \\ \{A, B\} & \{C, E\} & \{D\} \end{pmatrix} \qquad \hat{\mu}^{MSDA} = \begin{pmatrix} x & y & z \\ \{A, B\} & \{C, D\} & \{E\} \end{pmatrix}$$

It is clear that both mechanisms are wasteful (in both cases, task y has a vacancy and the worker assigned to task z prefers y to z). Furthermore, the MSDA is not fair since (E, y) form a blocking pair ($E \succ_y D$ and $y \succ_E z$). However, both matchings are feasible and thus outperform the WPDA and TPDA in time to completion. Unfortunately, one cannot conclusively determine if the ESDA outperforms the MSDA in this regard or vice versa. The time taken to complete the tasks under both matchings is:

$$\hat{\tau}^{ESDA} = \tau_x(4) + \tau_y(2) + \tau_z(3) \qquad \hat{\tau}^{MSDA} = \tau_x(4) + \tau_y(1) + \tau_z(4)$$

Comparing the two, we find that $\hat{\tau}^{ESDA} - \hat{\tau}^{MSDA} = \underbrace{(\tau_y(2) - \tau_y(1))}_{<0} + \underbrace{(\tau_z(3) - \tau_z(4))}_{>0} \lesseqgtr 0$.

5.2 Discussion

One thing to point out is that the preceding analysis imposes little to no structure on the matching environment. In particular, the only restriction we impose on τ is that it is strictly decreasing. As the result in section 5.1.2 implies, the characteristics of τ matters. Suppose that $\tau_x = \tau_y = \tau_z$ and are convex. Then it becomes apparent that $\hat{\tau}^{ESDA} < \hat{\tau}^{MSDA}$ which implies that the ESDA performs better than the MSDA when it comes to time to completion and vice versa if τ is concave.

Furthermore, our model assumes that only the weakest person is relevant when it comes to the time taken to complete a particular task. Another possible approach to modeling τ is to base it off of some index computed using the scores of *all* the workers assigned a particular task. One particular index could be computed by taking the *average* measure of all the scores of workers assigned to a particular task. However, under this setting, the result that feasible elimination of justified envy being a necessary condition for time minimization does not hold. Intuitively, this is due to the fact that worker i in the feasible blocking pair (i, t') may have a higher than average score in his assigned task $t = \mu(i)$ and assigning him to task t' could result in a slower time to completion for t under the new matching. We intend to explore these possibilities in future work.

6 Conclusion

We present a mechanism design approach to the Time Minimization Assignment Problem studied in Operations Research. Workers have preferences over tasks and tasks have priorities over workers that stem from their compatibility scores. We show how the benchmark mechanism (the Gale-Shapley deferred acceptance algorithm) fails to achieve a stable and time minimizing matching and consider two mechanisms that always yield a feasible and strategy-proof matching, yet fail to satisfy stability. We compare the two mechanisms and find that in general, we cannot conclusively declare that one outperforms the other in terms of time taken to complete the tasks.

Appendix A Proofs

This section provides the proofs for the main results.

Proposition 1. *For any feasible matching μ that fails to feasibly eliminate justified envy, there exists another feasible (and fairer) matching δ that (weakly) reduces the collective time to completion i.e. $\sum_{t\in T} \tau_t(\underline{s}_t(\delta)) \leq \sum_{t\in T} \tau_t(\underline{s}_t(\mu))$.*

Proof. Consider a matching μ that fails to feasibly eliminate justified envy. Then there exists a feasible blocking pair i.e there exists a pair $(i^*, \hat{t})$ such that $\hat{t} \succ_{i^*} \mu(i^*)$, $i^* \succ_{\hat{t}} \hat{i}$ for some $\hat{i} \in \mu(\hat{t})$. Furthermore, the matching μ' such that $\mu'(i^*) = \hat{t}$ and $\mu'(i') = \mu(i')$ for all $i' \neq i^*$ is feasible. Denote $\mu(i^*) = t^*$ and let $\delta = \mu'$. Since δ is feasible, it must be the case that $1 \leq |\delta(t)| \leq q_t$ for all $t \in T$. Furthermore, it must be the case that for all $t \in T \setminus \{\hat{t}, t^*\}$, we have that $\mu(t) = \delta(t)$. Thus for all $t \in T \setminus \{\hat{t}, t^*\}$, $\underline{s}_t(\delta) = \underline{s}_t(\mu) \Rightarrow \tau_t(\underline{s}_t(\delta)) = \tau_t(\underline{s}_t(\mu))$.

Now consider $\hat{t}$. We show that $\tau_{\hat{t}}(\underline{s}_{\hat{t}}(\delta)) = \tau_{\hat{t}}(\underline{s}_{\hat{t}}(\mu))$. Note first that $\delta(\hat{t}) = \mu(\hat{t}) \cup \{i^*\}$. Furthermore, $i^* \succ_{\hat{t}} \hat{i}$ for some $\hat{i} \in \mu(\hat{t})$. This implies that $s^{i^*}(f(\hat{t})) > s^{\hat{i}}(f(\hat{t}))$. Also, $s^{\hat{i}}(f(\hat{t})) \geq \underline{s}_{\hat{t}}(\mu)$. Thus $s^{i^*}(f(\hat{t})) > \underline{s}_{\hat{t}}(\mu)$ which implies that $\underline{s}_{\hat{t}}(\delta) = \underline{s}_{\hat{t}}(\mu)$ and therefore $\tau_{\hat{t}}(\underline{s}_{\hat{t}}(\delta)) = \tau_{\hat{t}}(\underline{s}_{\hat{t}}(\mu))$.

Now consider task t^*. We show that $\tau_{t^*}(\underline{s}_{t^*}(\delta)) \leq \tau_{t^*}(\underline{s}_{t^*}(\mu))$. First note that $\delta(t^*) = \mu(t^*) \setminus \{i^*\}$. There are 2 cases:

Case 1: Worker i^* is the "weakest link" in task t^* under matching μ. Then $s^{i^*}(f(t^*)) = \underline{s}_{t^*}(\mu)$. This implies that (since there are no ties) $\underline{s}_{t^*}(\delta) > \underline{s}_{t^*}(\mu)$ and therefore $\tau_{t^*}(\underline{s}_{t^*}(\delta)) < \tau_{t^*}(\underline{s}_{t^*}(\mu))$.

Case 2: Worker i^* is not the "weakest link" in task t^* under matching μ. Then $s^{i^*}(f(t^*)) > \underline{s}_{t^*}(\mu)$ which implies that $\underline{s}_{t^*}(\delta) = \underline{s}_{t^*}(\mu)$ and therefore $\tau_{t^*}(\underline{s}_{t^*}(\delta)) = \tau_{t^*}(\underline{s}_{t^*}(\mu))$.

Thus under matching δ, we have that $\sum_{t \in T} \tau_t(\underline{s}_t(\delta)) \leq \sum_{t \in T} \tau_t(\underline{s}_t(\mu))$. Furthermore, this matching is fairer since we have one less blocking pair. *Q.E.D*

Proposition 2. *Consider an arbitrary matching μ. If μ is feasibly non-wasteful then μ feasibly eliminates justified envy.*

Proof. Suppose that μ is feasibly non-wasteful. Suppose on the contrary that μ fails to feasibly eliminate justified envy. Then there exists a pair (i, t) such that i) $t \succ_i \mu(i)$, ii) for some $i' \in \mu(t)$, $i \succ_t i'$ and iii) the matching δ such that $\delta(i) = t$ and $\delta(i') = \mu(i')$ for all $i' \neq i$ is feasible. Note that iii) implies that for all tasks $t' \in T$, $1 \leq |\delta(t')| \leq q_{t'}$. Since $\delta(t) = \mu(t) \cup \{i\}$ and $|\delta(t)| \leq q_t$, this implies that iv) $|\mu(t)| < q_t$. Also, $\delta(\mu(i)) = \mu(i) \setminus \{i\}$ and $|\delta(\mu(i))| \geq 1$. This implies that v) $|\mu(i)| > |\delta(\mu(i))| \geq 1$. But i), iv) and v) contradict the fact that μ is feasibly non-wasteful. *Q.E.D*

Corollary 4. *The MSDA feasibly eliminates justified envy.*

Proof. By Theorem 4, MSDA is feasibly non-wasteful. By Proposition 2, the MSDA feasibly eliminates justified envy. *Q.E.D*

Proposition 3. *Consider the setting with $|I| = |T|$ and $q_t = 1$ for all tasks $t \in T$. Let μ^{ESDA}, μ^{MSDA} and μ^{WPDA} be the matchings obtained under the ESDA, the MSDA and the WPDA respectively. Then $\mu^{ESDA} = \mu^{MSDA} = \mu^{WPDA}$.*

Proof. We first show that $\mu^{WPDA} = \mu^{ESDA}$. Consider the extended market $(I, \tilde{T}, \tilde{\succ}_I, \tilde{\succ}_T, \tilde{Q})$. First note that for any extended task t^*, $\tilde{q}_{t^*} = 0$ since in the original market $q_t = p_t = 1$. Furthermore, since the number of workers is equal to the number of tasks, $e = |I| - \sum_{t \in T} p_t = |I| - |T| = 0$. Thus if any worker at any stage of the ESDA applies to an extended task t^*, they will be rejected and will apply to the standard task next on their preference list. This essentially yields the same result as the WPDA.

We now show that $\mu^{MSDA} = \mu^{WPDA}$. To see this, first note that for all tasks $t \in T$, $q_t = p_t = 1$. Furthermore, $|I| = |T|$ which implies that $r^1 = \sum_{t \in T} p_t = |I|$. Then, in stage 1 of the algorithm, $R^0 = R^1 = I$ which implies that $R^0 \setminus R^1 = \emptyset$. By Step 1 b) of the algorithm, this means that we apply the WPDA on I with quotas $p_t = 1 = q_t$ for all t. The algorithm ends in Stage 1. Thus $\mu^{MSDA} = \mu^{WPDA}$. *Q.E.D*

Proposition 4. *Consider the setting with $|I| = |T|$ and $q_t = 1$ for all tasks $t \in T$. Let μ^{TPDA} be the matching obtained by the Task Proposed Deferred Acceptance algorithm and let δ be any other stable matching. Then $\sum_{t \in T} \tau_t(\underline{s}_t(\mu^{TPDA})) \leq \sum_{t \in T} \tau_t(\underline{s}_t(\delta))$.*

Proof. The TPDA yields the Task Optimal stable matching. Furthermore, this matching is feasible since $|I| = |T|$ and $p_t = q_t = 1$ for all t. Thus, for all $t \in T$, we have that $\mu^{TPDA}(t) \succsim_t \delta(t)$ where δ is any other stable matching. Since in any stable matching μ, $|\mu(t)| = 1$ for all t, this implies that $s^{\mu^{TPDA}(t)}(f(t)) \geq s^{\delta(t)}(f(t))$. Furthermore, for all $t \in T$, $\underline{s}_t(\mu) = s^{\mu(t)}(f(t))$. Thus for all tasks t, we have that $\underline{s}_t(\mu^{TPDA}) \geq \underline{s}_t(\delta)$ which implies that for all tasks t, $\tau_t(\underline{s}_t(\mu^{TPDA})) \leq \tau_t(\underline{s}_t(\delta))$. Thus, we have that $\sum_{t \in T} \tau_t(\underline{s}_t(\mu^{TPDA})) \leq \sum_{t \in T} \tau_t(\underline{s}_t(\delta))$. *Q.E.D*

Appendix B Details on Example

Consider the following time minimization assignment problem with 5 workers $I = \{A, B, C, D, E\}$ and 3 tasks $T = \{x, y, z\}$. Consider 3 possible characteristics $\{c_x, c_y, c_z\}$ with $f(x) = c_x$, $f(y) = c_y$ and $f(z) = c_z$. The preference profile of the workers, their characteristics scores, the induced priority schedule of the tasks as well and their capacities are given below:

	A	B	C	D	E
	x	x	y	y	y
	z	y	x	z	x
	y	z	z	x	z
$s(c_x)$	5	4	3	2	1
$s(c_y)$	5	3	4	1	2
$s(c_z)$	2	1	5	3	4

	x	y	z
	A	A	C
	B	C	E
	C	B	D
	D	E	A
	E	D	B
q	2	3	1

B.1 The matching obtained under ESDA

The extended market $(I, \tilde{T}, \tilde{\succsim}_I, \tilde{\succsim}_T, \tilde{Q})$ is given below:

A	B	C	D	E
x	x	y	y	y
x^*	x^*	y^*	y^*	y^*
z	y	x	z	x
z^*	y^*	x^*	z^*	x^*
y	z	z	x	z
y^*	z^*	z^*	x^*	z^*

	x	x^*	y	y^*	z	z^*
	A	A	A	A	C	C
	B	B	C	C	E	E
	C	C	B	B	D	D
	D	D	E	E	A	A
	E	E	D	D	B	B
$\tilde{q}$	1	1	1	2	1	0

Set $e = |I| - \sum_t p_t = 2$, $\overline{q}_t^* = 0$ for all tasks t and fix the ordering of tasks as x^*, y^*, z^*. The algorithm proceeds as follows:

Stage 1: Workers A and B apply to task x and the remaining workers apply to task y. The tasks choose the top $\tilde{q}$ workers on their priority list. Thus workers B, D and E are rejected.

Stage 2: Worker B applies to x^* and D and E apply to y^*. The extended tasks then choose their top prioritized worker one by one in the order x^*, y^*, z^*. Then x^* is tentatively assigned to B and y^* is tentatively assigned to E. At this point, 2 workers have been tentatively assigned to an extended task so D is rejected.

Stage 3: Worker D then applies to task z and is assigned. The algorithm terminates.

The final matching is:

$$\mu^{ESDA} = \begin{pmatrix} x & x^* & y & y^* & z & z^* \\ A & B & C & E & D & \varnothing \end{pmatrix} = \begin{pmatrix} x & y & z \\ \{A, B\} & \{C, E\} & \{D\} \end{pmatrix}$$

Note that this matching is fair (there exists no blocking pair) and also, this matching is wasteful (D prefers y to z and y still has excess capacity). Furthermore, the time taken to complete all tasks is given by $\tau^{ESDA} = \tau_x(4) + \tau_y(2) + \tau_z(3)$.

B.2 The matching obtained under MSDA

Consider the precedence list $A \succ_{PL} B \succ_{PL} C \succ_{PL} D \succ_{PL} E$. The algorithm proceeds as follows:

Stage 0: Set $R^0 = I$, $p^1_x = p^1_y = p^1_z = 1$ and $q^1_x = 2$, $q^1_y = 3$ and $q^1_z = 1$. Set $r^1 = 3$.

Stage 1: The lowest 3 workers in the precedence list are $R^1 = \{C, D, E\}$. Then $R^0 \setminus R^1 = \{A, B\}$. Applying the worker proposed DA on workers A and B with maximum quotas q^1_t yields the matching μ^1 such that $\mu^1(A) = \mu^1(B) = x$. Removing the assigned workers from the market, we update the minimum and maximum quotas and obtain $q^2_x = 0$, $q^2_y = 3$, $q^2_z = 1$ and $p^2_x = 0$, $p^2_y = p^2_z = 1$ and $r^2 = 2$.

Stage 2: The lowest 2 workers on the precedence list are $R^2 = \{D, E\}$. Then $R^1 \setminus R^2 = \{C\}$. Applying the worker proposed DA on worker C with maximum quotas q^2_t yields the matching $\mu^3(C) = y$. Removing C and updating the quotas yields $q^3_x = 0$, $q^3_y = 2$, $q^3_z = 1$ and $p^3_x = p^3_y = 0$, $p^3_z = 1$ and $r^3 = 1$.

Stage 3: The last worker on the precedence list is $R^3 = \{E\}$ so $R^2 \setminus R^3 = \{D\}$. Applying the worker proposed DA on worker D with maximum quotas q^3_t yields the matching $\mu^3(D) = y$. Removing D and updating the quotas yields $q^4_x = 0$, $q^4_y = 1$, $q^4_z = 1$ and $p^4_x = p^4_y = 0$, $p^4_z = 1$ and $r^4 = 1$.

Stage 4: The last worker on the precedence list is $R^4 = \{E\}$ and $R^3 \setminus R^4 = \emptyset$. We apply the worker proposed DA on worker E with maximum quotas p^4_t which yields the matching $\mu^4(E) = z$. All workers have been assigned so terminate the algorithm.

The resulting matching μ^{MSDA} is:

$$\mu^{MSDA} = \begin{pmatrix} x & y & z \\ \{A, B\} & \{C, D\} & \{E\} \end{pmatrix}$$

Notice that the matching is not fair: (E, y) form a blocking pair. However, it does feasibly eliminate justified envy. Note that it is also wasteful since $|\mu^{MSDA}(y)| < q_y$ and $y \succ_E \mu^{MSDA}(E)$. The time taken to complete the tasks is given by: $\tau^{MSDA} = \tau_x(4) + \tau_y(1) + \tau_z(4)$.

References

Abdulkadiroğlu, Atila (2005) "College admissions with affirmative action," *International Journal of Game Theory*, Vol. 33, No. 4, pp. 535–549.

Abdulkadiroğlu, Atila and Tayfun Sönmez (2003) "School Choice: A Mechanism Design Approach," *American Economic Review*, Vol. 93, No. 3, pp. 729–747.

——— (2013) "Matching Markets: Theory and Practice," in *Advances in Economics and Econometrics*, Chap. 1, pp. 3–47.

Arora, Shalini and M.C. Puri (1998) "A variant of time minimizing assignment problem," *European Journal of Operational Research*, Vol. 110, No. 2, pp. 314 – 325, EURO Best Applied Paper Competition.

Balinski, Michel and Tayfun Sönmez (1999) "A Tale of Two Mechanisms: Student Placement," *Journal of Economic Theory*, Vol. 84, No. 1, pp. 73 – 94.

Chauvet, F., J.-M. Proth, and A. Soumare (2000) "The simple and multiple job assignment problems," *International Journal of Production Research*, Vol. 38, No. 14, pp. 3165–3179.

Erdil, Aytek and Haluk Ergin (2017) "Two-sided matching with indifferences," *Journal of Economic Theory*, Vol. 171, pp. 268 – 292.

Ergin, Haluk I. (2002) "Efficient Resource Allocation on the Basis of Priorities," *Econometrica*, Vol. 70, No. 6, pp. 2489–2497.

Fragiadakis, Daniel, Atsushi Iwasaki, Peter Troyan, Suguru Ueda, and Makoto Yokoo (2016) "Strategyproof Matching with Minimum Quotas," *ACM Transactions on Economics and Computation.*, Vol. 4, No. 1, pp. 6:1–6:40.

Gale, D. and L. S. Shapley (1962) "College Admissions and the Stability of Marriage," *The American Mathematical Monthly*, Vol. 69, No. 1, pp. 9–15.

Goto, Masahiro, Fuhito Kojima, Ryoji Kurata, Akihisa Tamura, and Makoto Yokoo (2017) "Designing Matching Mechanisms under General Distributional Constraints," *American Economic Journal: Microeconomics*, Vol. 9, No. 2, pp. 226–62.

Jain, Ekta, Kalpana Dahiya, Anuj Sharma, and Vanita Verma (2018) "An improved algorithm for two stage time minimization assignment problem," *Journal of Combinatorial Optimization*.

Kamada, Yuichiro and Fuhito Kojima (2015) "Efficient Matching under Distributional Constraints: Theory and Applications," *American Economic Review*, Vol. 105, No. 1, pp. 67–99.

——— (2017) "Recent developments in matching with constraints," *American Economic Review: Papers and Proceedings*, Vol. 107, No. 5, pp. 200–204.

——— (2018) "Stability and strategy-proofness for matching with constraints: A necessary and sufficient condition," *Theoretical Economics*, Vol. 13, No. 2, pp. 761–793.

Kojima, Fuhito (2012) "School choice: Impossibilities for affirmative action," *Games and Economic Behavior*, Vol. 75, No. 2, pp. 685 – 693.

——— (2017) *Recent Developments in Matching Theory and Their Practical Applications*, Vol. 1 of Econometric Society Monographs, p. 138–175: Cambridge University Press.

Roth, Alvin E., Tayfun Sönmez, and M. Utku Ünver (2004) "Kidney Exchange," *The Quarterly Journal of Economics*, Vol. 119, No. 2, pp. 457–488.

Sonia and M.C. Puri (2008) "Two-stage time minimizing assignment problem," *Omega*, Vol. 36, No. 5, pp. 730 – 740.