Unbalanced Random Matching Markets with Partial Preferences

Aditya Potukuchi
Department of Electrical Engineering and Computer Science, York University, Toronto, Canada

Shikha Singh
Department of Computer Science, Williams College, Williamstown, MA, USA

Abstract

Properties of stable matchings in the popular random-matching-market model have been studied for over 50 years. In a random matching market, each agent has complete preferences drawn uniformly and independently at random. Wilson (1972), Knuth (1976) and Pittel (1989) proved that in balanced random matching markets, the proposers are matched to their $\ln n$th choice on average. In this paper, we consider competitive markets with n jobs and $n+k$ candidates, and partial lists where each agent only ranks their top d choices. Despite the long history of the problem, the following fundamental question remains unanswered for these generalized markets: *what is the tight threshold on list length d that results in a perfect stable matching with high probability?* In this paper, we answer this question exactly – we prove a sharp threshold $d_0 = \ln n \cdot \ln \frac{n+k}{k+1}$ on the existence of perfect stable matchings when $k = o(n)$. That is, we show that if $d < (1-\epsilon)d_0$, then no stable matching matches all jobs; moreover, if $d > (1+\epsilon)d_0$, then all jobs are matched in every stable matching with high probability. This bound improves and generalizes recent results by Kanoria, Min and Qian (2021).

Furthermore, we extend the line of work studying the effect of imbalance on the expected rank of the proposers (termed the "stark effect of competition"). We establish the regime in unbalanced markets that forces this stark effect to take shape in markets with partial preferences.

2012 ACM Subject Classification Theory of computation → Randomness, geometry and discrete structures

Keywords and phrases stable matching, probabilistic method, Gale-Shapley algorithm

Digital Object Identifier 10.4230/LIPIcs.ICALP.2025.125

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2402.09667

Funding *Aditya Potukuchi*: Research supported by an NSERC Discovery Grant.
Shikha Singh: Research supported in part by NSF CCF 1947789.

Acknowledgements We thank anonymous reviewers for their valuable feedback to help improve this paper.

1 Introduction

The stable matching problem is a classic problem in computer science, economics and mathematics. In the standard and perhaps antiquated description of the problem as the "stable marriage problem", the market instance consists of n men and n women and a complete preference ordering of size n for each, where every man ranks all the women and vice versa. A matching is considered to be *stable* if no pair of agents would prefer to break off from their current match and match with each other instead. Stability has proved to be an extremely robust notion in the success of centralized two-sided matching markets [26]. The *deferred-acceptance (DA) algorithm* by Gale and Shapley [9] to compute stable matchings

has proved extremely influential and has since been used in many real-life matching markets such as matching doctors to hospitals, students to schools, and applicants to jobs; see for example [27, 1, 2, 8].
Initiated by [28, 17], there has been over 50 years of fascinating research to understand the properties of stable matchings in random-matching markets, where the preferences of the agents are generated uniformly and independently at random. Typically, the markets studied are balanced (equal number of agents on both sides) with complete preference lists.

The early results [28, 17, 23] were primarily motivated by the average complexity of the aforementioned DA algorithm. In the process, they uncovered a glimpse into other interesting market phenomena. For instance, if there are n candidates applying to n jobs, the average candidate is matched to their $\ln n$th preferred job whereas the average job is matched to its $(n/\ln n)$th preferred candidate. This quantifies and highlights the stark (and from a real-world perspective, rather untenable) advantage of the candidates over the jobs.

It turns out that this advantage is an artifact of the (unrealistic in some labour markets) assumption of having an equal number of jobs and candidates. Indeed, a particularly remarkable result by [5] shows that having even one extra candidate (i.e., $n+1$ candidates and n jobs) reverses the situation entirely. The average matched candidate gets their $\Omega(n/\ln n)$th preferred job and the average job is matched to their $O(\ln n)$th preferred candidate – as if the jobs were "applying" to the candidates. In general, [5] show that if there were $n+k$ candidates, then the average (matched) candidate gets assigned to their $\Omega(n/\ln(n/k))$th preferred job. This was dubbed the "stark effect of competition".

Of course, all of the discussion above assumes that each candidate is required to submit a full ranking of *every* job. This is infeasible even in mildly-large labor markets such as the National Residency Matching Program (NRMP), college admissions, and school choice (which all rely on the DA algorithm) where there are more than 1000 jobs ("positions"). In fact, the NRMP allows each candidate to only provide their top 20 choices.[1] The efficiency of these labor markets are measured by the number of positions filled (every year the NRMP publishes its "fill rate"). Thus, from the point of view of having every position filled, a fundamental market-design question is:

If each candidate and job only rank only their top d choices (uniformly and independently at random), how big does d have to be to ensure that all jobs are matched?

Given that the DA algorithm has been in use for over 70 years, first adopted by NRMP in 1952, it is surprising that this natural question has not been answered.

1.1 Our Results

Our main result answers this question. We say that a stable matching is perfect if every job is matched. We tightly characterize the existence of perfect stable matchings in random matching markets with imbalance and partial preferences.

▶ **Theorem** (Informal Statement of Theorem 6). *Consider a matching market with n jobs and $n+k$ candidates where $k = o(n)$, and each candidate has a uniform-random preference list of length d. Then, there is a threshold $d_0 = \ln n \cdot \ln \frac{n+k}{k+1}$ such that,*

- *if $d > (1+o(1))d_0$, then all stable matchings are perfect with high probability, and*
- *if $d < (1-o(1))d_0$, then no stable matching is perfect with high probability.*

[1] Candidates submit 20 ranks for free; each additional rank costs \$30 up to a maximum of 300 [22].

Before we proceed, lets put this in context with respect to recent NRMP data. In 2024, there were about $50,500$ registered applicants, and about $41,500$ positions. So going by the prediction of the above theorem with $k = 9000$, to ensure that all positions are filled, each applicant should submit their top $\ln(41500) \cdot \ln(505000/9000) \approx 19$ preferences. Admittedly there are many factors and decades of refinement that determine the details of systems such as medical residency programs. However this back-of-the-envelope calculation, though admittedly fortuitous, does seem to offer a theoretical explanation as to why (a) the NRMP settled on soliciting about 20 preferences over time, and (b) it is able to successfully fill most positions.

To motivate our second result, we refocus on the aforementioned "stark effect of competition" [5]. It turns out that this stark effect is not observed in many real-life datasets. Kanoria et al [15] proposed that this phenomenon is a consequence of the fact that in these settings, the applicants typically apply to a small subset of the jobs. However, this explanation only holds for a restricted parameter range (when $k < n^{1-\Omega(1)}$ and $d = o(\log^2 n)$). We show a general lower bound on the rank that the average (matched) candidates can obtain.

▶ **Theorem** (Informal Statement of Theorem 12). *Consider a stable matching market with $n + k$ candidates, n jobs and uniform-random preference lists of length d where $k = o(n)$ and $d = \omega(\log n)$. Let $r_{\mathcal{C}}$ denote the expected rank of the best-possible stable partner each candidate can obtain. Then we have $\mathbf{E}[r_{\mathcal{C}}] \geq (1 - o(1))\frac{d}{\ln \frac{n+k}{k+1}}$.*

To put this result in context, we compare it with the general economic insight in [15] that *a matching market exhibits a significant short-side advantage if and only if the number of short side agents who remain unmatched is smaller than the market imbalance.* Note that while it is not a priori known how many agents will end up being unmatched, Theorem 12 provides a more complete picture in terms of known market parameters (n, k and d). In particular, it shows that the stark effect of competition is likely to take shape when the imbalance $k \gg ne^{-\sqrt{d}}$.

One final point worth noting, is that the characteristics of random matching markets seem to hold when preferences between the candidates are correlated. In particular, [4] show that even when the preferences are tiered (based on public scores), the average ranks observed in random markets continue to hold up to constants. This further provides further explanation for why random matching markets have served as a yardstick for understanding and guiding market-design over many decades [17, 16, 24, 15, 5, 6].

1.2 Background and Prior Work

Let us first review the deferred-acceptance algorithm [20, 9]. Given the preference lists of each candidate and job, consider the following procedure in which the candidates "propose" to jobs on their list. At each point, a candidate without a potential match (chosen arbitrarily) proposes to the highest job on their list that they have not been rejected by yet. On receiving a proposal from a candidate c, a job j becomes potentially matched to c if c is the highest ranked among the candidates that have proposed to j so far, otherwise j rejects c. The process goes on until every candidate has either been rejected by every job on their list, or has a potential match. At the end, the potential matches are final.

The DA algorithm has some interesting properties that those unfamiliar may find surprising: (a) it always find a stable matching, and (b) this stable matching results in the best-possible stable partner for all candidates, and the worst-possible stable partner for the jobs. Beyond algorithmic applications, these properties also make DA an important tool in the study of random instances.

The DA algorithm readily generalizes to partial preferences [12]. Notice that if the preference list length d is too small, some candidates/jobs may remain unmatched. Remarkably, the set of unmatched agents stays the same in every stable matching (this is referred to as the Lone Wolf Theorem [21]). Thus, if a perfect stable matching exists, then all stable matchings are perfect.

1.3 Model and Discussion

Before we proceed, we formally define the model. Consider a two-sided matching instance $M_{n,d,k}$ with $n+k$ candidates and n jobs, where $k \geq 0$. An edge (c,j) between a candidate c and job j occurs independently with probability d/n (thus the resulting graph is a bipartite Erdös-Rényi graph $G_{n+k,n,d}$ graph). Each candidate and job have random ranking over their neighbors, chosen uniformly and independently. So for example, $M_{n,n,0}$ is the random stable matching instance with complete preference lists. We abuse notation and use $M_{n,d,k}$ to refer both to the stable matching instance (graph and rankings) and the graph itself.

We refer to d as the **degree** of the underlying random graph. One may easily deduce that for a perfect stable matching, it must hold that $(n+k)d \geq (1-o(1))n \ln n$, since otherwise, $M_{n,d,k}$ has no perfect matching with high probability. As noted before, we show that critical threshold for perfect stable matchings is a bit higher, $\approx \ln n \cdot \ln(n/k)$.

The DA algorithm also naturally motivates another distribution over random stable matching instances, where each candidate chooses a set of d (chosen in advance) jobs along with a uniform ranking independently, and each job ranks the candidates uniformly and independently. This is the model used in past work on partial preferences [13, 14, 15]. We use $M^{\mathcal{C}}_{n,d,k}$ to denote this model, and $M^{\mathcal{J}}_{n,d,k}$ to denote the analogous model in which the jobs choose their top d candidates independently.

As one might expect, the above models are closely related to each other in the sense that some results (particularly, the type we are interested in) often translate between them when $d \gg \ln n$. Indeed, we prove that in this regime, a random instance from one of these models can be "sandwiched" using instances of another model that are close enough; see Lemma 4.

Discussion on Theorem 6

For balanced markets (when $k=0$), the best-known bound for the threshold for the existence of perfect stable matchings was studied by Pittel [24] in the context of the maximum number of proposals any agent makes in a balanced market with complete preferences, which is proved to be $\Theta(\ln^2 n)$. A note at the end of [24] mentions that the sharp threshold of $\ln^2 n$ may be recovered from the results of [25]. However, we are unaware of a self-contained proof of this fact. Thus, our analysis serves as a new combinatorial understanding of the $\ln^2 n$ threshold for balanced markets.

The threshold for unbalanced markets was studied by [15], who prove (among other closely related results), the following:

- If $d = o(\log^2 n)$ and $k = O(n^{1-\epsilon})$, then no perfect stable matching exists w.h.p., and,
- If $d = \Omega(\log^2 n)$ and $d = o(n)$, and $k = o(n)$, all stable matchings are perfect w.h.p.

We improve these results in two significant ways. First, Theorem 6 captures the correct dependence on the imbalance k. In particular, when $k = n/\ln n$, both the above bounds on d are not tight, and the right threshold, as given by Theorem 6 is $\ln n \ln \ln n$. Second, we provide a new and (in our opinion) significantly simpler analysis framework which can be leveraged to establish other properties of interest in these markets. An interesting empirical observation made in [15] is that constants hidden in their bounds are much smaller than 1 as the imbalance gets larger. Thus, the bound in Theorem 6 formally captures this empirical observation.

Discussion on Theorem 12

As mentioned earlier, [5] analyze the effect of competition in stable matching instances with complete preferences, that is, $M_{n,n,k}$.

More recently, [15] focused on the effect of competition as the degree d becomes smaller. In particular, they study stable matching instances $M^{\mathcal{C}}_{n,d,k}$ with *polynomially small imbalance*, that is, $1 \leq k \leq n^{1-\epsilon}$, and show the following dichotomy: (a) when $d = o(\ln^2 n)$, both candidates and jobs are matched to the $\sqrt{d}(1+o(1))$-ranked partner on average and thus there is no effect of competition, and (b) when $d = \Omega(\ln^2 n)$, there is a stark effect of competition – the candidates are matched to their $(d/\ln n)$th ranked job and the jobs to their $\ln n$th ranked candidate (in expectation).

The lower bound of $\sqrt{d}(1+o(1))$ is better than the one in Theorem 12 when $k \ll ne^{-\sqrt{d}}$. Interestingly enough, the regime that they work in, $k = n^{1-\epsilon}$ and $d = o(\log^2 n)$, reflects this and the competition in this regime plays a negligible role. Theorem 12 improves their lower bound of $(d/\ln n)$ for larger imbalance. It would be a natural guess that $k \approx ne^{-\sqrt{d}}$ is where the "stark effect of competition" phenomenon starts to take shape. So, we conjecture that the bound in Theorem 12 is tight when $k \gg ne^{-\sqrt{d}}$.

1.4 Technical Contributions

We develop the following technical tools in our analysis.

First, we compare the proposals in the DA algorithm to a balls-in-bins process. This comparison itself is not new: analyzing the DA as a balls-and-bins or coupon-collector game dates back to the original papers by [28, 17]. We add to this approach by providing some general bounds using simple methods that could simplify similar proofs in the future.

Second, our methods are simple, streamlined, and provide a new analysis framework for this model. To establish whether or not a stable matching exists, we look at which of the two competing forces in the DA algorithm occurs first: each of the n jobs receiving a proposal or $k+1$ candidates exhausting all d of their proposals. We analyze this competition by defining a simple probabilistic game on the "rejection chains" in the DA algorithm. The methods in [15] also study rejection chains, but at a conceptual level, understanding the competing forces seems to offer a fundamentally different approach. For instance, the analysis in [15] requires a bound on the total number of proposals made in the DA algorithm. The aforementioned game lets us circumvent this and isolate the event of interest (the existence of a perfect stable matching).

Finally, we study three closely related random models: the symmetric case $M_{n,d,k}$ which is arguably the most natural to state as it is independent of which side of the market proposes, and the asymmetric models $M^{\mathcal{C}}_{n,d,k}$ and $M^{\mathcal{J}}_{n,d,k}$ in which the candidates and jobs choose preference lists of length d respectively. Switching between these models helps analyze the problem from both sides: in particular, we analyze the candidate-proposing DA on $M^{\mathcal{C}}_{n,d,k}$ and the job-proposing DA on $M^{\mathcal{J}}_{n,d,k}$. We combine these results to obtain a bound for $M_{n,d,k}$ using a "sandwiching lemma" (Lemma 4). We believe that this approach of analyzing the DA on different random stable-matching instances provides new insights as past work has only studied the instance $M^{\mathcal{C}}_{n,d,k}$.

1.5 Additional Related Work

The original DA algorithm [9] was extended to the imbalanced case by McVitie and Wilson [20], and to the case of partial preferences by Gusfield and Irving [12]. Stable matchings have since been extensively studied over 60 years with several books on the topic; we refer to the readers to [18]. We mention closely related results below. For a more comprehensive survey on the history and results on random matching markets, see [19].

Wilson [28] and Knuth [17] reduce the problem of running deferred acceptance in random balanced matching markets with complete lists, $M_{n,n}$, to the coupon-collector problem and use it to compute its average complexity. This technique is used in various works (including this paper), e.g. [15, 4, 5]. [29] showed that the expected number of proposals in the candidate-proposing DA is at most nH_n. Knuth [17] improved this upper bound to $(n-1)H_n + 1$ and proved a matching lower bound of $nH_n - O(\ln^4 n)$. Thus, the candidates side are matched to their $\ln n$th ranked partner in expectation. In contrast, Pittel [23] showed that the receiving side is a lot worse and is matched to their $(n/\ln n)$th partner in expectation.

The model of partial preferences was first considered in [13, 14], who assume that the proposing side has preference lists consisting of only $O(1)$ jobs (drawn independently at random using an arbitrary distribution). They use the algorithm to enumerate all stable partners of an agent (by breaking matches and continually running deferred acceptance) originally suggested by McVittie [21] and Gusfield [11]. In this paper, we use the simplified version by [16]. This technique is used in [13, 14, 5] to show that the number of candidates with more than one stable partner is a vanishingly small fraction, the phenomenon referred to as "core convergence."

Non-uniform preference distributions have been studied, e.g. tiered preferences [4] and correlated preferences [10]. Partial preferences based on public scores was studied in [3].

1.6 Organization

We describe the model, the deferred acceptance algorithm, as well the main building blocks of our analyses – the probability game – in Section 2. We give the the tight upper and lower bounds on the threshold in Section 3. Finally, we analyze the expected rank in Section 4.

2 Preliminaries

In this section, we formally present our model, the DA algorithm and other definitions and notations used in the rest of the paper.

Stable Matchings

Consider a stable matching instance M with $n+k$ candidates $\mathcal{C}$ and n jobs $\mathcal{J}$. Without loss of generality assume that $k \geq 0$. Each candidate $c \in \mathcal{C}$ has a (partial) preference list of length d_c^M over a subset of jobs ranking them from the most to least preferred. Similarly, each job $j \in \mathcal{J}$ has a (partial) preference list of length d_j^M over a subset of candidates ranking them from most to least preferred. Consider the undirected bipartite graph G^M on $\mathcal{C} \cup \mathcal{J}$ where the edge (c, j) is present if c appears on j's preference list and vice versa. Let $L_c^M = (j_1, \ldots, j_{d_c^M})$ to denote the list of the candidate c in M. We define L_j^M for jobs analogously.

For stable matching instances M_1 and M_2 on $\mathcal{C} \cup \mathcal{J}$, we say that $M_1 \subseteq_{\mathcal{C}} M_2$ if for each candidate $c \in \mathcal{C}$, we have that $d_c^{M_1} \leq d_c^{M_2}$ and the first $d_c^{M_1}$ elements and their order in $L_c^{M_2}$ is the same as in $L_c^{M_1}$. Moreover, for every job j, the relative ranking of the candidates in $L_j^{M_1}$ remains the same. Similarly, we say that $M_1 \subseteq_{\mathcal{J}} M_2$ if the same holds for every $j \in \mathcal{J}$.

A *matching* μ is a set of vertex-disjoint edges in G^M. A matching μ is *perfect* if each job j is an endpoint of some edge in μ. For simplicity, we let $\mu(c)$ denote a candidate c's match in μ (let $\mu(c) = \emptyset$ if c is unmatched). We define $\mu(j)$ similarly for a job j. A matching μ is *stable* if there exists no *blocking pair* with respect to μ. A pair (c, j) where $c \in \mathcal{C}$ and $j \in \mathcal{J}$ is a blocking pair with respect to the matching μ if any one of the following conditions hold: (a) c prefers j to $\mu(c)$ and j prefers c to $\mu(j)$, (b) c prefers $\emptyset$ to $\mu(c)$, and (c) j prefers $\emptyset$ to $\mu(j)$.

Deferred-Acceptance Algorithm

We describe the *candidate-proposing deferred-acceptance* (CPDA) algorithm in Algorithm 1. The job-proposing DA (JPDA) is analogous.

Let $\mathcal{U}$ be the set of unmatched candidates. Let $\mathcal{R}$ be the set of candidates who have *exhausted* their preference list, that is, have been rejected by all jobs on their preference list.

Algorithm 1 Candidate Proposing Deferred-Acceptance (CPDA) Algorithm.

```
Fix an ordering over $\mathcal{C}$; Initialize $\mathcal{U} \leftarrow \mathcal{C}$ and $\mathcal{R} \leftarrow \emptyset$ and $\mu(j) \leftarrow \emptyset$ for all $j \in \mathcal{J}$
while $\mathcal{U} \setminus \mathcal{R} \neq \emptyset$ do
    Pick a candidate $c \in \mathcal{U}$ with the lowest index
    if $c \notin \mathcal{R}$ then
        Let $j$ be the most preferred job on $c$'s list that $c$ has not yet proposed to
        if $j$ is the last job on $c$'s list then
            add $c$ to $\mathcal{R}$
        end
        $c$ proposes to $j$; Let $c' = \mu(j)$
        if $j$ prefers $c$ to $c'$ then
            $j$ rejects $c'$
            if $c' \neq \emptyset$ then
                add $c'$ to $\mathcal{U}$
            end
            $j$ accepts $c$;
            Set $\mu(j) = c$ and remove $c$ from $\mathcal{U}$
        end
    end
end
```

We use the following properties of the CPDA algorithm.

▶ **Theorem 1** ([9, 20]). *The CPDA algorithm has the following properties:*

1. *It outputs a stable matching μ^*. The matching $\mu*$ is* candidate optimal, *that is, each candidate is matched to their best-possible stable match. Moreover, μ^* is* job pessimal, *that is, each job is matched to their worst-possible stable match.*
2. *The matching μ^* is independent of the order of the proposals. Moreover, the same proposals are made during the algorithm, regardless of which candidate is chosen to propose at each step.*
3. *(**Lone Wolf Theorem**) The candidates and jobs that are unmatched in μ^* do not have any possible stable partner. That is, the set of unmatched agents is the same across all stable matchings.*

Algorithm Terminology

Let the current *time step* of the CPDA algorithm be the number of proposals made so far. Since the next proposal is always from a currently unmatched candidate with the lowest index, a candidate keeps proposing until either they are matched or exhaust their preference list. Moreover, if the tth proposal results in a rejection of a candidate c (either because c made the proposal, or because the tth proposal causes a previously-matched c to become unmatched) the next proposal is always from c. This sequence of proposals is termed as a *rejection chain* in the literature, defined more formally below.

▶ **Definition 2** (Rejection Chain). *Fix any time step t in CPDA (Algorithm 1) and let c be the next candidate in line to propose. The rejection chain $R(t)$ starting at time step t is empty if c has been rejected by all jobs on their preference list. Otherwise, let j be the job c proposes to next.*

- *If j is currently unmatched, then $R(t) = (c, j, \text{accept})$.*
- *Otherwise, if j prefers $\mu(j)$ to c, then $R(t) = (c, j, \text{reject}) \circ R(t+1)$.*
- *Otherwise, if j prefers c to $\mu(j)$, then $R(t) = ((c, j, \text{accept}), (c', j, \text{reject})) \circ R(t+1)$.*

We use the following two observations about rejection chains in our analysis.

▶ **Observation 3.** *Consider the rejection chain $R(t)$ at time step t defined in Definition 2.*

1. *$R(t)$ terminates when either previously unmatched job receives a proposal and thus will remain matched till the end, or a candidate is rejected by all jobs on their preference list and thus will remain unmatched till the end.*
2. *In a rejection chain, if d contiguous proposals result in a rejection, then some candidate must have been rejected d times and must remain unmatched.*

Note that Part (2) of the above observation was also used by [15].

Deferred Acceptance on Random Instances

We recall the random stable matching instance $M_{n,d,k}$ defined in Section 1. In this instance, there are $m = n + k$ candidates and n jobs. The underlying graph is a Erdös-Rényi graph $G_{n+k,n,d/n}$. Thus, an edge (c, j) between a candidate c and job j occurs with independent probability d/n. Moreover, each candidate and job have a random ranking over their neighbors, chosen uniformly and independently. The stable matching instance $M_{n,d,k}$ refers to the graph $G_{n+k,n,d/n}$ and the preferences over the edges.

Let us define two other related models of random stable matching instances. Let us define $M^{\mathcal{C}}_{n,d,k}$ to denote a stable matching instance with n jobs, $n + k$ candidates, and each candidates chooses a preference list of d jobs uniformly and independently. This is the model in [15].

Similarly, we also define $M^{\mathcal{J}}_{n,d,k}$ where each job chooses a preference list of d candidates uniformly and independently.

We analyze the CPDA algorithm on $M^{\mathcal{C}}_{n,d,k}$ and the JPDA algorithm on $M^{\mathcal{J}}_{n,d,k}$. To obtain the final result on the symmetric stable matching instance $M_{n,d,k}$, we relate all three random instances using the following "sandwiching" lemma.

▶ **Lemma 4.** *(Sandwiching Lemma) For every $\delta > 0$, there is a joint distribution $(M^{\mathcal{C}}_{n,d(1-\delta),k}, M_{n,d,k}, M^{\mathcal{C}}_{n,d(1+\delta),k})$ such that with probability at least $1 - O\left(n \exp\{-\delta^2 d/3\}\right)$, we have*

$$M^{\mathcal{C}}_{n,d(1-\delta),k} \subseteq_{\mathcal{C}} M_{n,d,k} \subseteq_{\mathcal{C}} M^{\mathcal{C}}_{n,d(1+\delta),k}.$$

The same claim also holds for $(M^{\mathcal{J}}_{n,d(1-\delta),k}, M_{n,d(1+(k/n)),k}, M^{\mathcal{J}}_{n,d(1+\delta),k})$.

Since we (eventually) analyze the DA algorithm on a random instance $M^{\mathcal{C}}_{n,d,k}$, we reveal the next job j in the list of candidate c when they are called up to propose, uniformly among all jobs that have not already rejected c. We also reveal c's relative ranking among all the proposals received by j (including from c) when c proposes to j.

2.1 Probability Game

Define a multi-round probabilistic game as follows. In each round, there are three events that can occur: G (Good), B (Bad), and N (Neutral). Given an input parameter d, for each round, conditioned on the events in previous rounds, define $\Pr(G) = p_G$, $\Pr(B) \geq p_B$ and $\Pr(N) = p_N$. The game is *won* if G occurs. The game is *lost* if B occurs d times *in a row*.

Let $\Pr_{\text{win}}(p_G, p_B)$ denote the probability of winning this game. Then, we prove the following.

▶ **Lemma 5.**

$$\Pr_{\text{win}}(p_G, p_B) \leq d \cdot \frac{p_G}{p_B^d}$$

Proof. Suppose the game is played for i rounds, and let $S = (s_1, s_2, \ldots, s_i)$ denote the random sequence of events that occur. Consider a new game that is the same as the original except now $\Pr(B) = p_B$, and let $S' = (s'_1, s'_2, \ldots, s'_i)$ denote the random sequence of events that occur when the new game is played for i rounds. Let $\Pr_{\text{win}'}$ be the winning probability in the new game. Through a coupling argument, we first show that $\Pr_{\text{win}} \leq \Pr_{\text{win}'}$.

Given the sequence S', we create a sequence S as follows. For $i \geq 0$, as long as the game does not end in a loss at i rounds, do:

- if $s'_i = G$, then $s_i = G$,
- if $s'_i = B$, then $s_i = B$,
- if $s'_i = N$, then $s_i = N$ with probability $\frac{1-p_G-p_B^i}{1-p_G-p_B}$ and $s_i = B$ otherwise. Here $p_B^i = \Pr(s_i = B \mid s_{i-1}, \ldots, s_1) \geq p_B$.

Note that whenever S' ends in a loss (event B occurs d times in a row), then S also ends in a loss. Thus, $\Pr_{\text{win}} \leq \Pr_{\text{win}'}$.

Finally, we finish the proof of the lemma by using the following recurrence.

$$\begin{aligned}
\Pr_{\text{win}'} &= p_G + p_N \cdot \Pr_{\text{win}'} + p_B \cdot p_N \cdot \Pr_{\text{win}'} + p_B \cdot P_G + p_B^2 \cdot p_N \cdot \Pr_{\text{win}'} + p_B^2 \cdot p_G \\
&\quad + \ldots + p_B^{d-1} \cdot p_N \cdot \Pr_{\text{win}'} + p_B^{d-1} \cdot p_G \\
&= p_G + p_N \cdot \Pr_{\text{win}'} (1 + p_B + p_B^2 + \ldots + p_B^{d-1}) + p_B \cdot p_G (1 + p_B + p_B^2 + \ldots p_B^{d-2}) \\
&= p_G + p_N \cdot \Pr_{\text{win}'} \left(\frac{1 - p_B^d}{1 - p_B} \right) + p_B \cdot p_G \left(\frac{1 - p_B^{d-1}}{1 - p_B} \right)
\end{aligned}$$

So we have

$$\begin{aligned}
\Pr_{\text{win}'} &= \frac{p_G(1-p_B) + p_B \cdot p_G (1 - p_B^{d-1})}{1 - p_B - p_N + p_N \cdot p_B^d} = \frac{p_G - p_B^d}{p_G + p_N \cdot p_B^d} \\
&= \frac{p_G}{p_B^d} \cdot \frac{(1 - p_B^d)}{(p_G / p_B^d + p_N)} \leq d \cdot \frac{p_G}{p_B^d}
\end{aligned}$$ ◀

We say an event $\mathcal{E}$ occurs *with high probability* if $\mathbb{P}(\mathcal{E}) = 1 - o(1)$.

3 Sharp Threshold for Perfect Stable Matchings

In this section, we formally state and prove our main theorem.

▶ **Theorem 6.** *Consider a stable matching instance $M_{n,d,k}$ where $0 \le k = o(n)$ and $1 \le d \le n$. Then, for each $\epsilon > 0$,*

1. *if $d > (1+\epsilon)\ln n \cdot \ln\left(\frac{n+k}{k+1}\right)$, then w.h.p, all stable matchings are perfect, and*
2. *if $d < (1-\epsilon)\ln n \cdot \ln\left(\frac{n+k}{k+1}\right)$, then w.h.p, no stable matching is perfect.*

That is, we prove tight upper and lower bounds on the degree d that determines whether or not a stable matching exists in a random matching market with imbalance k and average degree d. As alluded to earlier, we prove the lower bound on $M^{\mathcal{C}}_{n,d(1+o(1)),k}$, and upper bound on $M^{\mathcal{J}}_{n,d(1-o(1)),k}$.

▶ **Lemma 7.** *For any $\epsilon > 0$, if $d < (1-\epsilon)\ln n \cdot \ln\left(\frac{n+k}{k+n/(n+k)}\right)$, then*

$$\mathbb{P}\left(M^{\mathcal{C}}_{n,d,k} \text{ has no perfect stable matching}\right) \ge 1 - (k/n)^{\Omega(\epsilon)}.$$

▶ **Lemma 8.** *For any $\epsilon > 0$, if $d > (1+\epsilon)\ln n \cdot \ln\left(\frac{n+k}{k+n/(n+k)}\right)$, then*

$$\mathbb{P}(M^{\mathcal{J}}_{n,d,k} \text{ has a perfect stable matching}) \ge 1 - n^{-\Omega(\epsilon)}.$$

We prove Lemmas 7 and 8 in Section 3.2 and Section 3.3 respectively. Using them, we finish the proof of Theorem 6 below.

Proof of Theorem 6. Let us denote $d_0 := \ln n \cdot \ln\left(\frac{n+k}{k+1}\right)$ and $k = o(n)$. Observe that $d_0 \sim \ln n \cdot \ln\left(\frac{n+k}{k+n/(n+k)}\right)$ for $k \le n$.

Denote $M = M_{n,d_0(1-\epsilon),k}$, and $M^{\mathcal{C}} = M^{\mathcal{C}}_{n,d_0(1-\epsilon/2),k}$. For the proof of part 1, it suffices to prove the same statement for $M^{\mathcal{C}}$. Indeed, since by Lemma 4, there is a joint distribution $(M, M^{\mathcal{C}})$ such that $M \subseteq_{\mathcal{C}} M^{\mathcal{C}}$ with probability at least $1 - n\exp\{-\Omega(\epsilon^2 d_0)\}$). By part 2 of Theorem 1, we have that if the CPDA algorithm on $M^{\mathcal{C}}$ does not give a perfect stable matching, then it does not give one even in M.

The proof of part 2 proceeds in a similar fashion, using $M^{\mathcal{J}}_{n,(1+(k/n))d_0(1+\epsilon/2),k}$ ◀

3.1 High-level Overview

We first describe the high-level idea of the upper and lower bound proofs for the balanced case ($k = 0$) to give some intuition.

Lower bound

To show that a perfect stable matching does not exist when the average degree $d < (1-\epsilon)\ln^2 n$ in $M_{n,d,0}$, we analyze the proposals when the CPDA algorithm is run on the random instance $M^{\mathcal{C}}_{n,d,0}$. Thus, each candidate has a preference list of length d. Notice that in the CPDA, as soon as all n jobs receive even a single proposal, the final matching must be perfect. This is because jobs only upgrade their potential matches as the algorithm proceeds. On the other hand, a candidate c remains unmatched if c has been rejected by all d jobs on their preference list. Thus, to show that a perfect stable matching does not exist, we show that some candidate exhausts all d of their proposals *before* all n jobs receive a proposal. We model this using the probability game from Section 2.1. Finally, we set the probabilities in the game using a careful coupling between CPDA and a balls-in-bins process.

This proof naturally extends to the unbalanced case: we simply analyze the probability that $k+1$ candidates exhaust all d of their proposals before all n jobs receive a proposal.

Upper bound

To show that a perfect stable matching exists when the average degree $d > (1+\epsilon)\ln^2 n$ in $M_{n,d,0}$, we analyze the proposals in the JPDA algorithm on the random instance $M^{\mathcal{J}}_{n,d,0}$. Thus, each job has a preference list of length d. To prove that each job gets matched, we look at all the proposals made by the last job j in JPDA and compute the probability that j gets rejected from all d candidates on its list and show that it is small. For a single proposal from j to a candidate c to be successful, not only must c accept it but the resulting rejection chain must not knock j off. We use the probability game to compute the probability of the latter.

3.2 Lower Bound on Threshold

Next, we prove the lower bound for any $k = o(n)$ formally.

Proof of Lemma 7. Let $k_0 := n^{1-O(\epsilon)}$ be small enough that $\frac{k_0}{n} \ll (d/2)^{-12/\epsilon}$. We split the analysis into whether k is larger or smaller than k_0.

Case 1: $k < k_0$. Let $\mathcal{P}$ denote the event that $M^{\mathcal{C}}_{n,d,k}$ has a perfect stable matching. Recall that in $M^{\mathcal{C}}_{n,d,k}$, each candidate has a preference list of length d. We consider the CPDA procedure in Algorithm 1 on $M^{\mathcal{C}}_{n,d,k}$. We define an *extended process* in the algorithm where there are $n+k$ real candidates and ∞ "fake" candidates and n jobs. Recall that we assume that there is some predetermined order on the candidates and the unmatched candidate with the least index is next in line to propose.

Fix $\kappa = n\ln n - Cn$, for $C = O(\epsilon \ln(n/(n+k)))$. Suppose κ proposals have been made. We consider the rejection chain $R(\kappa)$ as defined in Definition 2. In particular, consider the proposal made by candidate $\tilde{c}$ at step $\kappa+1$. We have the following cases: (a) the proposal goes to an unmatched job and is accepted, (b) the proposal goes a matched job and is rejected, and (c) the proposal goes to a matched job, is accepted, which causes its previous match c' now to be free. By Observation 3, $R(\kappa)$ ends when either a proposal goes to an unmatched job, or when some candidate exhausts all their d proposals (and thus remains unmatched).

Let us define the following events and then bound their probabilities.

- Let $\mathcal{M}$ be the event that CPDA finds a (potential) matching of size n before $k+1$ candidates have exhausted all d of their proposals.[2]
- Let $\mathcal{L}$ be the event that every job gets a proposal from a candidate in at most κ rounds.
- Let $\mathcal{B}$ be the event that after κ rounds, the probability that the next proposal from a candidate is accepted is greater than $\frac{n}{n+\kappa}(1+\frac{\epsilon}{3})$.
- Let $\mathcal{E}$ be the event that the number of unmatched jobs at κ rounds is at least e^{2C}.

Then,

$$\mathbb{P}(\mathcal{P}) \le \mathbb{P}(\mathcal{M}) \le \mathbb{P}(\mathcal{M}|\overline{\mathcal{BEL}}) + \mathbb{P}(\mathcal{B}) + \mathbb{P}(\mathcal{E}) + \mathbb{P}(\mathcal{L}). \tag{1}$$

As there are $n+k$ real candidates and n jobs, there is no perfect stable matching if more than k candidates end up with all d of their proposals rejected before all n jobs receive even a single proposal. We have the following claim whose proof is omitted.

[2] Note that a potential matching of size n is found as soon as n jobs have received a proposal from some candidate (not necessarily their final match).

▷ Claim 9. The following inequalities hold:

1. $\mathbb{P}(\mathcal{B}) \leq O(1/\sqrt{n})$.
2. $\mathbb{P}(\mathcal{E}) \leq e^{-C}$.
3. $\mathbb{P}(\mathcal{L}) \leq 2e^{-C/2} + O(1/\sqrt{n})$

We finish the proof as follows. Let $p_B := 1 - \frac{n}{n+\kappa}(1+\frac{\epsilon}{2})$. Conditioned on $\overline{\mathcal{B}}$, the next proposal is rejected with probability at least p_B. Conditioned on $\overline{\mathcal{E}}$, let p_G be the probability of the next proposal goes to an unmatched job. By Observation 3 (1) the number of unmatched jobs stays the same during all proposals that occur within the rejection chain $R(\kappa)$. By Observation 3 (2), any time there are d rejects in a row in $R(\kappa)$, some candidate must have exhausted all d of their proposals.

Now, we use the probability game from Section 2.1 with $p_G = \frac{e^{2C}}{n}$. Using Lemma 5 and the fact that $\mathbb{P}_{\text{win}}(p_G, p_B)$ is the probability that every subsequent rejection chain ends in a new job getting matched, we get the following:

$$\begin{aligned}\mathbb{P}_{\text{win}}(p_G, p_B) &\leq d \cdot p_G p_B^{-d}\\ &\leq d \cdot e^{2C} n^{-1} \left(1 - \frac{n}{n+\kappa}(1+\epsilon/3)\right)^{-(1-\epsilon)\ln n \ln\left(\frac{n+k}{k+n/(n+k)}\right)}\\ &\leq d \cdot e^{2C} n^{-1} \left(\frac{k+n/(n+k)}{n+k}\right)^{-1+\epsilon/2}.\end{aligned}$$

In the last inequality, we used the fact that $1 - x \geq e^{-x-x^2}$, and that $\kappa = \Omega(n \ln n)$. So, with probability at least $1 - \left(d \cdot e^{2C}\left(\frac{k}{n} + \frac{1}{n+k}\right)^{\epsilon/4} (1+\frac{k}{n})^{1-\epsilon/2}\right)$ the next $n \cdot \left(\frac{k}{n} + \frac{1}{n+k}\right)^{1-\epsilon/4} > k$ rejection chains end in an additional unmatched candidate. Therefore,

$$\mathbb{P}(\mathcal{M} | \overline{\mathcal{B}\mathcal{E}\mathcal{L}}) \leq d \cdot e^{2C}\left(\frac{k}{n} + \frac{1}{n+k}\right)^{\epsilon/4}\left(1 + \frac{k}{n}\right)^{1-\epsilon/2}. \tag{2}$$

Finally, substituting $C = O(\epsilon \ln(n/(n+k)))$ and $k/n \leq (d/2)^{-12/\epsilon}$, and plugging in the bounds from Claim 9 and Inequality (2) into the expression in Inequality (1), we get

$$\mathbb{P}(\mathcal{P}) \leq (k/n)^{\Omega(\epsilon)} \leq (\log n)^{-\Omega(1)}.$$

Case 2: $k \geq k_0$. We repeat a similar argument as before using the extended JPDA, setting $\kappa = (n+k)\ln\left(\frac{n+k}{k}\right)(1-\delta)$ for some small $\delta > 0$. As before, we have:

- Let $\mathcal{M}$ be the event that JPDA finds a (potential) matching of size n before any job has run out of proposals.[3]
- Let $\mathcal{B}$ be the event that after κ rounds, the probability that the next proposal to a candidate is accepted is greater than $\frac{n}{n+\kappa}(1+\frac{\epsilon}{3})$.
- Let $\mathcal{E}$ be the event that there are fewer than $\frac{2kn}{n+k}$ unmatched candidates at κ rounds.

We have

$$\mathbb{P}(\mathcal{P}) \leq \mathcal{P}(\mathcal{M}) \leq \mathbb{P}(\mathcal{M}|\overline{\mathcal{B}\mathcal{E}}) + \mathbb{P}(\mathcal{E}) + \mathbb{P}(\mathcal{B}).$$

Similar to Claim 9, we have the following claim:

[3] Note that a potential matching of size n is found as soon as n jobs have received a proposal from some candidate (not necessarily their final match).

▷ Claim 10. The following inequalities hold:
1. $\mathbb{P}(\mathcal{B}) \le O(1/\sqrt{n})$.
2. $\mathbb{P}(\mathcal{E}) \le O(1/\sqrt{n})$.

We finish the argument as follows: Conditioned on $\overline{\mathcal{E}}$, there are at least $\frac{kn}{n+k}$ unmatched jobs at this point. Conditioned on $\mathcal{B}$, the probability that the next job runs out of proposals without matching to a candidate is at least $\left(1 - \frac{n}{n+\kappa}\left(1+\frac{\epsilon}{3}\right)\right)^d \ge n^{-1+\frac{\epsilon}{4}}$. So, the probability that at least one job runs out of proposals in the next $\frac{kn}{n+k} = n^{1-O(\epsilon)}$ rejection chains is at least $1 - n^{-\Omega(\epsilon)}$. ◀

3.3 Upper Bound on Threshold

Below, we prove a tight upper bound on the threshold.

Proof of Lemma 8. Suppose $d > (1+\epsilon)\ln n \cdot \ln\left(\frac{n+k}{k+n/(n+k)}\right)$. Consider the JPDA algorithm (analogous to Algorithm 1). Fix a job j, and let us place it in the last place in our predetermined order for the JPDA algorithm – that is, j is the last job that has not yet proposed to any candidate on their list. Let κ be the number of proposals at the moment j starts proposing. Let (the random candidate) c be j's next proposal and N_c be the random variable denoting the number of proposals c has received so far. We have that $\mathbf{E}[N_c|\kappa] = \kappa/(n+k)$. Observe that j's proposal to c is *successful* if the following two events occur:
1. c accepts j's proposal, and
2. the resulting rejection chain $R(\kappa)$ does not end with j getting unmatched.

The first event occurs with probability $1/(N_c+1)$.

To bound the probability of the second event, we analyze the proposals in $R(\tau)$ using the probability game from Section 2.1. In the particular, consider the probability game with parameter $d = 1$. Let G be the event that j gets unmatched because another job j' proposes to c in $R(\tau)$. Then, $p_G \le \frac{1}{(n+k)(N_c+2)}$ because the probability that c is next on j' preference list is at most $1/m$ and c must prefer j' to the N_c+1 proposals it already has received. Let B be the event that the rejection chain $R(\kappa)$ ends without j getting kicked off from c in $R(\kappa)$. This occurs if $R(\kappa)$ ends in a new candidate getting matched. Thus $p_B \ge \frac{k}{n+k} + \frac{n}{(n+k)^2}$.

Then, $\mathbb{P}_{\text{win}}$ corresponds to the probability that j does not end up being matched to c at the end of $R(\kappa)$, and by Lemma 5 we have,

$$\mathbb{P}_{\text{win}} \le p_G/p_B \le \frac{n+k}{(k+n/(n+k))\cdot(n+k)\cdot(N_c+2)} \le \frac{1}{N_c+2} \tag{3}$$

Thus, the probability that j's proposal to c is successful is at least

$$\begin{aligned}&\mathbf{E}\left[\frac{1}{N_c+1}\left(1-\frac{1}{N_c+2}\right)\right]\\ &= \mathbf{E}\left[\frac{1}{N_c+2}\right] \ge \frac{1}{\mathbf{E}\left[N_c+2\right]} = \frac{n+k}{\mathbf{E}[\kappa]+2(n+k)} \ge \frac{n+k}{\mathbf{E}[\tau]+4n}.\end{aligned}$$

Here τ denotes the final number of proposals (at the end of JPDA) and κ is the number of proposals at the time j started proposing.

Finally, the probability that j remains unmatched at the end of JPDA is the probability that all d of j's proposals fail. This occurs with probability

$$\begin{aligned}&\leq \left[1-\frac{n+k}{\mathbf{E}[\tau]+4n}\right]^d \leq \exp\left\{-\frac{(n+k)\cdot d}{\mathbf{E}[\tau]+4n}\right\}\\ &\leq \exp\left\{-\frac{d}{4+\ln\left(\frac{n+k}{k+n/(n+k)}\right)}\right\} \leq \frac{1}{n^{1+\epsilon/2}}.\end{aligned} \tag{4}$$

Here, the inequality in step 4 follows from the following claim whose proof is omitted.

▷ Claim 11. $\mathbf{E}[\tau] \leq (1+\epsilon/2)\cdot m \cdot \ln\left(\frac{n+k}{k+n/(n+k)}\right) + o(1)$

Theorem 1 part 3 gives us that this holds for any job j. Thus the expected number of unmatched jobs is at most $n^{-\epsilon/2}$, which gives us the desired bound. ◀

4 Effect of Competition

In this section, we prove a lower bound on the expected ranks of the candidates and the number of proposals in CPDA in an unbalanced random matching market with imbalance k. In particular, we prove the following.

▶ **Theorem 12.** *Consider a stable matching instance $M^{\mathcal{C}}_{n,d,k}$ where $0 \leq k = o(n)$ and $d = \omega(\log n)$. Let $r_{\mathcal{C}}$ denote the expected rank of the best-possible stable partner each candidate can obtain. Then we have the following inequalities*

1. $\mathbf{E}[r_{\mathcal{C}}] \geq (1-o(1))\frac{d}{\ln\frac{n+k}{k+n/(n+k)}}$,
2. *if* $(n+k)e^{-\sqrt{d}} > 1$, *then* $\mathbf{E}[r_{\mathcal{C}}] \geq (1-o(1))\sqrt{d}$.

Proof of Theorem 12. Consider the JPDA algorithm on $M^{\mathcal{J}}_{n,d,k}$. We use the following lemma from past works [7, 13, 17].

▶ **Lemma 13** ([7]). *Fix a job j and for any preference list P_j of job j and index $i \in [n]$, let P^i_j denote the preference list P_j truncated at index i. Then, job j has a stable partner of rank better than i if and only if j is matched in CPDA even if j truncates their list after rank i.*

We fix a candidate c^*. Let X be the random variable that represents c^*'s highest-rank stable partner (among all stable matchings). Let Y be the highest rank offer c^* receives if in the JPDA algorithm c^* keeps rejecting all proposals (and we keep running JPDA). Lemma 13 gives us that $\mathbf{E}[X] = \mathbf{E}[Y]$. Let Z be the random variable representing the number of proposals received by c^* during this algorithm where c^* rejects all proposals.

Let us compare the JPDA process to a random balls-in-bins process. Assume that there are $m = n+k$ bins (one for each candidate). Suppose balls are being thrown uniformly and independently in the bins. Fix a bin c^* and let $\tilde{Z}$ be the number of balls in c^* at the time that exactly $n+1$ bins are occupied. Let κ be the number of balls thrown at this point. We show that this process *stochastically dominates* JPDA as follows.

▶ **Lemma 14.** *There is a coupling between the JPDA algorithm and the balls-in-bins process so that $Z \leq \tilde{Z}$, and $\tau \leq \kappa$.*

Proof. We abuse notation and use $[m]$ to denote both the set of bins and the set of candidates. Consider the following coupling between the processes of JPDA and balls-in-bins.

Suppose at a point in the JPDA algorithm, a job $j \in [n]$ is required to make a (random) proposal, and so far, $\mathbf{C}_j$ is the set of candidates that j has already proposed to. We keep throwing balls-into-bins uniformly at random until we hit a bin $\mathbf{x} \notin \mathbf{C}_j$. We assign $\mathbf{x}$ as the next proposal of j. Thus the distribution of the next proposal of j is uniform on $[m] \setminus \mathbf{C}_j$.

Let τ and κ denote the number of proposals made and number of balls thrown in the above algorithm respectively. Fix any $i \in [m]$, and let Z and $\hat{Z}$ denote the number of proposals received by the candidate i and the number of balls in the bin i respectively. We finish the proof by noting that the set $\mathcal{B}'$ of bins occupied is a superset of the set $\mathcal{C}'$ of candidates who received proposals (the algorithm could end earlier than the balls-in-bins process, where we have unlimited balls). Also, for the above process, $\tau \leq \kappa$ and $Z \leq \hat{Z}$. ◀

As an immediate consequence, using symmetry, we have

$$\mathbf{E}[Z] \leq \mathbf{E}[\tilde{Z}] = \mathbf{E}[\mathbf{E}[\tilde{Z}|\kappa]] = \mathbf{E}[\kappa/m].$$

Since c^* has d jobs on their list, the expected highest rank proposal they receive is

$$\mathbf{E}[r_{\mathcal{C}}] = \mathbf{E}[Y] = \mathbf{E}\left[\mathbf{E}[Y|Z]\right] = \mathbf{E}[d/(Z+1)] \geq d/\mathbf{E}[Z+1] \geq d \cdot m/(\mathbf{E}[\kappa]+m).$$

So the expected rank of the highest ranked stable match of every candidate is at least r_m. Therefore, the expected number of proposals in the CPDA algorithm is at least $d \cdot m^2/\mathbf{E}[\kappa]$.

The proof is completed by the following proposition, whose proof is identical the proof of to Claim 11 and so we omit it.

▶ **Proposition 15.** $\mathbf{E}[\kappa] \leq (1+o(1))m \ln\left(\frac{m}{k+n/m}\right)$. ◀

References

1 Atila Abdulkadiroğlu, Parag A Pathak, and Alvin E Roth. The new york city high school match. *American Economic Review*, 95(2):364–367, 2005.

2 Atila Abdulkadiroğlu, Parag A Pathak, Alvin E Roth, and Tayfun Sönmez. The boston public school match. *American Economic Review*, 95(2):368–371, 2005.

3 Ishan Agarwal and Richard Cole. Stable matching: Choosing which proposals to make, 2023. `arXiv:2204.04162`.

4 Itai Ashlagi, Mark Braverman, Amin Saberi, Clayton Thomas, and Geng Zhao. Tiered Random Matching Markets: Rank Is Proportional to Popularity. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, volume 185 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 46:1–46:16, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.ITCS.2021.46`.

5 Itai Ashlagi, Yash Kanoria, and Jacob D Leshno. Unbalanced random matching markets: The stark effect of competition. *Journal of Political Economy*, 125(1):69–98, 2017.

6 Itai Ashlagi and Afshin Nikzad. What matters in school choice tie-breakings? how competition guides design. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 767–768, 2016. `doi:10.1145/2940716.2940762`.

7 Linda Cai and Clayton Thomas. The short-side advantage in random matching markets. In *Symposium on Simplicity in Algorithms (SOSA)*, pages 257–267. SIAM, 2022. `doi:10.1137/1.9781611977066.19`.

8 Jose Correa, Rafael Epstein, Juan Escobar, Ignacio Rios, Bastian Bahamondes, Carlos Bonet, Natalie Epstein, Nicolas Aramayo, Martin Castillo, Andres Cristi, et al. School choice in chile. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 325–343, 2019.

9 David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *The American mathematical monthly*, 69(1):9–15, 1962.

10 Hugo Gimbert, Claire Mathieu, and Simon Mauras. Two-sided matching markets with strongly correlated preferences. In *Fundamentals of Computation Theory: 23rd International Symposium, FCT 2021, Athens, Greece, September 12–15, 2021, Proceedings 23*, pages 3–17. Springer, 2021. doi:10.1007/978-3-030-86593-1_1.

11 Dan Gusfield. Three fast algorithms for four problems in stable marriage. *SIAM Journal on Computing*, 16(1):111–128, 1987. doi:10.1137/0216010.

12 Dan Gusfield and Robert W Irving. *The stable marriage problem: structure and algorithms*. MIT press, 1989.

13 N Immoralica. Marriage, honesty, and stability. In *Proceedings of the 16th annual ACM-SIAM symposium on Discrete algorithms, 2005*, pages 53–62, 2005.

14 Nicole Immorlica and Mohammad Mahdian. Incentives in large random two-sided markets. *ACM Transactions on Economics and Computation (TEAC)*, 3(3):1–25, 2015. doi:10.1145/2656202.

15 Yash Kanoria, Seungki Min, and Pengyu Qian. In which matching markets does the short side enjoy an advantage? In *Proceedings of the Thirty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1374–1386, 2021. doi:10.1137/1.9781611976465.83.

16 Donald E Knuth, Rajeev Motwani, and Boris Pittel. Stable husbands. *Random Structures & Algorithms*, 1(1):1–14, 1990. doi:10.1002/RSA.3240010102.

17 Donald Ervin Knuth. Mariages stables et leurs relations avec d'autres problèmes combinatoires: introduction à l'analyse mathématique des algorithmes. *Les Presses de l'Universite de Montreal*, 1976.

18 David Manlove. *Algorithmics of matching under preferences*, volume 2. World Scientific, 2013. doi:10.1142/8591.

19 Simon Mauras. *Analysis of random models for stable matchings*. PhD thesis, Université Paris Cité, 2021.

20 David G McVitie and Leslie B Wilson. Stable marriage assignment for unequal sets. *BIT Numerical Mathematics*, 10(3):295–309, 1970.

21 David G McVitie and Leslie B Wilson. The stable marriage problem. *Communications of the ACM*, 14(7):486–490, 1971. doi:10.1145/362619.362631.

22 NRMP. How many programs can I rank? — nrmp.org. https://www.nrmp.org/help/item/how-many-programs-can-i-rank/#, 2024. [Accessed 09-02-2025].

23 Boris Pittel. The average number of stable matchings. *SIAM Journal on Discrete Mathematics*, 2(4):530–549, 1989. doi:10.1137/0402048.

24 Boris Pittel. On likely solutions of a stable marriage problem. *The Annals of Applied Probability*, 2(2):358–401, 1992.

25 Boris Pittel. The" stable roommates" problem with random preferences. *The Annals of Probability*, 21(3):1441–1477, 1993.

26 Alvin E Roth. The economist as engineer: Game theory, experimentation, and computation as tools for design economics. *Econometrica*, 70(4):1341–1378, 2002.

27 Alvin E Roth and Elliott Peranson. The redesign of the matching market for american physicians: Some engineering aspects of economic design. *American economic review*, 89(4):748–780, 1999.

28 LB Wilson. An analysis of the stable marriage assignment algorithm. *BIT Numerical Mathematics*, 12(4):569–575, 1972.

29 Leslie B. Wilson. An analysis of the stable marriage assignment algorithm. *BIT Numerical Mathematics*, 12:569–575, 1972.