The Stable Marriage Algorithm: Extending to Real-World Preferences

Irhan Iftikar

Abstract

This paper extends the classical Stable Marriage Problem (SMP) by incorporating the concept of preference ties, where participants can equally rank multiple options. Building upon the foundational works of Gale and Shapley, the paper presents a modified algorithm that allows proposers to simultaneously approach multiple equally preferred choices, with a subsequent selection process when multiple proposals are accepted. Using concrete examples with a 5×5 preference matrix, the paper demonstrates how the proposed extension incorporates preference ambiguities while preserving termination and stability. The analysis reveals important implications for the algorithm's optimality properties and complexity. This extended algorithm has significant applications in college admissions, medical residency matching, and other allocation problems, offering a more practical and adaptable framework for modern matching systems where strict preference orderings are unrealistic.

1. Introduction

The Stable Marriage Algorithm (SMA), first introduced by David Gale and Lloyd Shapley in 1962, revolutionized the way we think about matching problems in computer science. In 2012, the Nobel Prize in Economics was awarded to Lloyd Shapley and Alvin Roth for the "theory of stable allocations and the practice of market design." While originally designed to pair equal numbers of men and women in stable marriages where no two people would prefer each other to their assigned partners, the algorithm has found widespread applications in various real-world scenarios, from matching medical residents to hospitals to assigning students to schools. However, the classical algorithm assumes that all participants have strict preferences with no ties allowed, which often doesn't reflect real-world situations.

This paper explores an extended version of the Stable Marriage Algorithm that accommodates ties in preferences, making it more applicable to real-world scenarios. In many practical applications, a participant may view multiple choices as equally desirable—for instance, a student might equally prefer two different universities, or a hiring manager might consider several candidates to be equally qualified. We analyze how this modification affects the algorithm's properties, including its termination conditions, stability guarantees, and optimality characteristics. Special attention is given to how allowing ties impacts the traditional boy-optimal property of the algorithm and introduces new challenges in achieving stable matchings.

2. The Assignment Criteria

Matching problems arise in various applications, from college admissions to hospital residency placements and resource allocation systems. A common formulation involves assigning n applicants to m colleges, where each college has a fixed number of available spots and ranks applicants based on preferences. Applicants, in turn, rank colleges they would consider attending in their order of preference.

While a straightforward preference-based assignment might seem ideal, conflicts arise when mutual preferences do not align, leading to unstable assignments—cases where an applicant and a college would both prefer to be matched with each other over their assigned pair. To ensure fairness and efficiency, the concept of stability is introduced, ensuring no applicant-college pair has incentive to deviate from the final assignment. An assignment is optimal if no other stable matching provides a better outcome for any applicant.

An elegant solution would be an independent, trusted, and centralized clearinghouse versus the current decentralized college admissions system. Such a clearinghouse approach would immediately yield multiple benefits, including the elimination of strategic behavior (as students wouldn't need to guess about "reach" versus "safety" schools), the reduction of application inflation, a more predictable yield for colleges, and lower costs for both sides (with fewer applications to process and submit). In fact, this approach has already demonstrated success in other contexts, such as medical residency matching. However, its implementation in college admissions faces significant obstacles such as early decision programs, holistic evaluation criteria, and varying institutional priorities.

To establish some groundwork, an assignment of applicants to colleges is considered unstable if there exist two applicants who are assigned to colleges A and B, respectively, but one applicant prefers A over B, and college A also prefers this applicant over their current assignment. Additionally, a stable assignment is called optimal if every applicant is at least as well off under it as under any other stable assignment.

3. Related Work

The classical SMP provides a foundational algorithm for achieving stable matchings in two-sided markets where participants have strict preferences. Since its introduction, SMP has been widely applied to real-world allocation problems such as medical residency matching (Roth & Peranson, 1999) and college admissions. However, the assumption that all participants provide strict preference rankings is often unrealistic, as real-world decisions frequently involve indifference between multiple options. Irving (1994) extended the problem by allowing preference ties and introduced algorithms that find stable matchings under weak and strong stability criteria. Later, Manlove (2013) analyzed the computational complexity of handling indifferences in SMP, showing that while finding weakly stable matchings remains polynomial-time solvable, computing certain optimal solutions becomes NP-hard in some variants.

More recent work has explored tie-breaking mechanisms and their impact on fairness and efficiency in matching markets. Abdulkadiroğlu et al. (2009) examined how allowing preference ties affects school choice mechanisms, emphasizing the need for stable, strategy-proof solutions in real-world applications. Studies in market design (Roth, 2008) have further investigated how preference ties influence outcomes in centralized matching markets, such as residency placement and school admissions. Our work builds upon these foundations by modifying the Gale-Shapley algorithm to systematically handle ties in proposer preferences while preserving termination guarantees and stability. Unlike previous approaches that require external tie-breaking rules, our modification allows proposers to submit multiple simultaneous proposals, providing a more practical approach for modern matching systems where preference ties are unavoidable.

4. The Stable Marriage Problem

In exploring the existence of stable assignments, Gale and Shapley first examined a special case that would become foundational to their solution: a scenario where the number of applicants equals the number of colleges, and each college has a quota of one student. While this simplified case seemed unnatural for college admissions, they realized it mapped perfectly to another problem—matching marriage partners in a community with equal numbers of men and women. In this marriage analogy, each person ranks all potential partners of the opposite sex according to their preferences, and the goal is to find a "stable" set of marriages where no two people would prefer each other to their assigned partners. This reframing of the problem as a marriage matching scenario provided a clearer way to analyze and solve the underlying mathematical challenge of finding stable assignments. Hence, we are going to talk about this variant of the matching problem that has an elegant solution and that is frequently used in practice. Now, let's formalize the statement of the stable marriage problem.

In this setting, there are n boys and n girls. We assume the number of boys and girls is the same. Each boy has his own ranked preference of girls. Each girl has her own ranked preference of boys. The lists are complete and have no ties. Each boy ranks every girl and vice versa.

The end goal is to pair each boy with a unique girl so that there are no rogue couples. That is, to find a perfect matching so that every boy and girl are paired up one to one, with no potential for both to defect. Let's see if we can figure out a method for finding a stable match by looking at the preference matrix below, where in each cell (B_i, G_j) we have the ordered pair (x, y) where x is Boy i's ranking of Girl j, and y is Girl j's ranking of Boy i.

	G1	G2	G3	G4	G5
B1	(4, 4)	(2, 3)	(1, 4)	(5, 1)	(3, 4)
B2	(1, 3)	(2, 2)	(4, 5)	(5, 2)	(3, 3)
B3	(4, 1)	(3, 5)	(2, 2)	(1, 3)	(5, 2)
B4	(1, 5)	(4, 4)	(2, 1)	(3, 4)	(5, 1)
B5	(1, 2)	(2, 1)	(5, 3)	(3, 5)	(4, 5)

Table 1: Preference Matrix with no ties allowed

Let's try to use a greedy algorithm to find the matching. In this case, the greedy algorithm will have each boy pick his favorite girl that remains by the time his turn comes up. Running the greedy algorithm on our example, boy B1 picks his favorite girl G3, boy B2 picks his favorite girl G1, boy B3 picks his favorite girl G4, boy B4 picks his favorite remaining girl, G2 (since his top 3 choices are already taken), and finally boy B5 picks his favorite remaining girl (which at this point, is the only remaining girl), G5.

Now, let's see if there is a rogue couple. Boys B1, B2, and B3 are matched up with their preferred girls, leaving them satisfied. However, boy B4 is not satisfied with G2, his fourth choice. He approaches G1, but she rejects him since she prefers B2. In fact, she ranked B4 last, making a pairing between them highly undesirable. He runs into his first choice, G3, and she prefers him to B1, who is way down on her list. Evidently, there exists an interesting situation here. Both B4 and G3 prefer each other to their own partners. We could try to patch things up and pair B4 with G3 and then G2 with B1, but it's not clear that we would reduce the number of rogue couples. It happens that in this case pairing up B4 and G3 is a satisfactory action to do. Indeed, this is getting more and more complicated.

How about using an algorithm that is based on induction (or recursion)? Pair boy B1 with girl G3 and solve the rest by induction. By the induction hypothesis, the only rogue couples would involve B1 or G3. However, they can't involve B1, since he got his first choice. On the other hand, they might well involve G3 since B1 might be her last but one choice.

Induction would work if there were some boy and some girl who each ranked the other first. If there were such a boy and girl, then they would have to get paired with each other, or they would be a rogue couple. However, there might not be such a boy and girl—all too often, people do not like those that like them!

5. Gale-Shapley Algorithm

It turns out that finding a good way of pairing up the boys and girls can be tricky. However, the best approach is to use the Gale-Shapley Algorithm, with its method outlined below. The mating ritual takes place over several days (or rounds). The idea is that each of the boys go after the girls one by one, in order of their individual preference, crossing off girls from their list as they get rejected. Provided below is a more detailed specification.

To define initial conditions, each of the n boys have an ordered list of the n girls according to his preferences. Each of the girls has an ordered list of the boys according to her preferences. Every day, in the morning, each girl stands on her balcony. Each boy then stands under the balcony of his favorite girl whom he has not yet crossed off his list and serenades. If there are no girls left on his list, he stays home. In the afternoon, girls who have at least one suitor say to their favorite from among the suitors that day something along the lines of, "I need some time, please come back tomorrow." To the others, they say "No, I will never marry you!" In the evening, any boy who hears "No" crosses that girl off his list and goes home. Only, they will come back the next morning in search of their next option.

So, we can now start applying the Gale-Shapley Algorithm to the preferences shown in Table 1 above. In Round 1, B1 proposes to G3 (his #1), B2 proposes to G1 (his #1), B3 proposes to G4 (his #1), B4 proposes to G1 (his #1), and B5 proposes to G1 (his #1). At this stage, G1 has received 3 proposals from B2, B4, and B5. G1 ranks them in descending order of preference: B5 (2), B2 (3), then B4 (5). So, G1 keeps B5 (her best option) and rejects B2 and B4. The current status is as follows: G1 keeps B5 on hold, G3 keeps B1 on hold, G4 keeps B3 on hold, G2 and G5 have no proposals yet, and B2 and B4 have been rejected and must propose to their respective next choices.

In Round 2, B2 proposes to G2 (his #2) and B4 proposes to G3 (his #2). G3 has now received 2 proposals from B1 and B4. G3 ranks them as B4 (1) then B1 (4). G3 keeps B4 (her best option) and rejects B1. The current status is as follows: G1 keeps B5 on hold, G2 keeps B2 on hold, G3 keeps B4 on hold, G4 keeps B3 on hold, G5 has no proposals yet, and B1 has been rejected and must propose to his next choice.

In Round 3, B1 proposes to G2 (his #2). G2 has received 2 proposals from B2 and B1 and ranks them as B1 (3) then B2 (2). G2 keeps B2 (her best option) and rejects B1. The current status is as follows: G1 keeps B5 on hold, G2 keeps B2 on hold, G3 keeps B4 on hold, G4 keeps B3 on hold, G5 has no proposals yet, and B1 has been rejected again and must propose to his next choice.

In Round 4, B1 proposes to G5 (his #3). The current status is as follows: G1 keeps B5 on hold, G2 keeps B2 on hold, G3 keeps B4 on hold, G4 keeps B3 on hold, and G5 keeps B1 on hold. All girls have received proposals, and each girl now has one unique proposal only, meaning they accept these final proposals and the algorithm terminates.

The final stable matching goes as follows: B1 and G5, B2 and G2, B3 and G4, B4 and G3, and B5 and G1. This is a stable matching because no boy and girl who aren't matched with each other would both prefer each other.

6. Extending for Ties: Modified Algorithm

We propose an extension to handle ties in preferences, with the key modifications going as follows. To start, we introduce a new preference structure: instead of strict rankings, participants can now group their preferences into ties (i.e. Tier 1 equates to most preferred and can contain multiple choices, Tier 2 equates to second most preferred, and so on). Next, we introduce a modified proposal stage. When a participant has ties in their preferences, they can propose to all equally ranked choices simultaneously. If multiple proposals are received, the receiver must then choose one based on preference, with rejected participants returning to the pool. Finally, we introduce an example with ties. For example, the following table summarizes the ranking matrix of 5 boys (B_i) and 5 girls (G_j), where those rankings that are tied by the boys are shown bolded. Without lack of generality, we assume that only the boys are allowed to have ties and not the girls, since in this setting we assume the boys are the ones who propose to the girls and not the other way around.

	G1	G2	G3	G4	G5
B1	**(3, 4)**	(2, 3)	(1, 4)	(5, 1)	**(3, 4)**
B2	**(1, 3)**	**(1, 2)**	(4, 5)	(5, 2)	(3, 3)
B3	**(2, 1)**	(4, 5)	**(2, 2)**	(1, 3)	(5, 2)
B4	(1, 5)	(4, 4)	**(2, 1)**	**(2, 4)**	(5, 1)
B5	(1, 2)	(2, 1)	(5, 3)	(3, 5)	(4, 5)

Table 2: Preference Matrix with Ties Allowed

For example, the cell (B1, G1) is represented by (3, 4), meaning B1ranks G1 as his #3 choice, while G1 ranks B1 as her #4 choice. Let's identify each boy's preferences first and then go through the modified Gale-Shapley algorithm.

In Round 1, B1 proposes to G3 (his #1), B2 proposes to both G1 and G2 (tied #1s), B3 proposes to G4 (his #1), B4 proposes to G1 (his #1), and B5 proposes to G1 (his #1). The current tentative matches after Round 1 goes as follows: G1 keeps B5 (since G1 ranks B5 as #2, better than B2 #3 and B4 #5) and rejects B2 and B4, G2 keeps B2, G3 keeps B1, G4 keeps B3, and G5 is unmatched.

In Round 2, B4 (rejected from G1) must propose to his next choice. B4's next choices are G3 and G4 (tied at #2). However, G3 already has B1 (whom she ranks as #4), and G4 already has B3 (whom she ranks as #3). G4 ranks B4 as #4, so she keeps B3. Notably though, G3 ranks B4 as #1, so she accepts B4 and releases B1.

In Round 3, B1 (rejected from G3) must propose to his next choice. B1 proposes to G2 (his #2), where G2 ranks B1 as #3 versus the current B2 whom she ranks as #2. Therefore, G2 keeps B2.

In Round 4, B1 must now propose to either G1 or G5 (tied at #3). G1 has B5 (whom she ranks #2) and G5 is unmatched. Logically, B1 proposes to G5, and they match.

The final stable matching goes as follows: B1 and G5, B2 and G2, B3 and G4, B4 and G3, and B5 and G1.

7. Algorithm Properties with Ties

7.1 Termination

We now provide a formal proof that the modified Gale-Shapley algorithm with preference ties terminates after a finite number of rounds.

Theorem 1: The modified Gale-Shapley algorithm with preference ties terminates after at most $(n^2 + n)$ steps, where n is the number of participants on each side.

Proof:

We begin by defining some key variables to track the algorithm's progress. Let $P(t)$ be the set of all pairs (b_i, g_j) such that boy B_1 has proposed to girl G_j by the end of round t. Let $R(t)$ be the total number of rejections that have occurred by the end of round t.

First, we observe that for any boy B_i and girl G_j, the pair (b_i, g_j) can enter the set P at most once during the algorithm's execution, as boys never propose to the same girl twice. Since there are n boys and n girls, $|P|$ is bounded above by n^2. In each round of the algorithm, at least one of the following must occur: either a new proposal is made (i.e., $|P(t+1)| > |P(t)|$) or a girl switches from one boy to another, causing a rejection.

When a boy has ties in his preference list, he may make multiple proposals simultaneously. Let k be the maximum number of tied preferences for any boy. In the worst case, a boy proposes to k girls in a single round. However, this still only contributes k elements to P, and each boy can make at most n proposals in total across all rounds.

For termination, we need to bound the number of rejections. When a girl receives multiple proposals, including one from a boy she currently holds, she can reject at most $n - 1$ boys in a single round. Each rejection leads to a new proposal in a subsequent round, or the boy has exhausted his list.

The key insight is that the sum $|P(t)| + R(t)$ strictly increases with each round until termination. Since $|p| \leq n^2$ and each rejection leads to one fewer potential future proposal, the maximum number of rejections is also bounded by n^2.

Therefore, the total number of rounds is bounded by n^2 (for proposals) + n (for the initial round of simultaneous proposals). To account for ties, we observe that while a boy may make multiple simultaneous proposals, this doesn't increase the upper bound on total proposals beyond n^2. At most, it front-loads some proposals that would have occurred in later rounds in the classical algorithm.

Finally, we need to consider the case where a girl with tied preferences receives multiple acceptances and must choose one. This selection step adds at most one additional operation per round but doesn't affect the asymptotic bound on the number of rounds. Therefore, the modified algorithm terminates after at most $(n^2 + n)$ steps, with the additional n term accounting for potential overhead from handling ties.

Corollary 1: If each boy has at most one tie in his preference list (as specified in our constraint), and each tie involves only two girls, the algorithm terminates in $O(n^2)$.

This modified algorithm converges more slowly than the classical algorithm in the worst case but still maintains polynomial time complexity. The introduction of ties changes the nature of the termination condition: rather than merely waiting until every girl has at most one suitor, we must also resolve situations where boys with ties must select among multiple accepting girls.

7.2 Stability

With this in mind, the definition of stability needs modification. A matching is now stable if there exists no pair (B_i, G_j) where B_i strictly prefers G_j to their current match, G_j strictly prefers B_i to their current match, or neither is in a tie situation with their current match. We now provide a formal proof that the modified Gale-Shapley algorithm with preference ties produces stable matchings according to our extended definition of stability.

To start, we provide two key definitions. First, for a matching M with ties in preferences, we define the following: for a boy b and girls g and $g^{'}$, we write $g >^{b} g'$ if b strictly prefers g over g' and $g =^{b} g'$ if b is indifferent between g and $g^{'}$. Similarly, for a girl g and boys b and b', we write $b >^{g} b'$ if g strictly prefers b to b', and $b =^{g} b'$ if g is indifferent between b and b'.

Second, a matching M is stable under preference ties if there exists no pair (b, g) such that b is matched to some $g^{'}$ in M, and $g >^{b} g'$, and g is matched to some b' in M and $b >^{g} b'$. We refer to such a pair (b, g) as a "blocking pair" or "rogue couple."

Theorem 2: The modified Gale-Shapley algorithm with preference ties produces a stable matching.

Proof:

We proceed by contradiction. Suppose the algorithm terminates with a matching M that is not stable. Then there exists a blocking pair (b, g) where boy b is matched to girl $g^{'}$ in M, and b strictly prefers g to $g^{'}$ $g >^{b} g'$, and girl g is matched to boy b' in M, and g strictly prefers b to b' $b >^{g} b'$. We consider the execution history of the algorithm to derive a contradiction, as follows.

Case 1: Boy b proposed to girl g during the algorithm. Since b is not matched with g in the final matching M, girl g must have rejected b at some point. By the algorithm's design, g rejects b only if she receives a proposal from some boy b" whom she strictly prefers to b, or if she already holds a proposal from such a boy. Since the algorithm ensures that a girl's match quality never decreases (she only switches from boy b_1 to boy b_2 if $b_2 >^{g} b_1$, the girl g's final match b' must satisfy $b' \geq^{g} b" >^{g} b$. This contradicts our assumption that $b >^{g} b'$.

Case 2: Boy b did not propose to girl g during the algorithm. Since the algorithm requires boys to propose in decreasing order of preference (possibly with simultaneous proposals for tied preferences), the fact that b did not propose to g implies either boy b obtained a match with some girl g" where $g" =^{b} g$ (a tie) before he would have proposed to g, or boy b strictly prefers g' to g $(g' >^{b} g)$.

Subcase 1: If $g" =^{b} g$, then b must have chosen $g^{'}$ over g" when both accepted his proposals, or $g^{'}$ is the same as g". However, this means b does not strictly prefer g to $g^{'}$, contradicting our assumption that $g >^{b} g'$.

Subcase 2: If $g' >^{b} g$, this directly contradicts our assumption that $g >^{b} g'$.

Therefore, no blocking pair can exist, and the matching produced by the modified Gale-Shapley algorithm is stable.

Theorem 3: When boys have ties in their preference lists but girls have strict preferences, the modified Gale-Shapley algorithm produces a boy-optimal stable matching among all possible stable matchings.

Proof:

Let M be the matching produced by the modified algorithm and let M' be any other stable matching. We need to show that no boy is worse off in M than in M'. Now, we define the following sets for each round t of the algorithm. Let $Q(t)$ be the set of ordered pairs (b, g) where boy b has been rejected by girl g by round t. Next, let $A(t)$ be the set of ordered pairs (b, g) where girl g has accepted (at least temporarily) a proposal from boy b by round t. We claim that for any stable matching M', if $(b, g) \in Q(t)$ for any t, then $(b, g) \notin M'$. We prove this by induction on t. We define the base case where at $t = 0, Q(0) = \emptyset$, with this claim holding vacuously.

Inductive step: Assume the claim holds for $Q(t)$. Consider any rejection that occurs to form $Q(t+1)$. If girl g rejects boy b, it must be because she accepted a boy b' whom she strictly prefers $(b' >^g b)$.

Now, suppose for contradiction that $(b, g) \in M'$. Then in M', girl g is matched to b, while in the algorithm at round $t+1$, she prefers b'. If b' is not matched to a girl he strictly prefers to g in M', then (b', g) would form a blocking pair for M', contradicting the stability of M'. Therefore, in M', boy b' must be matched with some girl g' where $g' \geq^b g$.

By our induction hypothesis, b' has not been rejected by any girl he strictly prefers to g' by round t. Given the algorithm's behavior with ties, this means that either $g' >^b g$ (contradicting our assumption that b ended up proposing to g), or $g' =^b g$ (implying that b' would have proposed to both simultaneously).

In the latter case of a tie, we need to consider how b' chose among multiple accepting girls. The precise choice mechanism becomes relevant here but doesn't affect the overall optimality result, as the preferred choices among tied options are considered equivalent from the boy's perspective. Therefore, each boy in the modified algorithm obtains a match that is at least as good as his match in any other stable matching, proving the boy-optimality of the resulting matching.

Corollary 2: The presence of ties in boys' preferences produces a set of boy-optimal matchings rather than a unique boy-optimal matching. The specific outcome depends on how boys choose when multiple equally ranked girls accept their proposals. This rigorous analysis establishes that our modified algorithm preserves the critical stability property of the original Gale-Shapley algorithm, while accommodating the more realistic scenario of preference ties. The extension maintains most structural properties of the classical result while introducing new insights into the potential multiplicity of boy-optimal matchings under indifference.

7.3 Optimality

The boy-optimal property of the original algorithm becomes more complex, as multiple stable matchings may exist with equal optimality and the choice made by proposers in tie situations affects the final outcome.

The optimality properties of the modified Gale-Shapley algorithm with preference ties require careful analysis. In the classical algorithm without ties, a key result establishes that the algorithm produces a boy-optimal stable matching—one where each boy receives the best partner he could obtain in any stable matching. However, the introduction of ties fundamentally alters this landscape. When preference ties are allowed, there no longer exists a unique boy-optimal stable matching, but rather a set of potentially incomparable boy-optimal matchings. Consider two stable matchings M_1 and M_2 where a boy b has a tie between two girls, g_1 and g_2. In M_1, he is matched with g_1, while in M_2, he is matched with g_2. Since b is indifferent between these outcomes, both matchings can claim optimality from his perspective.

Mathematically, this creates a partial ordering rather than a total ordering of stable matchings. If we denote the set of all stable matchings as $\mathcal{M}$, then we obtain a set of maximal elements under the partial order defined by boy preference, rather than a unique maximum element. This set, which we denote as $\mathcal{M}_b$, contains all stable matchings that are boy-optimal in at least one possible interpretation of the preference ties.

The choice mechanism employed when a boy receives multiple acceptances from tied preferences becomes crucial in determining which specific boy-optimal matching is produced. For instance, if boys employ a consistent tie-breaking rule (such as lexicographic ordering of girls' names when indifferent), the algorithm will deterministically produce one specific matching from $\mathcal{M}_b$. Alternative tie-breaking strategies can yield different elements from this set without sacrificing stability or boy-optimality within the framework of ties. This observation has important practical implications: system designers can implement secondary criteria for resolving ties (such as geographical proximity or complementary preferences) while maintaining the fundamental stability and optimality guarantees of the matching algorithm.

8. Real-World Applications with Ties

In many university admissions processes, applicants are often grouped into broad tiers rather than being assigned strict, rank-ordered preferences. For instance, colleges may categorize candidates into groups such as Tier 1 (Outstanding Candidates), Tier 2 (Strong Candidates), and Tier 3 (Acceptable Candidates). This practice reflects the reality that admissions officers frequently view multiple applicants as equally qualified within a given tier. By incorporating preference ties into the stable matching framework, the extended algorithm provides a more realistic approach to modeling the admissions process, allowing for a more equitable distribution of placements while maintaining stability and efficiency.

Similarly, in medical residency programs, hospitals evaluate groups of final-year candidates as equally qualified, particularly when assessing applicants from the same academic institutions or similar training backgrounds. Traditional stable matching mechanisms require strict rankings, which may not align with real-world selection processes where hospitals consider multiple applicants as interchangeable within certain categories. By enabling preference ties, our modified algorithm offers a more flexible and practical approach to residency matching, reducing the need for arbitrary tie-breaking while ensuring that stability and fairness are preserved in the final placements.

9. Conclusion

The extension of the Stable Marriage Algorithm to handle ties enhances its applicability to real-world scenarios where strict preference rankings may be impractical or unrealistic. Many real-world matching problems, including college admissions, medical residency placements, and job markets, involve cases where decision-makers consider multiple candidates as equally desirable. By allowing for preference ties and modifying the Gale-Shapley algorithm accordingly, we provide a more flexible and inclusive solution while still preserving termination, stability, and fairness.

Although the introduction of ties increases computational complexity, our modified algorithm maintains a polynomial runtime and ensures stable matchings without introducing unnecessary tie-breaking mechanisms. This makes it particularly valuable for large-scale matching systems where fairness and efficiency are equally critical. Furthermore, this extension opens up new avenues for future research, such as exploring the impact of bidirectional ties (allowing both sides to have preference ties), analyzing alternative tie-breaking strategies, and applying the model to more complex multi-agent matching markets.

Overall, this study contributes to the ongoing evolution of matching theory by bridging the gap between theoretical algorithms and practical real-world applications. By refining existing models to better reflect actual decision-making processes, we can continue to improve the efficiency and fairness of matching mechanisms across various domains.

References:

1. Gale, D., & Shapley, L. S. (1962). College Admissions and the Stability of Marriage. The American Mathematical Monthly, 69(1), 9-15. https://doi.org/10.2307/2312726

2. Irving, R. W. (1994). Stable Marriage and Indifference. Discrete Applied Mathematics, 48(3), 261-272. https://doi.org/10.1016/0166-218X(92)00179-P

3. Manlove, D. F. (2013). Algorithmics of Matching Under Preferences. World Scientific Publishing.

4. Seminario, E. (2018). Stable Marriage Problem. Università di Palermo. Retrieved from https://www.unipa.it/dipartimenti/matematicaeinformatica/.content/documenti/2018_Seminario_Erasmus_Lecture_Stable_Marriage_Problem.pdf