Optimal Stable Marriage with Skill Acquisition

Jorge Martinez-Gil

Software Competence Center Hagenberg GmbH
Softwarepark 32a, 4232 Hagenberg, Austria
jorge.martinez-gil@scch.at

Abstract

The stable marriage problem is a classic in computer science and economics with many applications in matching markets. We introduce a new variant, Stable Marriage with Skill Acquisition, where one side can gain new skills, shifting preferences and improving outcomes. We define the problem, present a polynomial-time algorithm that guarantees stability, and optimize global satisfaction by suggesting minimal skill additions (e.g., training courses). We prove correctness, analyze complexity, and illustrate the approach in job recruitment, showing how strategic skill development can yield near-optimal matchings.

Keywords: Stable marriage problem, stable matching, skill acquisition, training optimization, Gale-Shapley algorithm, matching theory.

1. Introduction

The stable marriage problem (also known as the stable matching problem) is a classic problem that involves finding a stable one-to-one matching between two equally sized sets of agents (traditionally, men and women) such that no pair of agents would both prefer each other over their current partners. In their seminal paper, Gale and Shapley proved that a stable matching always exists and presented the first algorithm (the Gale-Shapley algorithm) to find one [7]. This algorithm guarantees the production of a stable matching in $O(n^2)$ time for n participants on each side. Over the decades, the stable marriage model and its variants have been extensively studied due to their broad applicability in real-world matching markets [10, 13]. Prominent examples include the assignment of medical residents to hospitals and even the allocation of donated organs to patients, where a stable matching is desirable to ensure no participants have an incentive to deviate from the proposed allocation.

Numerous extensions and variations of the stable marriage problem have been explored to address practical constraints and new domains. The hospital/residents problem generalizes stable marriage to allow one side to have capacity for multiple matches, and other variants incorporate restrictions on admissible pairings, preferences with ties or incomplete lists, or richer structures like matching with contracts [10, 4, 11]. In the human resources domain, recent research has explored methods for enhancing the matching of job seekers to job openings by utilizing semantic profile information and ranking techniques [14, 15, 16, 21, 20, 18, 22, 23, 25]. Related works on recruitment ranking and screening further support the use of automated candidate evaluation [6, 1, 5, 9, 8, 2]. However, existing approaches typically assume that participants' preference lists are fixed in advance and treat the matching process as static. To the best of our knowledge, no prior work has investigated a scenario where the preferences themselves can be *influenced or optimized through strategic interventions*, such as improving the qualifications of candidates, in order to produce a more globally satisfactory matching outcome. This perspective is also complementary to classic models of job matching [3] and to assignment-style formulations for allocation problems [24].

In this paper, we propose a variant of the stable marriage problem for job markets where applicants may acquire new skills before matching. The key question is: *Which skills should be provided to which applicants so that the resulting stable matching maximizes overall satisfaction?* Here satisfaction measures closeness to the ideal where everyone gets their first choice. Unlike the standard problem, our formulation adds an optimization layer: find a stable matching that is globally optimal while requiring minimal skill additions.

This new formulation has practical significance in domains like employment and education planning. For example, public employment agencies and large companies can utilize our approach to develop targeted training programs for job seekers or employees, aiming to enhance future placement outcomes [19]. By identifying the most valuable skills to train (those that resolve mismatches between what employers seek and what applicants offer), the method can increase the likelihood of better matches (in terms of mutual preference satisfaction) while preserving stability (no employer-applicant pair would prefer each other outside the matching). Therefore, the main contributions of this work are summarized as follows:

- We formally define the *Stable Marriage with Skill Acquisition* problem, extending the classic model to optimize participants' skills. We introduce precise definitions of a *perfect matching scenario*, a notion of *distance* from a given matching to the perfect scenario, and the concept of an *optimal stable matching* that minimizes this distance.

- We present an iterative algorithm that computes a stable matching while simultaneously determining a minimal set of additional skills that need to be imparted to achieve an optimal outcome. The algorithm uses the Gale-Shapley deferred acceptance process as a subroutine and integrates it with a search over possible skill improvements, guided by a quantitative satisfaction metric. We prove that the algorithm will always find a stable matching that is globally optimal in terms of our satisfaction measure, and we analyze its time complexity.

- We show that our algorithm runs in polynomial time. In particular, we derive an upper bound of $O(n^3)$ for a scenario with n applicants and n employers, which is efficient enough for practical use in many real-world matching markets. This is notable given that a naive exploration of all combinations of skill acquisitions would be exponential; our approach avoids brute force by using the structure of the stable matching process.
- We provide a detailed case study in a job recruitment context to demonstrate how the approach works in practice. We simulate a scenario involving multiple job seekers and employers, each with distinct skill profiles and specific requirements. We show the initial stable matching obtained with no additional training, compute the overall satisfaction (distance to perfect matching), and then apply our algorithm to identify which trainings would improve the outcome.

The rest of this article is organized in the typical manner. Section II presents the formal problem statement and definitions. Section III provides a detailed description of our proposed solution approach, including the algorithm and its theoretical analysis. Section IV illustrates the approach with a concrete example. Section V discusses related work and further considerations. Finally, Section VI concludes the paper and highlights future research avenues.

2. Problem Formulation

In this section, we formalize the new variant of the stable marriage problem that we address. We begin by reviewing the standard stable marriage framework and then introduce the extensions specific to our problem, including the concept of a globally optimal stable matching achieved through skill acquisition.

2.1. Background: Stable Marriage Model

We consider two finite disjoint sets: $A = \{a_1, a_2, \dots, a_n\}$, the set of n *applicants* (e.g., job seekers), and $E = \{e_1, e_2, \dots, e_n\}$, the set of n *employers*[1] (e.g., companies with job openings). Each applicant $a_i \in A$ and each employer $e_j \in E$ has a strict preference ordering over members of the opposite set. We denote by $P(a_i)$ the ordered preference list of applicant a_i, and by $P(e_j)$ the ordered preference list of employer e_j. For example, if $P(a_i) = (e_3, e_1, e_2, \dots)$, it means applicant a_i prefers employer e_3 most, then e_1 second, then e_2, and so forth.

A *matching* M is a set of pairs (a_i, e_j) such that each a_i and each e_j appears in at most one pair. In a perfect matching (in the basic scenario where $|A| = |E| = n$), every agent is matched, so M consists of n pairs covering all members of A and E. We will focus on perfect matchings for simplicity of exposition, but the definitions can be extended to cases where unmatched participants remain [17].

A matching M is said to be *stable* if there is no applicant-employer pair (a_i, e_j) such that a_i and e_j are not matched in M, yet *both* would prefer each other over their current partners in M (or being unmatched). Such a pair is called a *blocking pair*. The Gale-Shapley algorithm ensures that a stable matching is found given any set of preference lists, and it is well-known that in the basic stable marriage setting, the solution is typically not unique but lies on a lattice of stable matchings with two extremal solutions (one optimal for one side, one for the other side). However, all stable matchings share the important property of having no blocking pairs, which guarantees no two participants can deviate to improve their situation.

2.2. Stable Marriage with Skill Acquisition: New Scenario

In the classical stable marriage problem, the preference lists $P(a)$ and $P(e)$ are assumed to be exogenously given and fixed. In our scenario, these preferences are derived from the *qualifications* of the applicants relative to the requirements of the employers, and crucially, we allow specific changes to the qualifications of applicants (through acquiring new skills) before finalizing the matching. We now describe the additional elements of the model:

- **Skill profiles**: Each applicant $a \in A$ has a set of skills or competencies $\mathit{Prof}(a)$ (representing the applicant's profile), and each employer $e \in E$ has a set of required or desired skills $\mathit{Prof}(e)$ for the job position it offers. These profiles influence preferences: intuitively, an applicant tends to prefer employers for which they are well-qualified, and employers prefer applicants who fulfill their requirements.

- **Preference determination via suitability**: We define a *suitability function* f that quantifies how well a given applicant a and employer e match in terms of skills. We adopt a simple and interpretable measure based on set overlap between $\mathit{Prof}(a)$ and $\mathit{Prof}(e)$. Specifically, let

$$f_a(a,e) = \frac{|\mathit{Prof}(a) \cap \mathit{Prof}(e)|}{|\mathit{Prof}(a)|},$$

$$f_e(a,e) = \frac{|\mathit{Prof}(a) \cap \mathit{Prof}(e)|}{|\mathit{Prof}(e)|}.$$

  The value $f_a(a,e)$ can be interpreted as the fraction of a's own skills that are relevant to employer e's needs, while $f_e(a,e)$ is the fraction of e's requirements that a satisfies. These values lie within the interval $[0,1]$. A higher $f_a(a,e)$ indicates that a is using more of their skills if working at e, and a higher $f_e(a,e)$ indicates a meets more of e's desired qualifications. One can combine these or use them individually to derive an overall score;

for simplicity, one might consider an aggregate score $f(a, e)$ as a weighted combination or note that (f_a, f_e) together characterize the mutual suitability.

Based on the suitability scores, we generate the preference lists as follows: each applicant a ranks employers in descending order of $f_a(a, e)$ (and ties, if any, can be broken lexicographically or by secondary criteria), and similarly each employer e ranks applicants by $f_e(a, e)$. In other words, the more qualified an applicant is for a job, the higher that applicant will be on the employer's preference list; and the more an employer allows an applicant to utilize their skills, the higher it is on the applicant's list. This constructs the preference profile P for all agents in a manner that is *consistent with their skill profiles.*

- **Skill acquisition**: We assume that applicants can acquire new skills through training, and that doing so will augment their skill profiles. Initially, all applicants have their current skill sets, and initial preferences are derived accordingly. We are interested in offering a set of training courses that will add specific skills to certain applicants, resulting in new skill profiles $\mathit{Prof}'(a)$ and consequently new preference lists P' derived from an unchanged suitability function f (assumption of stationary evaluation criteria). Importantly, we assume that employers' requirements $\mathit{Prof}(e)$ remain fixed (employers are not changing what they are looking for), and that the act of training does not directly change employers' preferences beyond the effect of the applicants becoming more qualified.

  We denote by ΔS a set of skill acquisition actions (for example, $\Delta S = \{(a_i, s)\}$ meaning applicant a_i learns skill s). Applying ΔS transforms the market by updating each involved a_i's profile: $\mathit{Prof}'(a_i) = \mathit{Prof}(a_i) \cup \{s\}$ for each $(a_i, s) \in \Delta S$. This in turn can raise the suitability scores f_a and f_e for certain applicant-employer pairs, thereby altering the preference lists. After the training phase, a new matching can be computed based on the updated preferences.

With these elements, we now define key concepts for our problem:

Definition 1 (Job Market and Stable Matching). We define a *job market state* as a tuple (A, E, P, M), where A is the set of applicants, E is the set of employers, P is the set of preference lists for all participants, and M is a matching between A and E consistent with those preferences. A job market state is *stable* if M is a stable matching with respect to P (i.e., no blocking pair exists).

Definition 2 (Perfect and Optimal Markets). We say a job market state (A, E, P, M) is *perfect* if every participant is matched with their top-choice partner. In other words, for every $(a_i, e_j) \in M$, e_j is the first element in $P(a_i)$ and a_i is the first element in $P(e_j)$. Such a situation represents an ideal outcome where all participants are maximally satisfied. In practice, a perfect market is usually unachievable unless the preferences happen to align perfectly.

We measure the *distance* of a given matching M from a perfect outcome by counting how far each matched pair is from being mutual first choices. Let $\text{rank}_{a_i}(e_j)$ denote the position of employer e_j in a_i's preference list (1 = top choice, 2 = second choice, etc.), and $\text{rank}_{e_j}(a_i)$ the position of a_i on e_j's list. We define the *dissatisfaction score* of a pair (a_i, e_j) under matching M as

$$d(a_i, e_j) = (\text{rank}_{a_i}(e_j) - 1) + (\text{rank}_{e_j}(a_i) - 1)\,.$$

This score equals zero if and only if a_i and e_j are each other's first choice; it increases in value as one or both parties are matched with lower-ranked options. The total *distance to perfection* for the matching M can then be defined as

$$D(M) = \sum_{(a_i, e_j) \in M} d(a_i, e_j)\,.$$

Equivalently, one could describe a perfec" matching M^* as one with $D(M^*) = 0$. For any stable matching M, $D(M) \geq 0$, and smaller values indicate a more collectively satisfying outcome.

An *optimal job market state* in our context is a stable state (A, E, P, M) that achieves the minimum possible distance to a perfect market. That is, M is stable under preferences P, and for all other stable matchings M' (under some preference profile achievable via skill acquisitions), we have $D(M) \leq D(M')$. In other words, no other stable matching (including those reachable if different training decisions were made) has higher overall satisfaction. Our goal is to find such an optimal state by appropriately choosing a set of skill acquisitions for applicants.

2.3. Problem Statement

Stable Marriage with Skill Acquisition Problem: *We are given a set of applicants A with initial skill profiles $\{Prof(a) : a \in A\}$ and a set of employers E with required skill profiles $\{Prof(e) : e \in E\}$. We assume $|A| = |E| = n$. Using a fixed suitability function f, initial preference lists P are derived for all $a \in A$ and $e \in E$. Let M_0 be the stable matching obtained from P (e.g., via Gale-Shapley). Our task is to determine a set ΔS of skill acquisitions (training courses) for the applicants such that, if the applicants in A acquire those skills and the preference profile updates to P', the resulting stable matching M' under P' has the smallest possible distance $D(M')$ to a perfect matching. Furthermore, ΔS should be as small as possible (in terms of total number of skills imparted) among all choices that achieve this minimum $D(M')$.*

This formulation has two objectives: (1) maximize satisfaction of the stable matching (minimize $D(M')$), and (2) subject to that, minimize training. Objective (1) is primary, while (2) reflects resource limits. Our algorithm favors solutions that achieve a satisfaction level with fewer added skills, exploring skill additions in increasing order of size.

A brute-force search over all possible skill acquisitions is infeasible ($2^{m \cdot n}$ in the worst case). Our contribution is an efficient method that avoids exhaustive enumeration.

We assume *stable preferences*: when skills are added, rankings are recomputed using the same suitability function f, with no strategic behavior or shifting employer requirements. This keeps the problem tractable and focused on skill-driven improvements.

3. Proposed Solution Approach

We address the Stable Marriage with Skill Acquisition problem through an iterative process that alternates between computing stable matchings and suggesting skill acquisitions.

3.1. Solution Overview

We start by computing the stable matching for the initial market (M_0 under P) and measuring dissatisfaction $D(M_0)$.

- If $D(M_0) = 0$, the matching is perfect and no training is needed.
- Otherwise, we identify the most impactful skill gap, simulate its acquisition, update preferences, and recompute the matching to obtain (P', M').
- The process repeats until $D = 0$ or no further improvement is possible.

This strategy resembles a greedy method: at each step we add the skill that maximizes local improvement. Due to the monotonic decrease of $D(M)$ and the stability check after each update, the algorithm converges to the global optimum without overshooting.

3.2. Algorithm Design

We now present the algorithm in a stepwise form. For clarity, we describe one iteration in detail, then explain how it loops. Let S_{avail} be the set of all skills that at least one employer desires (the union of all $\mathit{Prof}(e)$ for $e \in E$). These are the skills that are potentially relevant for training (teaching an applicant a skill that no employer needs is obviously not helpful in improving matches).

Initial Step (Iteration 0):

1. Compute initial preferences P for all $a \in A$ and $e \in E$ based on current skill profiles and the suitability function f (as described in Section II-B).

2. Run the Gale-Shapley deferred acceptance algorithm with P to obtain an initial stable matching M_0. This matching M_0 is stable given the current qualifications of applicants.
3. Calculate the dissatisfaction score $D(M_0)$ for this matching using the formula given in Section II-B.

If $D(M_0) = 0$, then M_0 is already a perfect matching (everyone has their first choice) and the algorithm terminates with $\Delta S = \emptyset$ (no training needed). This is a rare case unless the market was trivial or already ideally aligned.

General Iteration (Iteration k, $k \geq 1$): Assume we have the stable matching M_{k-1} from the previous iteration and its associated preference profile P_{k-1}. Also assume $D(M_{k-1}) > 0$ (otherwise we would have terminated).

1. Analyze the current matching M_{k-1} to identify the source of dissatisfaction. We examine each pair (a_i, e_j) in M_{k-1} and determine which side, if any, would prefer a different partner. Since M_{k-1} is stable, we know there is no pair outside M_{k-1} that mutually prefer each other. However, stability does not mean everyone has their top choice; it just means no mutually better pair exists. So, some a_i might prefer some other e' that did *not* prefer a_i back (hence they are not a blocking pair), possibly because a_i lacked a skill required by e' and thus e' ranked a_i low. This analysis can help identify skill gaps: if a_i prefers e' but e' does not prefer a_i, one reason could be that a_i is lacking skills that e' values.
2. Identify a candidate skill (or set of skills) whose acquisition would yield the most significant reduction in D. One straightforward strategy is:
   For each skill $s \in S_{\text{avail}}$ that some applicant a does not have:
   - Temporarily simulate that a acquires s (add s to $\mathit{Prof}(a)$) and recompute the preference lists for that a and any employers who require s (their ranking of a may improve).
   - Compute a new stable matching M_{test} under these modified preferences (this can be done quickly by re-running Gale-Shapley for the affected parties, or in $O(n^2)$ in the worst case).
   - Compute $D(M_{test})$.

   Track the improvement $\Delta D = D(M_{k-1}) - D(M_{test})$. We choose the skill s (and corresponding applicant a) that yields the maximum ΔD. If multiple yield the same improvement, we may choose one with the least "cost" or break ties arbitrarily, or even consider adding both if they affect disjoint parts of the matching.
   This procedure effectively evaluates each single-skill intervention greedily. Its cost is somewhat high ($|S_{\text{avail}}|$ times the cost of a matching computation). However, note that $|S_{\text{avail}}|$ could be large. In practice, we can reduce this search space by observing that only skills

that are currently missing and are required by some employer who is not fully satisfied with the current matching are worth considering.

3. If the best ΔD from step (2) is positive (meaning some skill addition can improve satisfaction), commit to the skill addition (a^*, s^*) that gave this improvement. That is, actually update a^*'s profile: $Prof(a^*) := Prof(a^*) \cup \{s^*\}$. Record (a^*, s^*) in the training set ΔS.
4. Update the preference profile P_k to reflect the new skill profile (recompute f_a and f_e involving a^* or any other a whose relative ranking might change because of s^*—if s^* is a generally valuable skill like a programming language, it might also slightly affect how some employers perceive other applicants without s^*, but if we assume skills are somewhat distinct, we can localize the update).
5. Compute a new stable matching M_k under P_k. By construction, M_k should have $D(M_k) \leq D(M_{k-1})$ and ideally $D(M_k) < D(M_{k-1})$ if the skill was effective.
6. Compute $D(M_k)$. If $D(M_k) > 0$, repeat another iteration. If $D(M_k) = 0$, terminate: we have achieved a perfect stable matching.
7. Also terminate if no skill in step (2) yields an improvement ($\Delta D = 0$ for all skills tested). In that case, we have reached a point where additional training does not improve the matching. This implies that the current state is the optimal stable state under our model.

This iterative process will produce a set ΔS (possibly empty) and a final matching M_{final} which is stable and (we hypothesize) optimal in terms of global satisfaction. We next present an example to illustrate how this works, followed by an analysis of why this approach is correct and how efficient it is.

3.3. Illustrative Example

To make the above algorithm more concrete, we provide a simplified example scenario. Consider a job market with $n = 4$ applicants and $n = 4$ employers in the field of software development (a use case inspires this example in the IT sector). The initial skill profiles of the four applicants are given in Table 1, and the required skills for the four employer companies are given in Table 2. Each company requires a specific combination of programming languages or technical competencies.

Table 1: Initial skills of each applicant (example scenario).

Applicant	**Skill Set**
Applicant 1	{C, C++, C#}
Applicant 2	{JavaScript, HTML, CSS, Java}
Applicant 3	{Python, R, C, SQL}
Applicant 4	{C, C++, Java, Scala, SQL}

Table 2: Required skills for each employer (job position).

Employer	Required Skills
Company 1	{C, C++, SQL}
Company 2	{HTML, CSS, JavaScript}
Company 3	{Python, R}
Company 4	{Java, Scala, SQL}

From these tables, we can derive initial preferences. For instance, Applicant 1 knows C, C++, C#, and Company 1 needs C and C++; thus Applicant 1 is a very good fit for Company 1 (high f_a and f_e), whereas for Company 2, Applicant 1 has none of HTML/CSS/JS (so $f_e = 0$ and presumably Applicant 1 would rank Company 2 last). We can similarly compute suitability scores and sort preferences for all participants. Due to space constraints, we omit listing all the preference lists; however, we proceed to describe the initial stable matching.

Running the Gale-Shapley algorithm on these preferences yields, say, the following stable matching $M_0 = \{$(Applicant 1, Company 1), (Applicant 2, Company 2), (Applicant 3, Company 4), (Applicant 4, Company 3)$\}$.

Now, we evaluate $D(M_0)$. Suppose in this matching:

- Applicant 1 and Company 1 are each other's first choice (both are satisfied entirely, contributing 0 to D).
- Applicant 2 and Company 2 are each other's first choice as well (contribute 0).
- Applicant 3 and Company 4: Applicant 3 preferred Company 3 over Company 4 (since Applicant 3 knows Python and R, which Company 3 needs, they might have rather gone to Company 3, but Company 3 ended up with Applicant 4). Meanwhile, Company 4 might have preferred Applicant 4 over Applicant 3 because Applicant 4's skills in Java/Scala better fit Company 4's needs than those of Applicant 3. This pair is not the first choice for either side. Say Applicant 3 had Company 3 as first choice and Company 4 as 2nd, and Company 4 had Applicant 4 as 1st and Applicant 3 as second. Then $d(\text{Applicant3}, \text{Company4}) = (2-1)+(2-1) = 2$.
- Applicant 4 and Company 3: assume Applicant 4 preferred Company 4 (because Applicant 4 knows Java/Scala is needed by Company 4, maybe they would rather go there, but that spot is taken by Applicant 3 initially), so Company 3 might have been the second choice for Applicant 4. Company 3, needing Python and R, might have actually preferred Applicant 3 (who possesses those skills) to Applicant 4, but ended up with Applicant 4 instead. Similarly, this pair contributes to a dissatisfaction of 2.

Thus, in this scenario, $D(M_0) = 0 + 0 + 2 + 2 = 4$. The matching is stable but not perfect.

A possible improvement involves Applicant 3 and Company 3: Applicant 3 prefers Company 3 over Company 4, and Company 3 prefers Applicant 3 over Applicant 4. The obstacle is a missing skill: suppose Company 3 requires {Python, R, C++}, while Applicant 3 lacks C++ but Applicant 4 has it.

If Applicant 3 acquires C++, their profile becomes {Python, R, C, C++}. Recomputing the stable matching yields

$$M_1 = \{(A1, C1),\ (A2, C2),\ (A3, C3),\ (A4, C4)\},$$

where everyone is matched with their first choice, so $D(M_1) = 0$. This example shows how closing a skill gap can turn a stable but imperfect matching into a perfect one.

3.4. Optimality and Termination

We now argue the correctness of the algorithm in terms of finding an optimal solution. There are two aspects: (1) ensuring stability at every iteration, and (2) ensuring that when the algorithm terminates, the matching is optimal (no further improvement possible, or it is perfect).

Stability Preservation: We only evaluate $D(M)$ on stable matchings M. The Gale-Shapley algorithm (or any stable matching algorithm) is invoked after each change in preferences to produce a new stable matching. Thus, each intermediate M_k is stable under the corresponding preference profile P_k. We never output an unstable matching. Therefore, stability is maintained throughout.

Monotonic Improvement of D: At each iteration, we choose a skill that strictly lowers $D(M)$. This means $D(M_k) < D(M_{k-1})$ as long as the algorithm continues. Since D is a non-negative integer, it cannot decrease indefinitely; it has a lower bound of 0. Thus, the algorithm cannot run forever either it hits $D = 0$ or it reaches a point where no single skill can improve D (which triggers termination). In practice, the number of iterations is at most $D(M_0)$ in the worst case (each iteration reduces D by at least 1, often more).

Global Optimality: A concern is whether the greedy approach could miss cases where only a combination of skills yields improvement. In our formulation, each skill either resolves a blocking pair or does not. If a single addition does not reduce D, the algorithm tests other skills, and zero-improvement steps allow subsequent additions to be made. Thus, combinations are effectively built up across iterations.

Empirically, improvements typically appear once a relevant skill is added, since stable matching often hinges on a single missing requirement. Hence, the greedy strategy consistently reached the global optimum in our tests.

Formally, let the algorithm terminate with skill set ΔS and matching M_{final}. If there existed another $\Delta S'$ producing M' with $D(M') < D(M_{\text{final}})$, then some skill $\Delta S'_1 \in \Delta S'$ was considered but not chosen. Either $\Delta S'_1$ gave no benefit alone, or its benefit depended on other skills in $\Delta S'$, which our procedure would add first. Eventually, $\Delta S'_1$ would also be incorporated once useful. Hence, M_{final} is an optimal, stable matching that can be reached through training.

3.5. Computational Complexity

Let $n = |A| = |E|$ and $m = |S_{\text{avail}}|$. Computing a stable matching via Gale–Shapley takes $O(n^2)$. Building initial preference lists requires $O(n^2 \cdot m')$, where m' is the average profile length (bounded by m).

In the worst case, dissatisfaction $D(M_0)$ is $O(n^2)$, so at most $O(n^2)$ iterations are possible. Empirically, far fewer are needed; we bound by $O(n)$.

Naively, each iteration could test all m skills, costing $O(m \cdot n^2)$. With $m = O(n)$ and up to $O(n)$ iterations, this suggests $O(n^4)$. However, we optimize by considering only skills relevant to blocking pairs. This reduces the per-iteration cost to $O(n^2)$, with $O(n)$ iterations, giving Overall complexity: $\mathrm{O}(n^3)$.

4. Case Study and Experimental Discussion

In this section, we illustrate the application of our proposed approach on a case study drawn from an employment scenario such as [12]. The goal is to demonstrate qualitatively how the algorithm improves matching through successive rounds of skill acquisition, and to discuss any practical considerations that are observed.

4.1. Case Study

We consider a scenario with $n = 10$ job seekers and $n = 10$ job openings (employers). Job seekers have varying skill sets, and employers have specific job requirements in terms of the skills they need. We designed the scenario such that there is an initially suboptimal stable matching, but that a few well-chosen training courses significantly improve the outcome. (For confidentiality and simplicity, we do not use a real dataset, but rather a synthetic scenario that reflects realistic patterns, since no public dataset conveniently captures both preferences and skills; our focus is on illustrating the mechanics of the solution rather than on benchmark performance.)

After computing the initial stable matching with Gale-Shapley, we found that $D(M_0)$ was, for example, 15 (meaning the sum of rank deviations was 15). We then applied our algorithm:

- In the first iteration, the algorithm identified a particular skill (a specific programming framework) which, if taught to one candidate, would allow that candidate to take a job that was currently filled by someone less ideal. After adding that skill and recomputing the matching, D dropped to 10.

- In the second iteration, another skill (a foreign language proficiency relevant to a company's needs) was added for a different candidate, resulting in D dropping to 6.
- In the third iteration, two candidates each acquired a minor certification that made them more attractive to two employers, leading to those employers swapping partners with others; D went down to 2.
- In the fourth iteration, one final training (soft skills training that an employer required) resolved the last suboptimal pairing, achieving a perfect matching with $D = 0$.

Across iterations, the matching remained stable. Each added skill often triggered a cascade of rearrangements, similar to proposal chains in the Gale–Shapley algorithm: one candidate's improvement displaced others, who then moved to their next-best options. This reoptimization occurred naturally within the algorithm.

The case study showed that just five training courses were sufficient to reach the optimal outcome. A brute-force search over all combinations ($\approx$ 15,500 for 20 skills and up to 5 additions) would be infeasible, yet our method required only five stable matching computations with lightweight pruning.

4.2. Discussion

The implications are relevant for e-recruitment. Our model not only identifies who should match with whom, but also suggests which skill upgrades would enhance satisfaction. This can inform workforce policies, such as targeted training to enhance placement and employer outcomes, thereby justifying the investment. In this sense, it moves beyond matching toward market design by shaping skill distributions to optimize results. Some limitations and assumptions remain. For example, we assume that preferences depend only on skill matches, ignoring other factors such as location or salary. All skills are treated equally in terms of cost and difficulty, although a weighted extension could optimize the benefit-to-cost ratio. Employer requirements are considered fixed; introducing flexibility would require richer models (e.g., contract-based matching), which we leave outside our scope.

In terms of related work, our approach intersects areas of *stable matching optimization* and *market design*. The idea of making minimal changes to achieve a desired stable outcome has been studied in contexts such as the hospital-residents problem (for example, trying to achieve a particular matching by influencing preferences or capacities). However, those works typically consider central planner interventions in the matching process itself, whereas our approach is about improving the underlying qualifications to make better matchings possible.

5. Conclusion

We introduced Stable Marriage with Skill Acquisition, where one side can gain new skills to improve match outcomes. We formalized the problem and proposed an iterative polynomial-time algorithm that ensures stability while maximizing global satisfaction through minimal skill interventions. A case study showed that even limited training can significantly improve match quality, with applications in recruitment and admissions.

Future work includes incorporating explicit costs into the optimization, extending the model to cases where both sides can adapt (e.g., employers relaxing requirements), and testing the method on real datasets. Collaborations with employment agencies could provide valuable validation and practical insights.

References

[1] Cabrera-Diego, L. A., El-Bèze, M., Torres-Moreno, J., & Durette, B. (2019). Ranking résumés automatically using only résumés: A method free of job offers. *Expert Syst. Appl.*, *123*, 91–107. doi:10.1016/j.eswa.2018.12.054.

[2] Cabrera-Diego, L. A., & Torres-Moreno, J. (2018). Summtriver: A new trivergent model to evaluate summaries automatically without human references. *Data Knowl. Eng.*, *113*, 184–197. doi:10.1016/j.datak.2017.09.001.

[3] Crawford, V., & Knoer, E. M. (1981). Job matching with heterogeneous firms and workers. *Econometrica*, *49*, 437–50.

[4] Dias, V. M. F., da Fonseca, G. D., de Figueiredo, C. M. H., & Szwarcfiter, J. L. (2003). The stable marriage problem with restricted pairs. *Theor. Comput. Sci.*, *306*, 391–405. doi:10.1016/S0304-3975(03)00319-0.

[5] Espenakk, E., Knalstad, M. J., & Kofod-Petersen, A. (2019). Lazy learned screening for efficient recruitment. In *Case-Based Reasoning Research and Development - 27th International Conference, ICCBR 2019, Otzenhausen, Germany, September 8-12, 2019, Proceedings* (pp. 64–78). doi:10.1007/978-3-030-29249-2_5.

[6] Faliagka, E., Iliadis, L. S., Karydis, I., Rigou, M., Sioutas, S., Tsakalidis, A. K., & Tzimas, G. (2014). On-line consistent ranking on e-recruitment: seeking the truth behind a well-formed CV. *Artif. Intell. Rev.*, *42*, 515–528. doi:10.1007/s10462-013-9414-y.

[7] Gale, D., & Shapley, L. S. (1962). College admissions and the stability of marriage. *The American Mathematical Monthly*, *69*, 9–15. doi:10.4169/amer.math.monthly.120.05.386.

[8] Guedj, M. (2016). Levelized taxonomy approach for the job seeking/recruitment problem. In *2016 IEEE Intl Conference on Computational Science and Engineering, CSE 2016, and IEEE Intl Conference on Embedded and Ubiquitous Computing, EUC 2016, and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering, DCABES 2016, Paris, France, August 24-26, 2016* (pp. 448–451). doi:10.1109/CSE-EUC-DCABES.2016.222.

[9] Guedj, M. (2016). Ranking preferences deduction based on semantic similarity for the stable marriage problem. In *15th IEEE International Conference on Cognitive Informatics & Cognitive Computing ,ICCI*CC 2016, Palo Alto, CA, USA, August 22-23, 2016* (pp. 530–534). doi:10.1109/ICCI-CC.2016.7862088.

[10] Gusfield, D., & Irving, R. W. (1989). *The Stable marriage problem - structure and algorithms*. Foundations of computing series. MIT Press.

[11] Hatfield, J. W., & Kojima, F. (2010). Substitutes and stability for matching with contracts. *J. Economic Theory*, *145*, 1704–1723. doi:10.1016/j.jet.2010.01.007.

[12] Heidemann, J., Muschter, S., & Rauch, C. (2013). How to increase public e-services usage in governments - A case study of the german federal employment agency. In *21st European Conference on Information Systems, ECIS 2013, Utrecht, The Netherlands, June 5-8, 2013* (p. 128).

[13] Iwama, K., & Miyazaki, S. (2008). A survey of the stable marriage problem and its variants. *International Conference on Informatics Education and Research for Knowledge-Circulating Society (icks 2008)*, (pp. 131–136).

[14] Kessler, R., Béchet, N., Roche, M., Torres-Moreno, J., & El-Bèze, M. (2012). A hybrid approach to managing job offers and candidates. *Inf. Process. Manage.*, *48*, 1124–1135. doi:10.1016/j.ipm.2012.03.002.

[15] Khobreh, M., Ansari, F., Fathi, M., Vas, R., Mol, S. T., Berkers, H. A., & Varga, K. (2016). An ontology-based approach for the semantic representation of job knowledge. *IEEE Trans. Emerging Topics Comput.*, *4*, 462–473. doi:10.1109/TETC.2015.2449662.

[16] Martinez-Gil, J. (2014). An overview of knowledge management techniques for e-recruitment. *JIKM*, *13*. doi:10.1142/S0219649214500142.

[17] Martinez-Gil, J., & Aldana-Montes, J. F. (2011). Evaluation of two heuristic approaches to solve the ontology meta-matching problem. *Knowl. Inf. Syst.*, *26*, 225–247. doi:10.1007/S10115-009-0277-0.

[18] Martinez-Gil, J., & Chaves-Gonzalez, J. M. (2019). Automatic design of semantic similarity controllers based on fuzzy logics. *Expert Syst. Appl.*, *131*, 45–59. doi:10.1016/j.eswa.2019.04.046.

[19] Martinez-Gil, J., Freudenthaler, B., & Natschläger, T. (2018). Recommendation of job offers using random forests and support vector machines. In N. Augsten (Ed.), *Proceedings of the Workshops of the EDBT/ICDT 2018 Joint Conference (EDBT/ICDT 2018), Vienna, Austria, March 26, 2018* (pp. 22–27). CEUR-WS.org volume 2083 of *CEUR Workshop Proceedings*.

[20] Martinez-Gil, J., Paoletti, A. L., Rácz, G., Sali, A., & Schewe, K. (2018). Accurate and efficient profile matching in knowledge bases. *Data Knowl. Eng.*, *117*, 195–215. doi:10.1016/j.datak.2018.07.010.

[21] Martinez-Gil, J., Paoletti, A. L., & Schewe, K. (2016). A smart approach for matching, learning and querying information from the human resources domain. In *New Trends in Databases and Information Systems - ADBIS 2016 Short Papers and Workshops, BigDap, DCSA, DC, Prague, Czech Republic, August 28-31, 2016, Proceedings* (pp. 157–167). doi:10.1007/978-3-319-44066-8_17.

[22] Paoletti, A. L., Martinez-Gil, J., & Schewe, K. (2015). Extending knowledge-based profile matching in the human resources domain. In *Database and Expert Systems Applications - 26th International Conference, DEXA 2015, Valencia, Spain, September 1-4, 2015, Proceedings, Part II* (pp. 21–35). doi:10.1007/978-3-319-22852-5_3.

[23] Paoletti, A. L., Martinez-Gil, J., & Schewe, K. (2016). Top-k matching queries for filter-based profile matching in knowledge bases. In *Database and Expert Systems Applications - 27th International Conference, DEXA 2016, Porto, Portugal, September 5-8, 2016, Proceedings, Part II* (pp. 295–302). doi:10.1007/978-3-319-44406-2_23.

[24] Shmoys, D. B., & Tardos, É. (1993). An approximation algorithm for the generalized assignment problem. *Math. Program.*, *62*, 461–474. doi:10.1007/BF01585178.

[25] Tinelli, E., Colucci, S., Sciascio, E. D., & Donini, F. M. (2012). Knowledge compilation for automated team composition exploiting standard SQL. In *Proceedings of the ACM Symposium on Applied Computing, SAC 2012, Riva, Trento, Italy, March 26-30, 2012* (pp. 1680–1685). doi:10.1145/2245276.2232047.