[All Figures/Images have been created in Microsoft Powerpoint]

4. Techincal Section

4.1 The Gale-Shapely Matching Algorithm

(Talk about motivation for using this algorithm. O(n^2), greedy approach, stability in matches, etc.)

(Also mention the drawbacks, such as the proposal/preference datasets must be completely filled; the number of men has to equal the number of women; NxN datasets. Otherwise, there might not be a stable one-to-one matching for everyone in both parties.)

```
function stableMatching {
    Initialize all m ∈ M and w ∈ W to free
    while ∃ free man m who still has a woman w to propose to {
        w = first woman on m's list to whom m has not yet proposed
        if w is free
          (m, w) become engaged
        else some pair (m', w) already exists
          if w prefers m to m'
            m' becomes free
           (m, w) become engaged
         else
           (m', w) remain engaged
    }
    return all engaged pairs(m, w)
}
```

[Figure 1: Gale-Shapley Matching Algorithm Pseudocode]

(Mention that in our case, the men are candidates and the women are employers. The candidates are the proposers.)

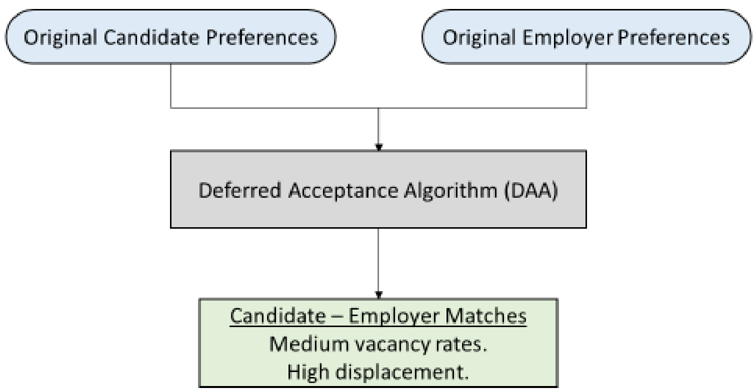
4.2 The Deferred Acceptance Algorithm (DAA)

(Talk about motivation for using DAA. Low complexity and offers multiple matches per person, rather than just one match.)

```
function DAA {
    numberOfMatches = 5 // The maximum number of matches to find.
    matchNumber = 0 // A counter for the current number of matches.
    allMatches // A List to store all matches.
    while preference datasets are not empty or matchNumber >= numberOfMatches {
        matches = stableMatching()
        append matches to allMatches
        for every match in matches
            remove match from original preference datasets.
        matchNumber++
    }
    return allMatches
}
```

[Figure 2: DAA Pseudocode]

(Mention that we only care about the top 5 matches.)



[Figure 3: General Input/Output of Normal DAA]

	Preference of emp1	Preference of emp2	Preference of emp3
cand1	3	1	2
cand2	3	2	1
cand3	1	3	2

Convert Preferences into a Descending List of Preferences. (Sparse to Dense Format)

	Preference 1	Preference 2	Preference 3
cand1	emp2	emp3	emp1
cand2	emp3	emp2	emp1
cand3	emp1	emp3	emp2

	Preference of emp1	Preference of emp2	Preference of emp3
cand1	-----	1	2
cand2	3	-----	-----
cand3	1	-----	2

Convert Preferences into a Descending List of Preferences. (Sparse to Dense Format)

	Preference 1	Preference 2	Preference 3
cand1	emp2	emp3	-----
cand2	emp3	-----	-----
cand3	emp1	emp3	-----

[Figure 4: Converting the preference datasets from sparse format to dense format]

(Show DAA example, see below)

	Preference 1	Preference 2	Preference 3
cand1	emp2	emp3	emp1
cand2	emp3	emp2	emp1
cand3	emp1	emp3	emp2

	Preference 1	Preference 2	Preference 3
emp1	cand3	cand1	cand2
emp2	cand1	cand3	cand2
emp3	cand2	cand3	cand1

Gale–Shapley Matching Algorithm

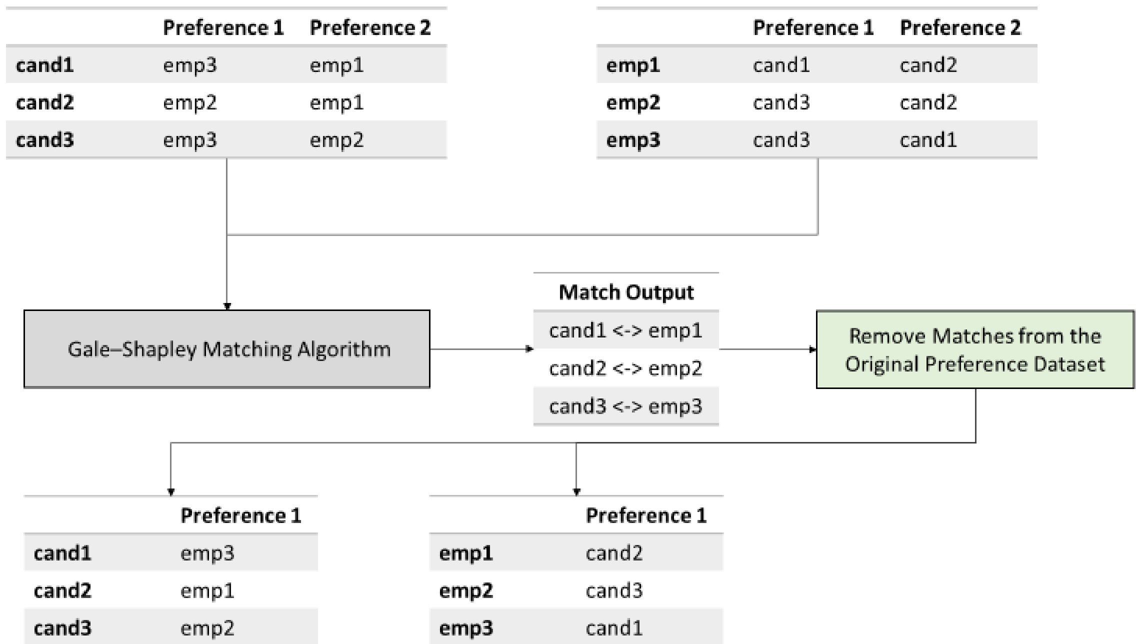
Match Output
cand1 <-> emp2
cand2 <-> emp3
cand3 <-> emp1

Remove Matches from the Original Preference Dataset

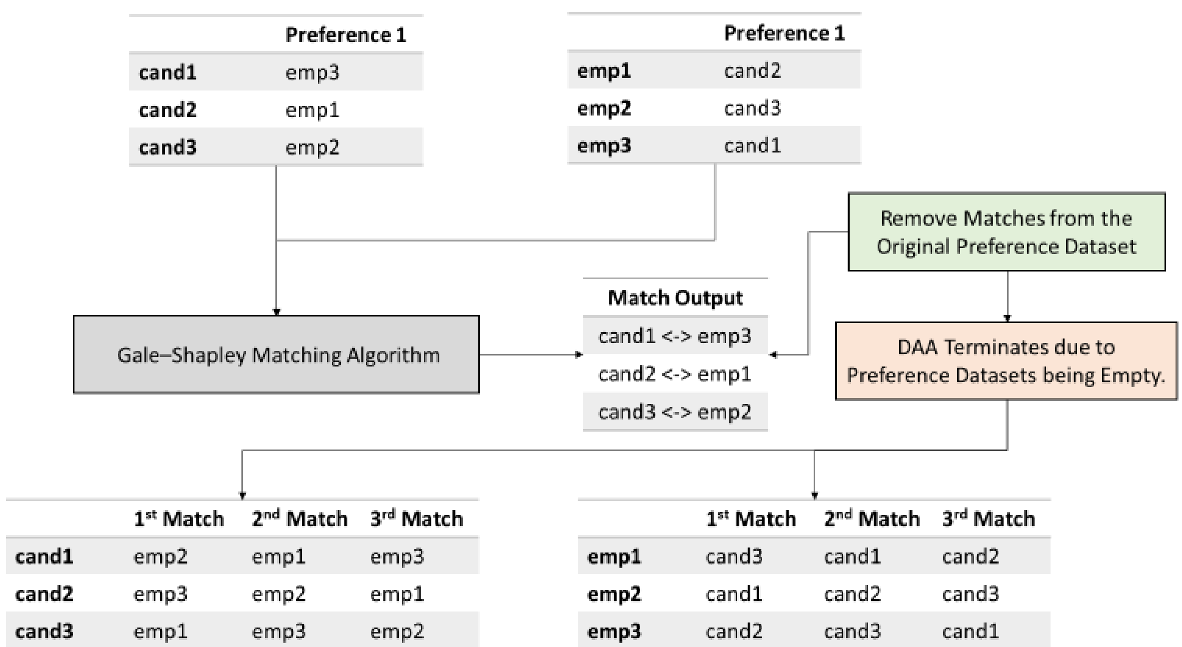
	Preference 1	Preference 2
cand1	emp3	emp1
cand2	emp2	emp1
cand3	emp3	emp2

	Preference 1	Preference 2
emp1	cand1	cand2
emp2	cand3	cand2
emp3	cand3	cand1

[Figure 5: DAA Example, Round 1/3]



[Figure 6: DAA Example, Round 2/3]



[Figure 7: DAA Example, Round 3/3. Also shows all matches]

4.3 Problems with DAA

(Here we want to talk about how and when the DAA fails, mainly due to sparsity in the preference data. The following are some examples of why we have sparsity in the preference datasets: 1) In real world datasets, the number of candidates should be larger than the number of employers. 2) Candidates will not apply for every single employer. 3) Sparsity can get worse as the DAA proceeds to later rounds, since we are removing preferences from the original dataset.)

(Show DAA failure example, see below)

	Preference 1	Preference 2	Preference 3
cand1	emp2	emp1	emp3
cand2	emp3	emp1	-----
cand3	emp1	-----	-----

	Preference 1	Preference 2	Preference 3
emp1	cand1	cand2	cand3
emp2	cand1	-----	-----
emp3	cand2	cand1	-----

Gale–Shapley Matching Algorithm

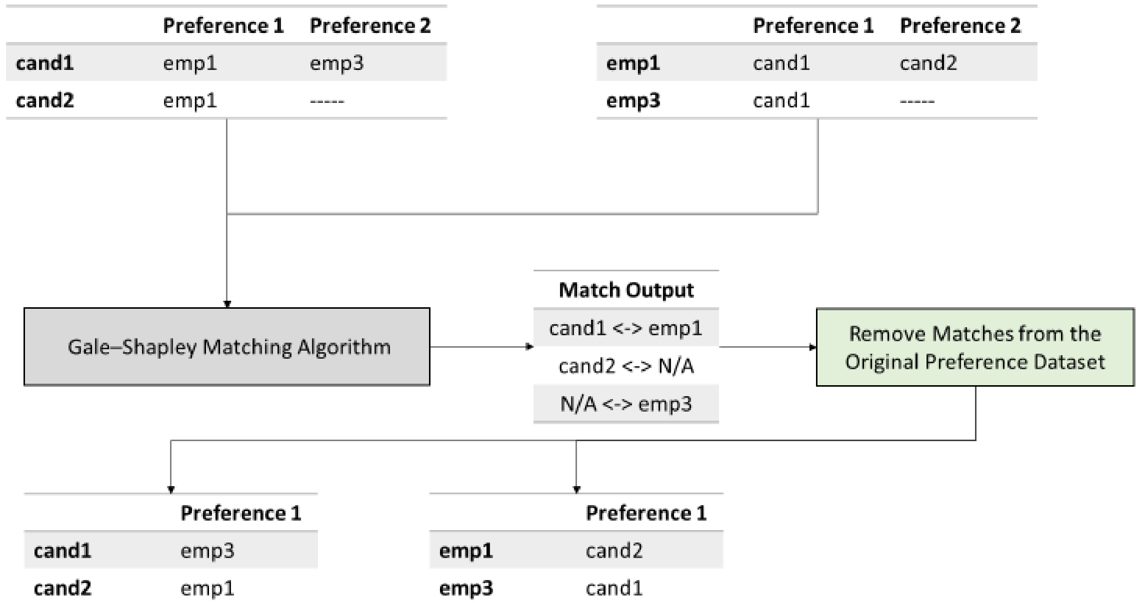
Match Output
cand1 <-> emp2
cand2 <-> emp3
cand3 <-> emp1

Remove Matches from the Original Preference Dataset

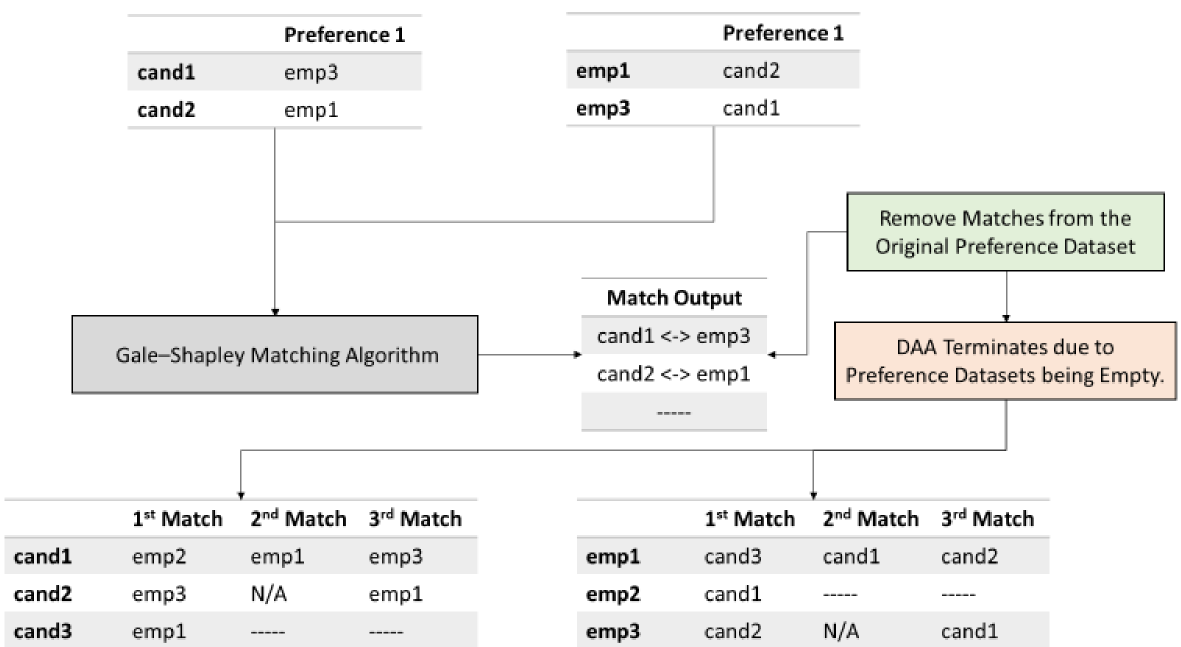
	Preference 1	Preference 2
cand1	emp1	emp3
cand2	emp1	-----
cand3	-----	-----

	Preference 1	Preference 2
emp1	cand1	cand2
emp2	-----	-----
emp3	cand1	-----

[Figure 8: DAA Failure Example, Round 1/3]



[Figure 9: DAA Failure Example, Round 2/3]



[Figure 10: DAA Failure Example, Round 3/3. Also shows all matches]

(Be sure to note that on round 2, it was impossible to find a stable one-to-one match for everyone in both parties. This was because after removing the matches in round 1, our preference datasets didn't meet the requirements of the Gale-Shapely Matching Algorithm. Both datasets must be completely filled, and have square dimensions, NxN.)

(Elaborate why having someone not get matched with anyone/sitting out/vacancy is a bad thing. Sitting out of rounds means you'll get that preference as a match in a later round, but what's the point in having it in a later match when someone else will get it before you?)

4.4 Non-Negative Matrix Factorization

(Talk about how we can help deal with the sparsity in the preference datasets, LMF!)

(Show example of performing LMF on a sparse preference dataset, see below)

	Preference of emp1	Preference of emp2	Preference of emp3	Preference of emp4
cand1	2	3	1	-----
cand2	2	-----	-----	1
cand3	1	2	-----	3
cand4	1	-----	3	2
cand5	-----	1	2	-----

Non-Negative Matrix Factorization (LMF)

	Preference of emp1	Preference of emp2	Preference of emp3	Preference of emp4
cand1	1.72	2.30	1.62	2.64
cand2	1.10	1.19	1.23	1.19
cand3	1.43	2.12	1.34	2.25
cand4	1.44	2.14	2.07	2.00
cand5	0.75	1.27	1.44	1.16

Re-Scale the Preferences to Integers

	Preference of emp1	Preference of emp2	Preference of emp3	Preference of emp4
cand1	2	3	1	4
cand2	1	3	4	2
cand3	2	3	1	4
cand4	1	4	3	2
cand5	1	3	4	2

[Figure 11: LMF example]

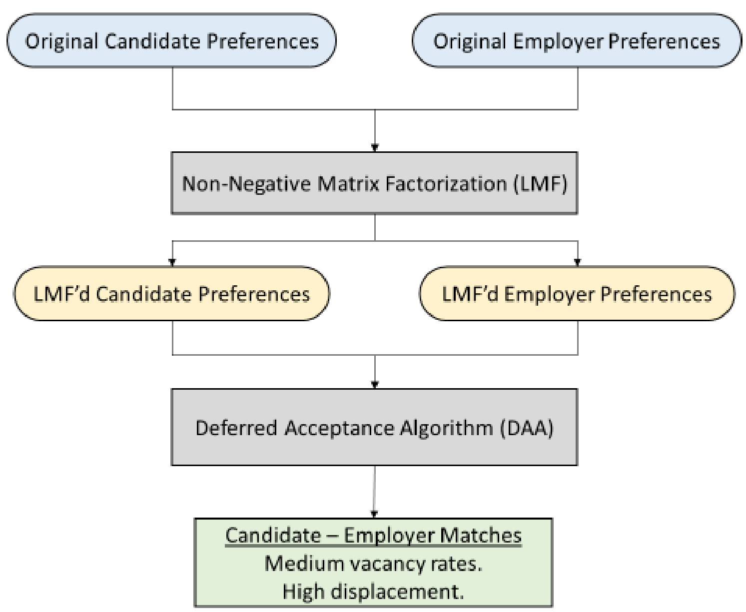
	Preference of emp1	Preference of emp2	Preference of emp3	Preference of emp4
cand1	2	3	1	4
cand2	1	3	4	2
cand3	2	3	1	4
cand4	1	4	3	2
cand5	1	3	4	2

Convert Preferences into a Descending List of Preferences. (Sparse to Dense Format)

	Preference 1	Preference 2	Preference 3	Preference 4
cand1	emp3	emp1	emp2	emp4
cand2	emp1	emp4	emp2	emp3
cand3	emp3	emp1	emp2	emp4
cand4	emp1	emp4	emp3	emp2
cand5	emp1	emp4	emp2	emp3

[Figure 12: Converting the preference dataset from sparse format to dense format]

(Show new DAA architecture, see below)



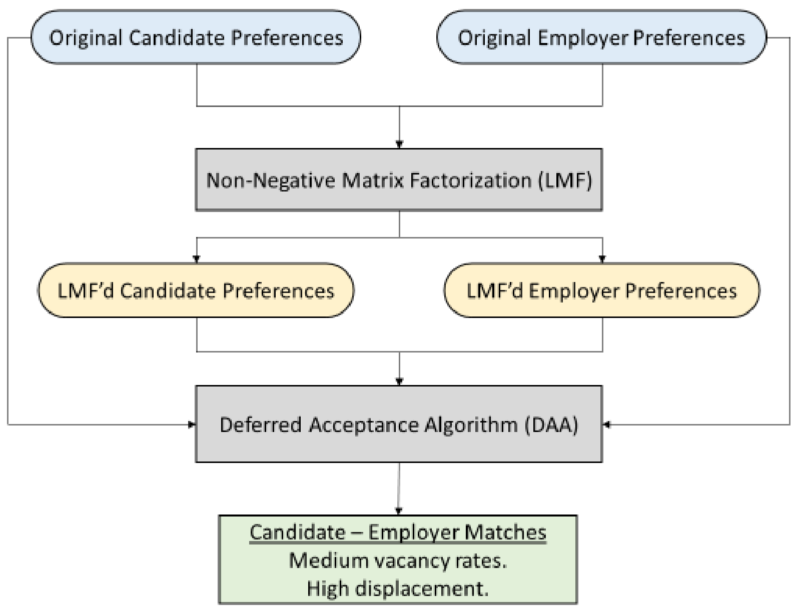
[Figure 13: General Input/Output of DAA with LMF Architecture]

(Mention that LMF only helps *reduce* sparsity, but it doesn't completely get rid of it. This is because we are still removing preferences as we progress through each round of the DAA.)

(Also mention that the accuracy/displacement of using these new LMF'd preference datasets in the DAA is lower than using the original preference datasets.)

4.5 Mixed DAA

(Talk about how we can use both, the original preference datasets, and the LMF'd preference datasets, to fully minimize vacancy as well as achieve great accuracy/displacement.)



[Figure 14: General Input/Output of Mixed DAA Architecture]

(Talk about how the Mixed DAA works. We use the original preference datasets, but when someone has to sit out, we use the LMF'd datasets to find them a match.)

TODO:

Maybe create an example of the Mixed DAA? I feel like I already have a lot of examples for a technical section though.

Definitely need some more work on the Mixed DAA section, any suggestions?

I also have some data regarding each algorithm's vacancy and displacement per round. Should I include some of that?