Egalitarian Stable Matching with Ties

What is **Matching Problem**?

Given two sets R = {r1, …, rn}, H = {h1, …, hn}
Each agent in one set rank the agents in another set.

Goal: Match each agent to **exactly one** agent in the other set.

What is **Stable** Matching Problem?

Given two sets A = {a1, …, an}, B = {b1, …, bn}
Each agent in one set rank the agents in another set.

Goal: Match each agent to **exactly one** agent in the other set, respecting their preferences.

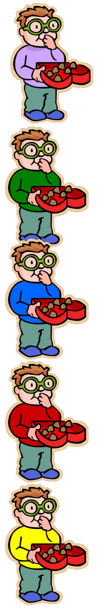
How do we “respect preferences”?

By avoiding block pairs.

Block Pair: Unmatched pair (r, h) where r prefers h to its current match, and h prefers r to its current match.

Example (Stable Marriage):

Preferences:



Boys	Girls
1: CBEAD	A : 35214
2 : ABECD	B : 52143
3 : DCBAE	C : 43512
4 : ACDBE	D : 12345
5 : ABDEC	E : 23415

What is Stable Matching Problem **with Ties**? (SMT)

The difference about it is that the preference lists include ties, in the sense that a person finds two of the persons of the opposite sex equally preferable and they both occupy the same position in his/her preference list.

What is Stable Matching Problem **with Incomplete List**? (SMI)

In this case, we have the situation where a woman might declare that one or more men are unacceptable for her, meaning she would under no circumstances accept a proposal from them even if she were single.

Example:

Standard problem

w_1: m_1 m_3 m_4 m_2

w_2: m_4 w_1 m_3 m_2

w_3: m_1 m_2 m_3 m_4

w_4: m_2 m_3 m_4 m_1

SMT

w_1: m_1 [m_3 m_4] m_2

w_2: m_4 m_1 [m_3 m_2]

w_3: m_1 m_2 m_3 m_4

w_4: [m_2 m_3 m_4] m_1

SMI

w_1: m_1

w_2: m_4 m_3 m_2

w_3: m_1 m_2

w_4: m_2 m_3 m_4 m_1

SMTI

w_1: m_1 [m_3 m_4]

w_2: [m_3 m_2]

w_3: m_1

w_4: [m_2 m_3 m_4] m_1

What is **Egalitarian?**

If M is a stable matching between n men and women, the position of a woman w in a man’s preference list is $P_m(W)$,and for woman $P_W(M)$ respectively.

The egalitarian happiness is defined as:

Egalitarian cost

$$eh(M) = \sum P_m(w) - \sum P_w(m)$$

Gale and Shapley Algorithm(GSS):

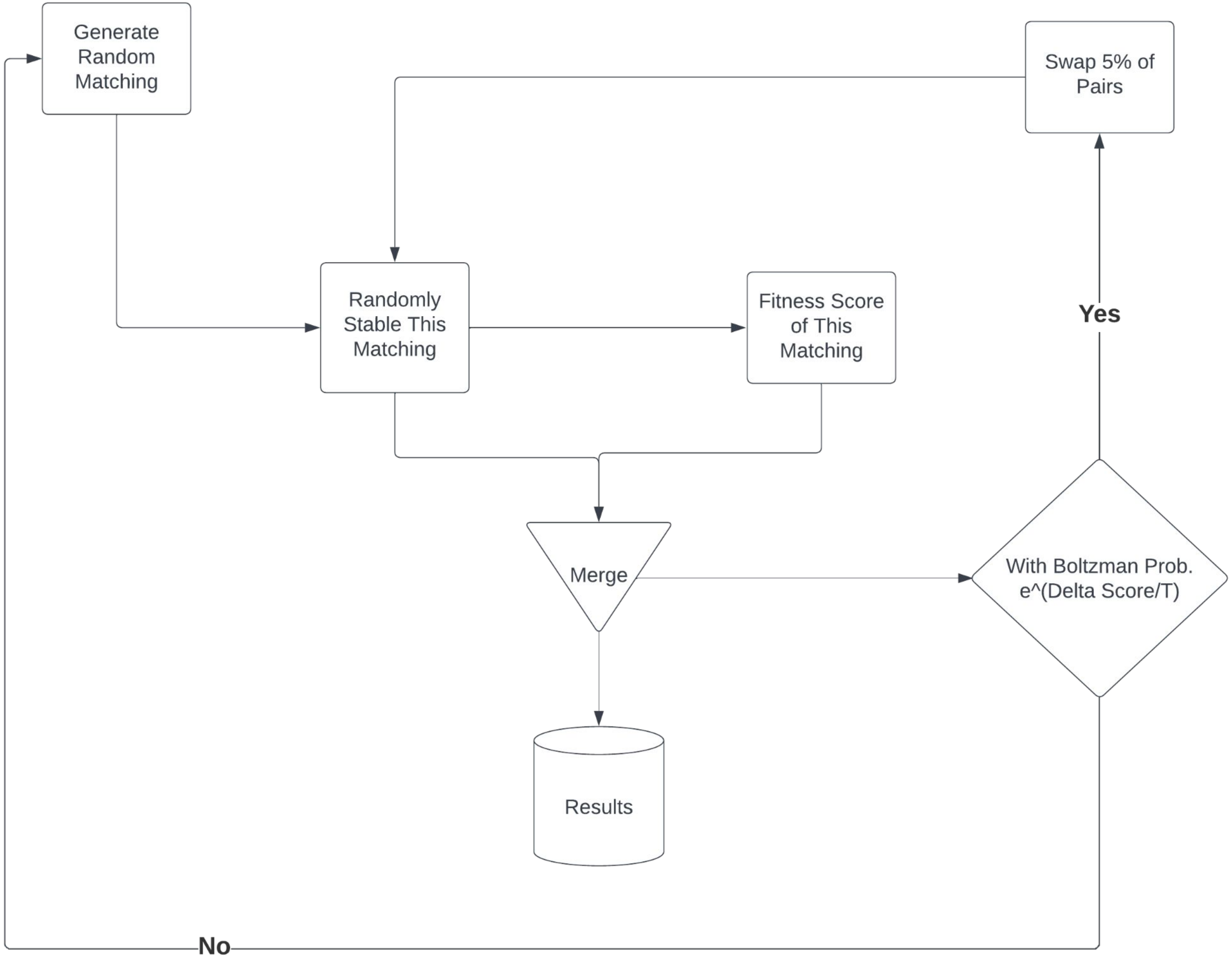
- Gale and Shapley Algorithm gives a matching that is stable for Stable Marriage and Stable Marriage with Ties problem.
- We compare our results with this Algorithm.

Gale and Shapley Algorithm(GSS):

Initially all r in R and h in H are free
While there is a free r
    Let h be highest on r's list that r has not proposed to
    if h is free, then match (r, h)
    else //h is not free
    suppose (r', h) are matched
        if h prefers r to r'
            unmatch (r', h)
            match (r, h)

Our Work

Hill Climbing Algorithm:



GENETIC ALGORITHM:

Fitness Score:

F = Stable pairs - Egalitarian Happiness.

Time Complexity:

O(n^2) <- For finding Randomly Stable Matching

Total : O(S * n^3) : S times we run the algorithm.

Genetic Algorithm:

Crossover :

Crossover is a reproduction technique that requires two chromosomes, the “parents”. By combining the two parents in a specific way, a number of new chromosomes is acquired, called “children”. That can be achieved in a variety of ways. The most common is one or two point crossover. A position is selected in both chromosomes and then they are split them into two pieces and the pieces in the two parents are swapped.

Parent 1: [3, 2, 5, 1, 4]
Parent 2: [1, 3, 4, 2, 5]
starting point is position **2**

Step	Parents	Children
1	**Parent 1**: [3, **2**, 5, 1, 4] **Parent 2**: [1, **3**, 4, 2, 5]	**Child 1** : [, 2, , ,] **Child 2** : []
2	**Parent 1**: [**3**, 2, 5, 1, 4] **Parent 2:** [**1**, 3, 4, 2, 5]	**Child 1** : [3, 2, , ,] **Child 2** : []
3	**Parent 1**: [3, 2, 5, **1**, 4] **Parent 2**: [1, 3, 4, **2**, 5]	**Child 1 :** [3, 2, , 1,] **Child 2** : []
4	**Parent 1**: [3, 2, **5**, 1, **4**] **Parent 2**: [1, 3, 4, 2, 5]	**Child 1 :** [3, 2, , 1,] **Child 2** : [, , 5, , 4]

Final children

Child 1 : [3, 2, 4, 1, 5]
Child 2 : [1, 3, 5, 2, 4]

Genetic Algorithm:

MUTATION:

Here we randomly uses two Mutation Techniques:

1. Mutation is a genetic operator used to maintain genetic diversity from one generation of a population of genetic algorithm chromosomes to the next. There are various mutation methods; inversion mutation is used here. Inversion mutation works as follows:

Step 1: select two random points

Step 2: swap them.

We do this on 5% population.

2. In the second Technique we locate all the unstable pairs in an individual and then select one of them randomly and swap it with another random element.

Genetic Algorithm:

SELECTION:

Selection is a method that randomly picks chromosomes out of the population according to their fitness function. The higher the fitness function, the more chance an individual has to be selected.

Roulette wheel selection is used:

In Roulette wheel selection method, each individual is assigned a slice of a circular “roulette wheel”, the size of the slice being proportional to the individual fitness. The wheel is spun N time, where N is the number of individuals in the population. On each spin, the individual under the wheel marker is selected to be in the pool of parents for the next generation.

Fitness Score:

F = Stable pairs - Egalitarian Happiness.

Time Complexity:

O(Pf*n^2) <- Cross Over

O(Pf*n) <- Selection Process

Total : O(S * (Pf * n^2 + Pf*n)) : S times we run the algorithm.

Results:

Gale and Shapley Algorithm

N	Stable Pairs	Egalitarian Happiness
30	30	7.2667
50	50	10.1
100	40	23.81

Hill Climbing

N	Stable Pairs	Egalitarian Happiness
30	23.51	1.0531
50	33.12	4.05
100	61.21	6.51

Genetic Algorithm

N	Stable Pairs	Egalitarian Happiness
30	29.51	0.145
50	47.211	2.1
100	73.011	3.11