EE 382V Social Computing

Incremental Stable Roommates Algorithm for Clustered Input Attributes

Presented By -
Bhargavi Karthik
Carmina Francia
Haleigh Walker
Nikita Trivedi

What is a “Stable Roommates” problem?

- Variation on the stable marriage problem by Gale and Shapley.

- In 1984, Robert Irving published an algorithm ‘*An efficient algorithm for the “stable roommates” problem*’

- Might not always have a stable matching possible.

1. **Two Phases of Irving's Algorithm explained using an example of 6 participants.**

2. **Apply the incremental approach on the same example to come up with new stable matches.**

1st Phase of the Irving's algorithm

Make Initial Proposals

Rule out the worst matches

Preference list of 6 people hoping to get their preferred roommates:

Person	Choice 1	Choice 2	Choice 3	Choice 4	Choice 5
Beth	Joe	Peter	Mike	Tim	Edwards
Joe	Mike	Edwards	Beth	Peter	Tim
Peter	Mike	Tim	Joe	Beth	Edwards
Edwards	Beth	Peter	Tim	Joe	Mike
Mike	Beth	Joe	Edwards	Tim	Peter
Tim	Joe	Beth	Edwards	Peter	Mike

Step 1: Making the Proposals:

Beth → Joe

Joe → Mike

Peter → Mike Tim

Edwards → Beth Peter

Mike → Beth

Tim → Beth Joe wards

Person	Choice 1	Choice 2	Choice 3	Choice 4	Choice 5
Beth	Joe	Peter	Mike	Tim	Edwards
Joe	Mike	Edwards	Beth	Peter	Tim
Peter	Mike	Tim	Joe	Beth	Edwards
Edwards	Beth	Peter	Tim	Joe	Mike
Mike	Beth	Joe	Edwards	Tim	Peter
Tim	Joe	Beth	Edwards	Peter	Mike

Step 2: Remove the worst matches

Beth → Joe

Joe → Mike

Peter → Tim

Edwards → Peter

Mike → Beth

Tim → Edwards

Person	Choice 1	Choice 2	Choice 3	Choice 4	Choice 5
Beth	Joe	Peter	Mike	Tim	Edwards
Joe	Mike	Edwards	Beth	Peter	Tim
Peter	Mike	Tim	Joe	Beth	Edwards
Edwards	Beth	Peter	Tim	Joe	Mike
Mike	Beth	Joe	Edwards	Tim	Peter
Tim	Joe	Beth	Edwards	Peter	Mike

Step 2: Removing Beth's worst partners

Beth → Joe

Joe → Mike

Peter → Tim

Edwards → Peter

Mike → Beth

Tim → Edwards

Person	Choice 1	Choice 2	Choice 3	Choice 4	Choice 5
Beth	Joe	Peter	Mike	Tim	Edwards
Joe	Mike	Edwards	Beth	Peter	Tim
Peter	Mike	Tim	Joe	Beth	Edwards
Edwards	Beth	Peter	Tim	Joe	Mike
Mike	Beth	Joe	Edwards	Tim	Peter
Tim	Joe	Beth	Edwards	Peter	Mike

Step 2: Removing Joe's worst partners

Beth → Joe

Joe → Mike

Peter → Tim

Edwards → Peter

Mike → Beth

Tim → Edwards

Person	Choice 1	Choice 2	Choice 3	Choice 4	Choice 5
Beth	Joe	Peter	Mike	Tim	Edwards
Joe	Mike	Edwards	Beth	Peter	Tim
Peter	Mike	Tim	Joe	Beth	Edwards
Edwards	Beth	Peter	Tim	Joe	Mike
Mike	Beth	Joe	Edwards	Tim	Peter
Tim	Joe	Beth	Edwards	Peter	Mike

Step 2: Continuing it for the rest of the people to get a Stable Table or a Reduced list:

Beth → Joe

Joe → Mike

Peter → Tim

Edwards → Peter

Mike → Beth

Tim → Edwards

Person	Choice 1	Choice 2	Choice 3	Choice 4	Choice 5
Beth	Joe	Peter	Mike	Tim	Edwards
Joe	Mike	Edwards	Beth	Peter	Tim
Peter	Mike	Tim	Joe	Beth	Edwards
Edwards	Beth	Peter	Tim	Joe	Mike
Mike	Beth	Joe	Edwards	Tim	Peter
Tim	Joe	Beth	Edwards	Peter	Mike

2nd Phase of the Irving's algorithm

Finalize the Best Matches

Step 3: Finalize the best matching by finding a rotation:

Person	Choice 1	Choice 2	Choice 3	Choice 4	Choice 5
Beth	Joe	Peter	Mike	Tim	Edwards
Joe	Mike	Edwards	Beth	Peter	Tim
Peter	Mike	Tim	Joe	Beth	Edwards
Edwards	Beth	Peter	Tim	Joe	Mike
Mike	Beth	Joe	Edwards	Tim	Peter
Tim	Joe	Beth	Edwards	Peter	Mike

p	Beth	Edwards	Peter	Mike	Beth
q	Peter	Tim	Beth	Joe	

Step 3: Finalize the best matching

Person	Choice 1	Choice 2	Choice 3	Choice 4	Choice 5
Beth	Joe	Peter	Mike	Tim	Edwards
Joe	Mike	Edwards	Beth	Peter	Tim
Peter	Mike	Tim	Joe	Beth	Edwards
Edwards	Beth	Peter	Tim	Joe	Mike
Mike	Beth	Joe	Edwards	Tim	Peter
Tim	Joe	Beth	Edwards	Peter	Mike

3 stable matches at the end of the first run of the algorithm:

Beth and Peter

Joe and Mike

Tim and Edwards

Incremental setting applied to the current scenario:

- New people enter into or leave the current setting after the stable matches have been determined.
- Assumption: People enter and leave only in the multiples of two.
- New preference lists are created whenever the size of the set changes.
- Our approach determines a stable matching only for the unhappy people in the set, while maintaining the current stable matching for the happy couples.

Incremental Setting Algorithm

```
if even number of new students join then
    for each student s that is joining do
        AddToUnhappyPool(s)
```

```
if even number of students leave then
    if not all leaving students are matched within themselves then
        for every student s who lost their partners do
            AddToUnhappyPool(s)
```

```
function ()AddToUnhappyPool(Students)
    FOR EACH PAIR (S1, S2) IN THE MATCHING DO
        IF BOTH ARE HAPPY WITH THEIR CURRENT ROOMMATE COMPARED TO S THEN
            KEEP THE MATCHING (S1, S2) INTACT
        ELSE IF AT LEAST ONE OF THEM IS NOT HAPPY WITH THEIR CURRENT ROOMMATE COMPARED TO S THEN
            ADDTOUNHAPPYPOOL(S1)
            ADDTOUNHAPPYPOOL(S2)
            REMOVE THE PAIR (S1, S2) FROM MATCHING
CONSTRUCT A NEW PREFERENCE LIST BY EXCLUDING THE MEMBERS IN MATCHING
```

Incremental Setting Algorithm

```
if even number of new students join then
    for each student s that is joining do
        AddToUnhappyPool(s)
```

```
if even number of students leave then
    if not all leaving students are matched within themselves then
        for every student s who lost their partners do
            AddToUnhappyPool(s)
```

```
function ()AddToUnhappyPool(Students)
    FOR EACH PAIR (S1, S2) IN THE MATCHING DO
        IF BOTH ARE HAPPY WITH THEIR CURRENT ROOMMATE COMPARED TO S THEN
            KEEP THE MATCHING (S1, S2) INTACT
        ELSE IF AT LEAST ONE OF THEM IS NOT HAPPY WITH THEIR CURRENT ROOMMATE COMPARED TO S THEN
            ADDTOUNHAPPYPOOL(S1)
            ADDTOUNHAPPYPOOL(S2)
            REMOVE THE PAIR (S1, S2) FROM MATCHING
CONSTRUCT A NEW PREFERENCE LIST BY EXCLUDING THE MEMBERS IN MATCHING
```

$O(n^2)$ -> O(nk)

n - total students

k - cluster size, k < n

Person	Choice 1	Choice 2	Choice 3	Choice 4	Choice 5	Choice 6	Choice 7
Beth	Joe	Peter	Mike	Alex	Stan	Tim	Edwards
Joe	Mike	Edwards	Beth	Peter	Tim	Alex	Stan
Peter	Beth	Stan	Joe	Alex	Edwards	Tim	Mike
Edwards	Beth	Peter	Tim	Stan	Joe	Alex	Mike
Mike	Beth	Joe	Edwards	Tim	Alex	Stan	Peter
Tim	Stan	Beth	Edwards	Peter	Joe	Mike	Alex
Stan	Peter	Joe	Edwards	Beth	Alex	Mike	Tim
Alex	Tim	Stan	Beth	Mike	Peter	Edwards	Joe

Person	Choice 1	Choice 2	Choice 3	Choice 4	Choice 5	Choice 6	Choice 7
Beth	Joe	Peter	Mike	Alex	Stan	Tim	Edwards
Joe	Mike	Edwards	Beth	Peter	Tim	Alex	Stan
Peter	Beth	Stan	Joe	Alex	Edwards	Tim	Mike
Edwards	Beth	Peter	Tim	Stan	Joe	Alex	Mike
Mike	Beth	Joe	Edwards	Tim	Alex	Stan	Peter
Tim	Stan	Beth	Edwards	Peter	Joe	Mike	Alex
Stan	Peter	Joe	Edwards	Beth	Alex	Mike	Tim
Alex	Tim	Stan	Beth	Mike	Peter	Edwards	Joe

- Stan
- Alex
- Tim
- Edwards

- Mike <-> Joe
- Beth <-> Peter

- Run Irving's algorithm with the Unhappy Pool $O(k^2)$

Final Stable Matches:

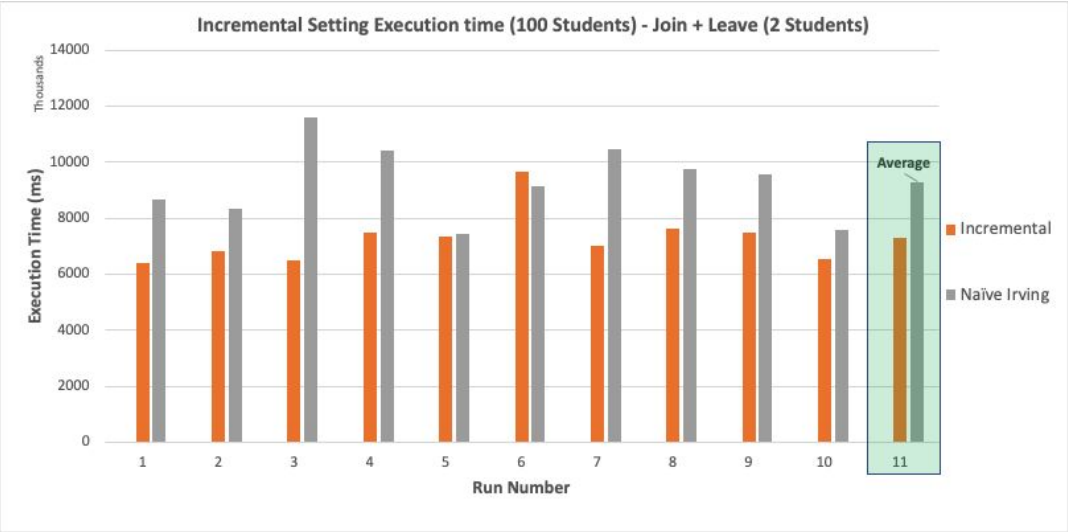
Beth and Peter

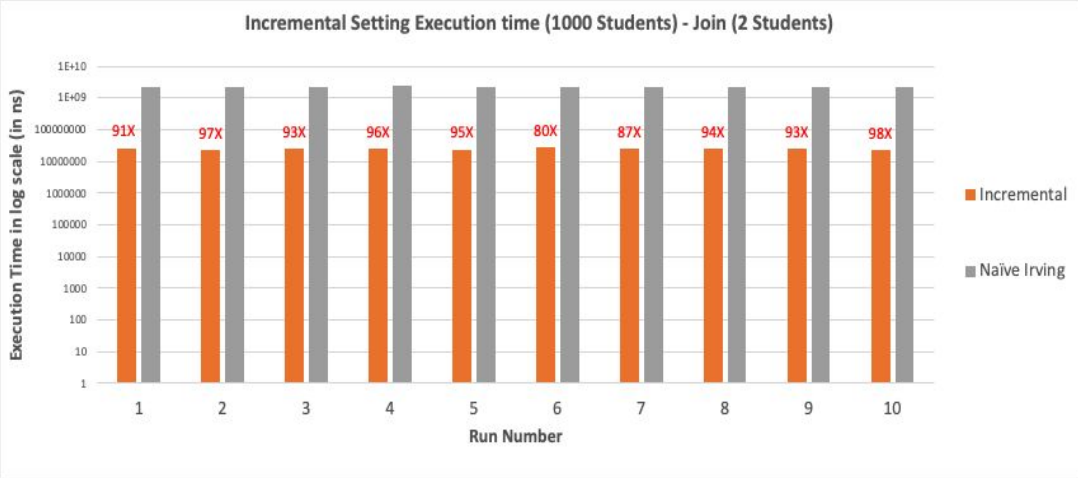
Tim and Edwards

Joe and Mike

Stan and Alex

Performance Results





- 100 X 100
- 5 clusters
- 2 new students join and leave
- 27% Speedup

- 1000 X 1000
- 2 skewed clusters
- 2 new students join
- 92X Speedup

- Caveat: What if everyone is unhappy?

DEMO

Conclusion

- Optimizes clustered attribute scenario
- Isolates matches affected by the increment and partitions into 2 sets
- Test cases contain clustered attributes
- 27 percent improvement in execution time (100 elements, five clusters)
- 92X speedup for best case scenarios (1000 elements, two skewed clusters)