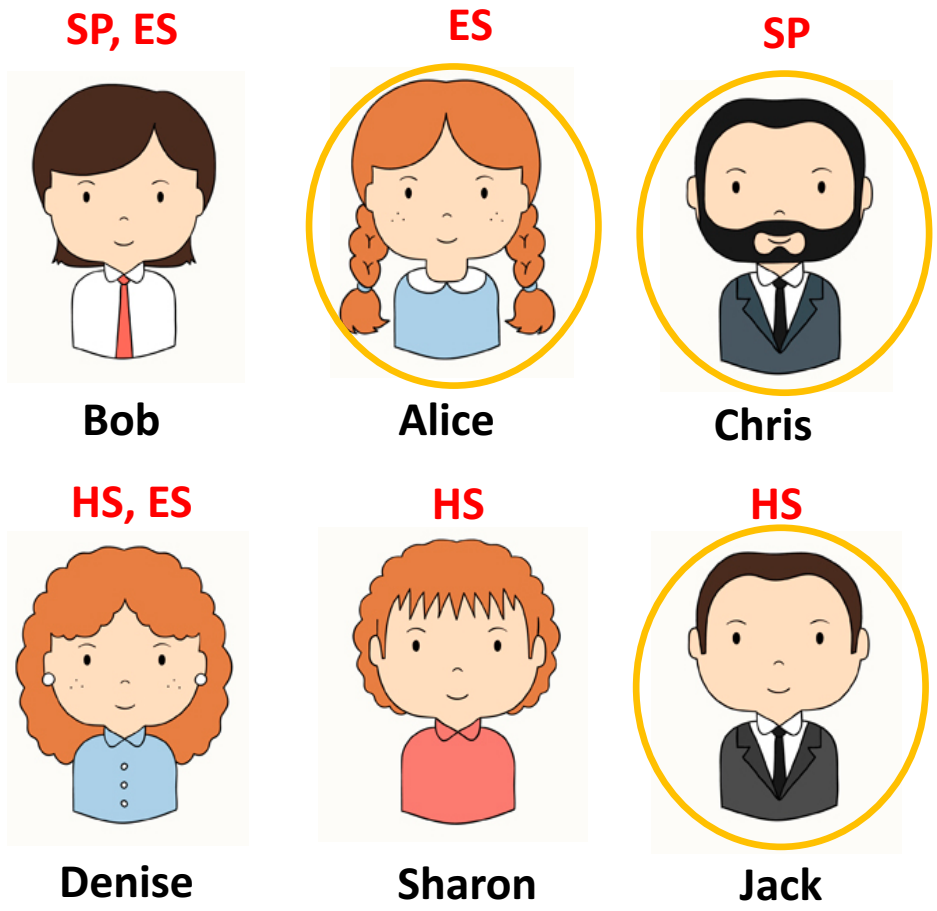
Crowdsourcing with Diverse Groups of Users

Sara Cohen Moran Yashinski

Team Formation problem

- Example: Forming an education board

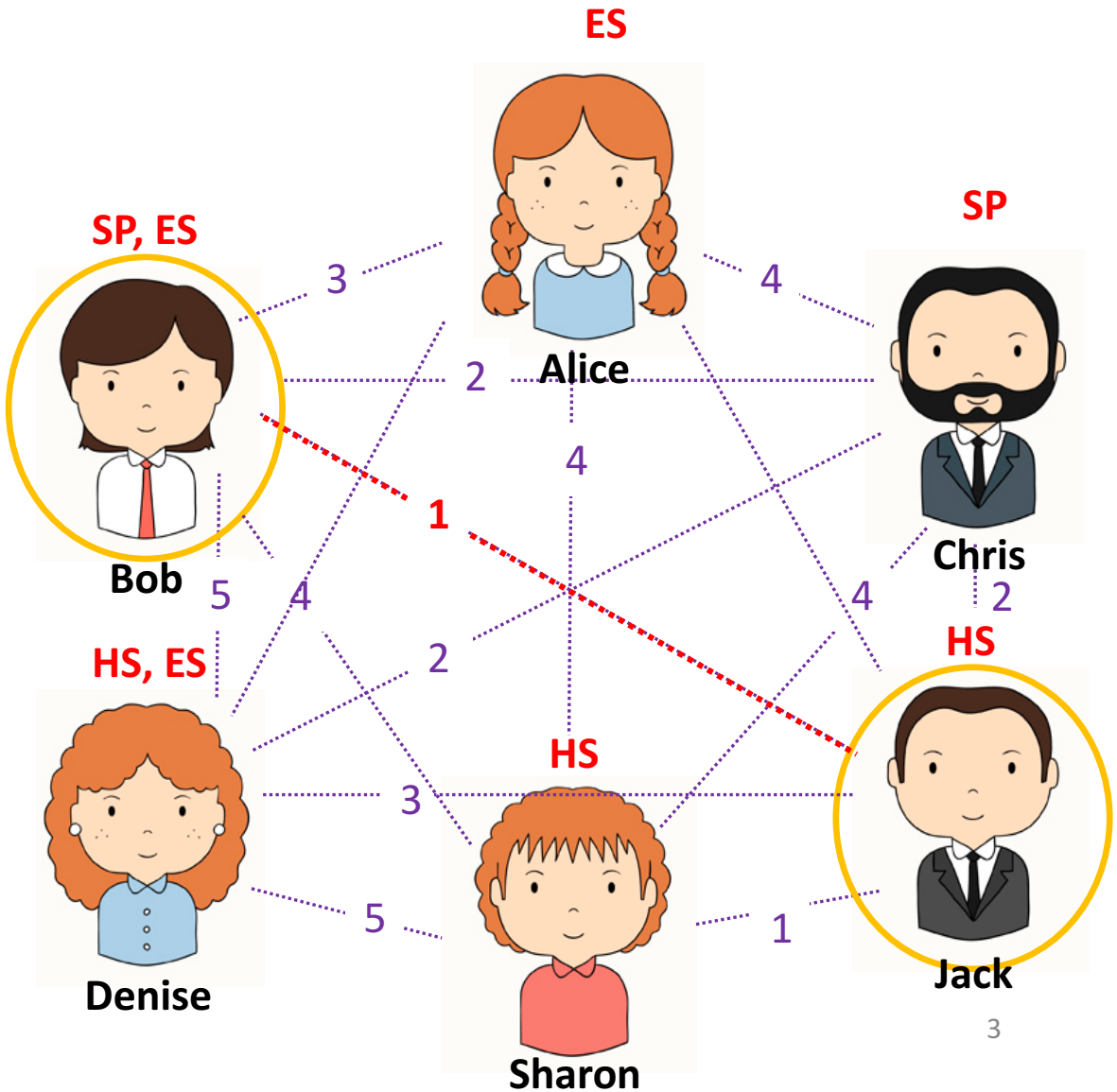
- Required skills:
  - School Principal (SP)
  - High School teacher (HS)
  - Elementary School teacher (ES)



Team Formation problem with Communication Cost

- Goal: Find a team that has all required skills, while minimizing communication cost

- Examples of communication costs
  - Distance in the social network
  - (An inverse of) the number of papers each 2 experts published together



Research Question

- What if we wanted to define diversity based on the properties?
  - Gender, Income, Age, Religion, Location, etc.

- We would like to define target diversity function for the different experts' properties

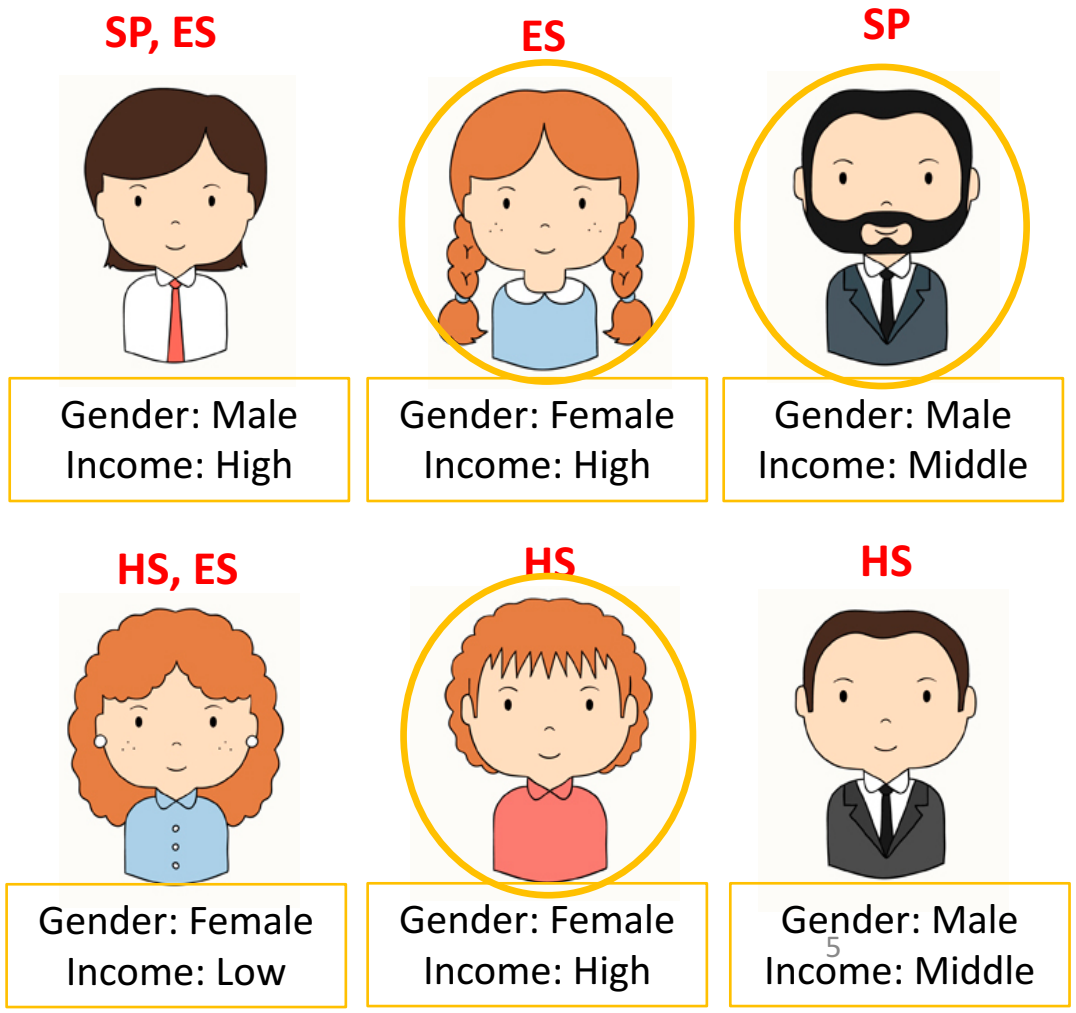
- Goal: Efficiently find a team that has all required skills, and is as close as possible to the desired target diversity

Team Formation with Target Diversity constraint

- Target Diversity based on **Properties**

- Goal: Efficiently find a team that has all required skills, and is as close as possible to the desired target diversity
- $\boldsymbol{Distribution\ Cost} = |Team\ Diversity\ - Target\ Disversity|_1$

- Example:
  - Gender Target Diversity: $[Male, Female] = [{}^1/_3, {}^2/_3]$
  - Income Target Diversity : $[High, Medium, Low] = [{}^1/_2, {}^1/_4, {}^1/_4]$



What are we going to discuss?

- Research Question: diversity based on personal properties ✓
- Advantages of Diversity (or.. why is it interesting?)
- Related work
- Algorithms and computational considerations
  - Fixed Parameters Tractable (Optimal) Algorithm
  - Greedy Approximation Algorithm
- Experimental Results
- Conclusions

Advantages of Diversity (or.. why is it interesting?)

- Advantages in the workplace
  - Increase in productivity and creativity (innovative solutions)
  - Increase morale in workplaces
  - Positive reputation/attraction of quality human resources
- When crowdsourcing, it is important to consider different points of views
- Defining the diversity of a team
  - Program committees
  - Adopting affirmative actions

Related Work

- Team formation with Communication Cost
  - Goal: Find a team that has all required skills, while minimizing communication cost (e.g. Sum of Distances, Diameter)
- Diversity in terms of social influence
  - Depends on the social influences between candidates
  - Low social influence is correlated with high productivity
- Diversity in query answering
  - The goal is to maximize the diversity of the results
  - Diversity based on different criteria (e.g. content, novelty and coverage)

What have we achieved?

- Finding an optimal solution is NP-complete
- Naïve algorithm
  - Check all possible options and finds optimal solution
  - Time complexity: $O(|C|^{|S|}|S||P|)$
  - Intractable in practice as $|C|$ might be huge
- Fixed Parameter Tractable (Optimal) Algorithm
  - Find an optimal solution in time complexity which is $poly(|C|)$ times $\exp(|S|, |P|)$
- Greedy Approximation Algorithm
  - Time complexity: $poly(|S|, |C|)$
  - Guaranteed to return 1/2-approximation of the optimal solution

Fixed Parameter Tractable (Optimal) Algorithm

- Finds optimal solution
- Complexity time: $poly(|C|)$ times $\exp(|S|, |P|)$
- Using preprocessed data structures in order to improve runtime performance
- Use the notion of Abstract (Optimal) Templates and Concrete Templates

Abstract (Optimal) Templates, Concrete Templates: Example

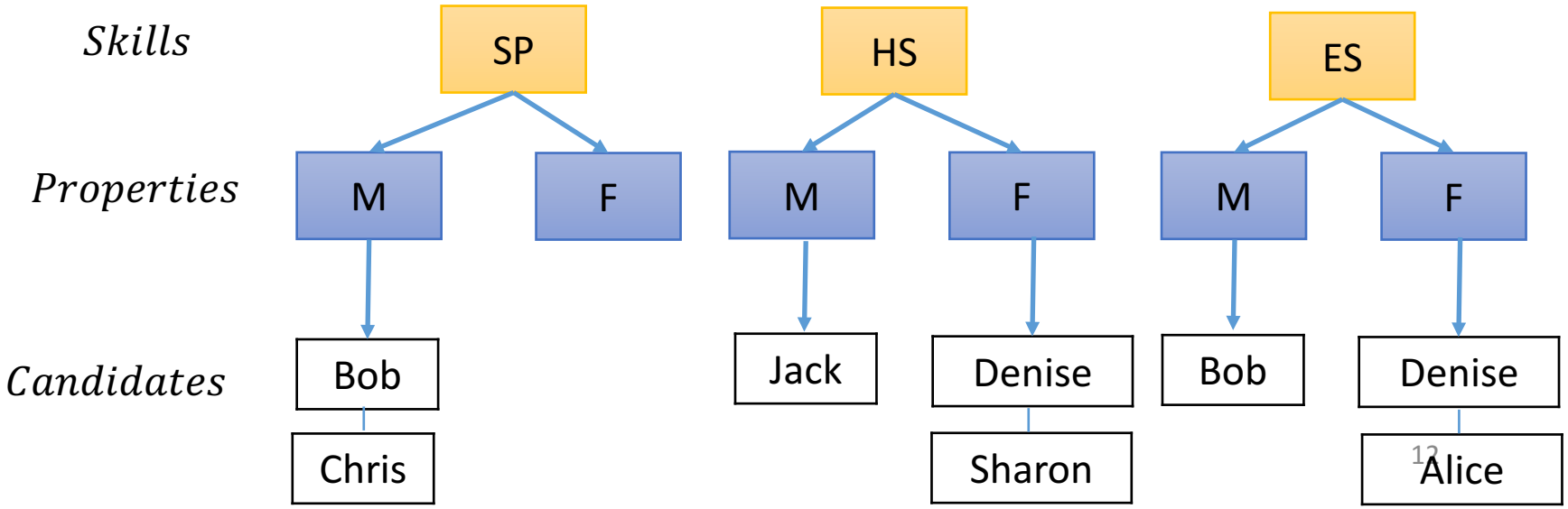
- One property (Gender):
  - $[Male, Female] = [{}^{2}/_{3}, {}^{1}/_{3}]$
- $S = \{SP, HS, ES\}$
- Abstract Optimal Template
  - Achieves **minimum distribution cost**
  - There could be many Abstract Optimal Templates
- Abstract Template (non optimal)
- Concrete Templates:
  - $gender(SP) = F, gender(HS) = M, gender(ES) = M$
  - $gender(SP) = M, gender(HS) = F, gender(ES) = M$
  - $gender(SP) = M, gender(HS) = M, gender(ES) = F$

Male	Female
2	1

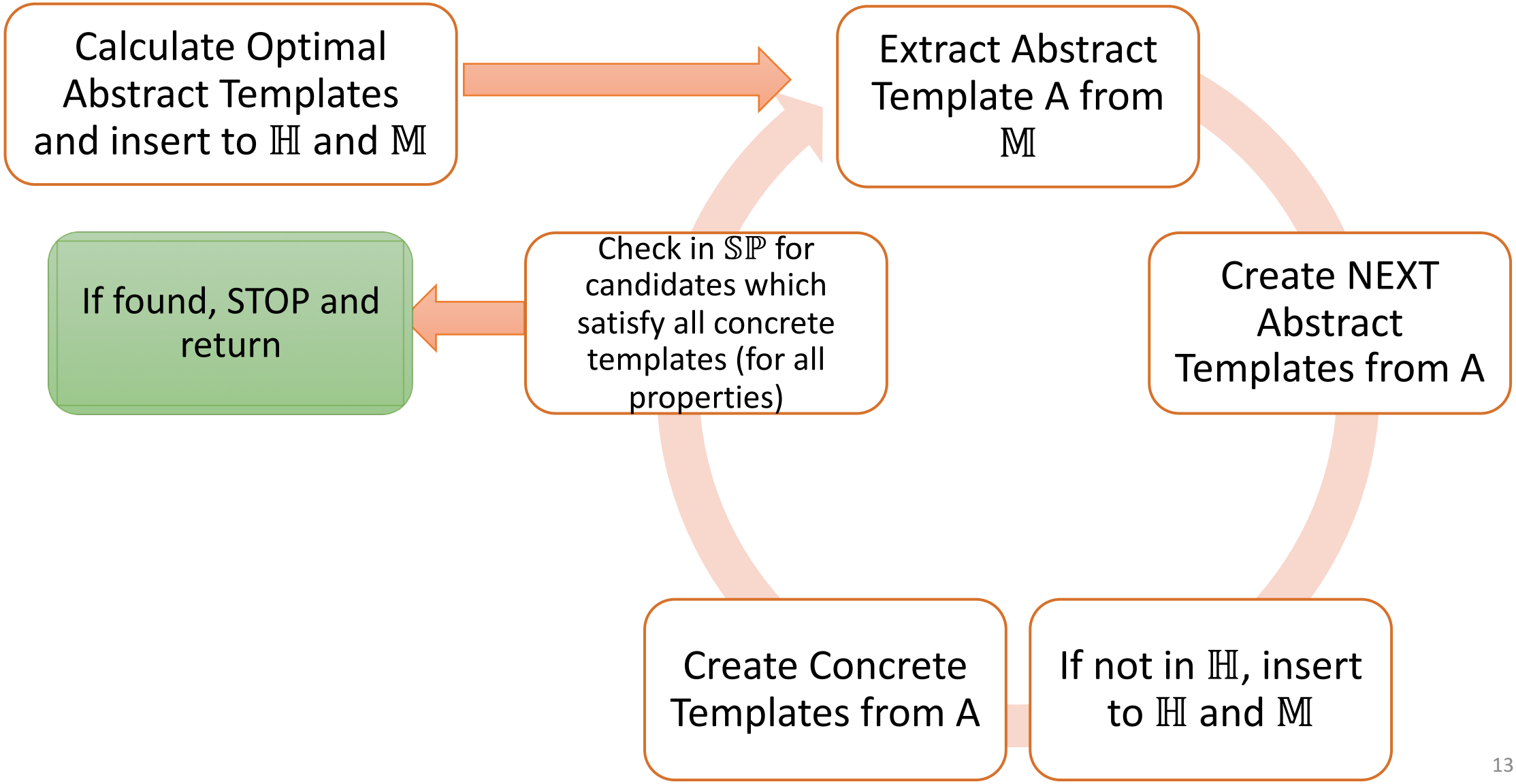
Male	Female
3	0

FPT Optimal Algorithm: Data structures

- Used to optimize runtime performance
- Hashset $\mathbb{H}$ to hold all the abstract templates
  - To avoid evaluating an abstract template more than once (very costly)
- minHeap $\mathbb{M}$ to efficiently return the abstract template which has minimum cost
- Structure $\mathbb{SPC}$
  - Calculated offline

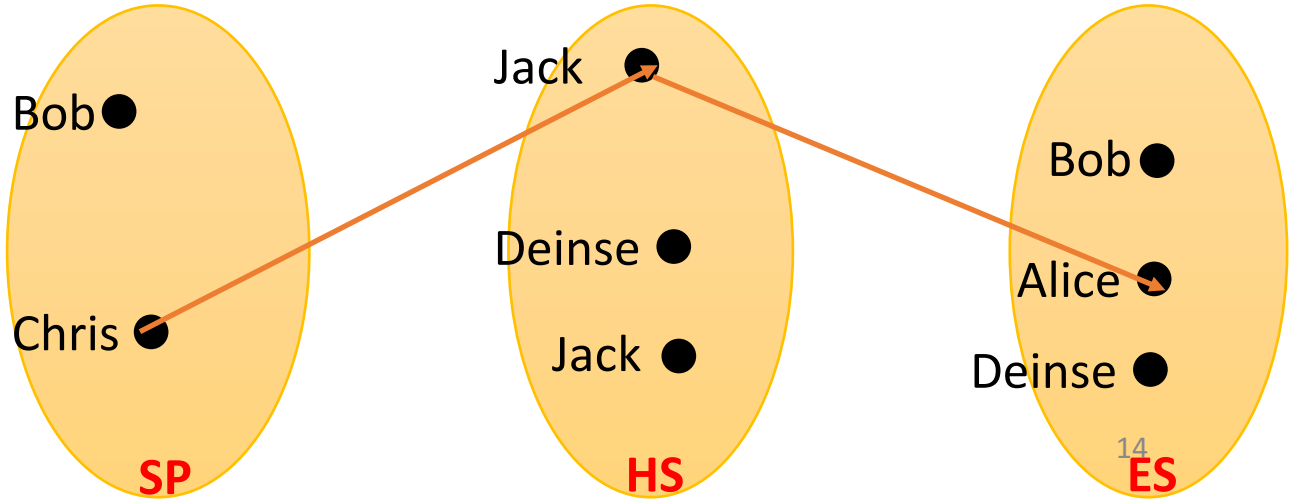


FPT Optimal Algorithm: Workflow



Greedy Approximation Algorithm

- Time complexity: $poly(|S|, |C|)$
- Using sets of candidates per skill
- Greedy solution: in each step chooses an unchosen skill and candidate with that skill which (locally) minimizes the distribution cost



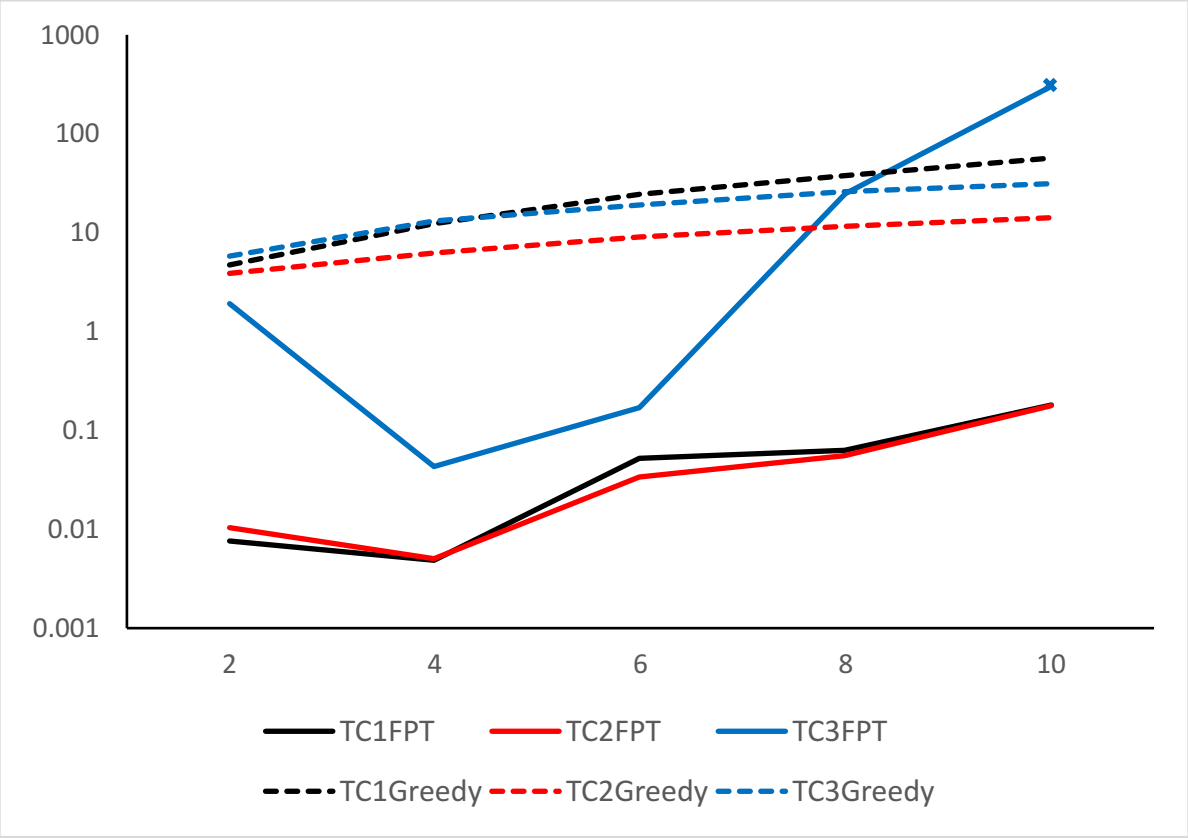
Greedy Approximation Algorithm (cont.)

- Optimizing a function call *benefit*, that is inversely proportional to the *distribution cost*
- The *benefit* function is a monotonic submodular function and therefore guaranteed to return 1/2-approximation of the optimal solution

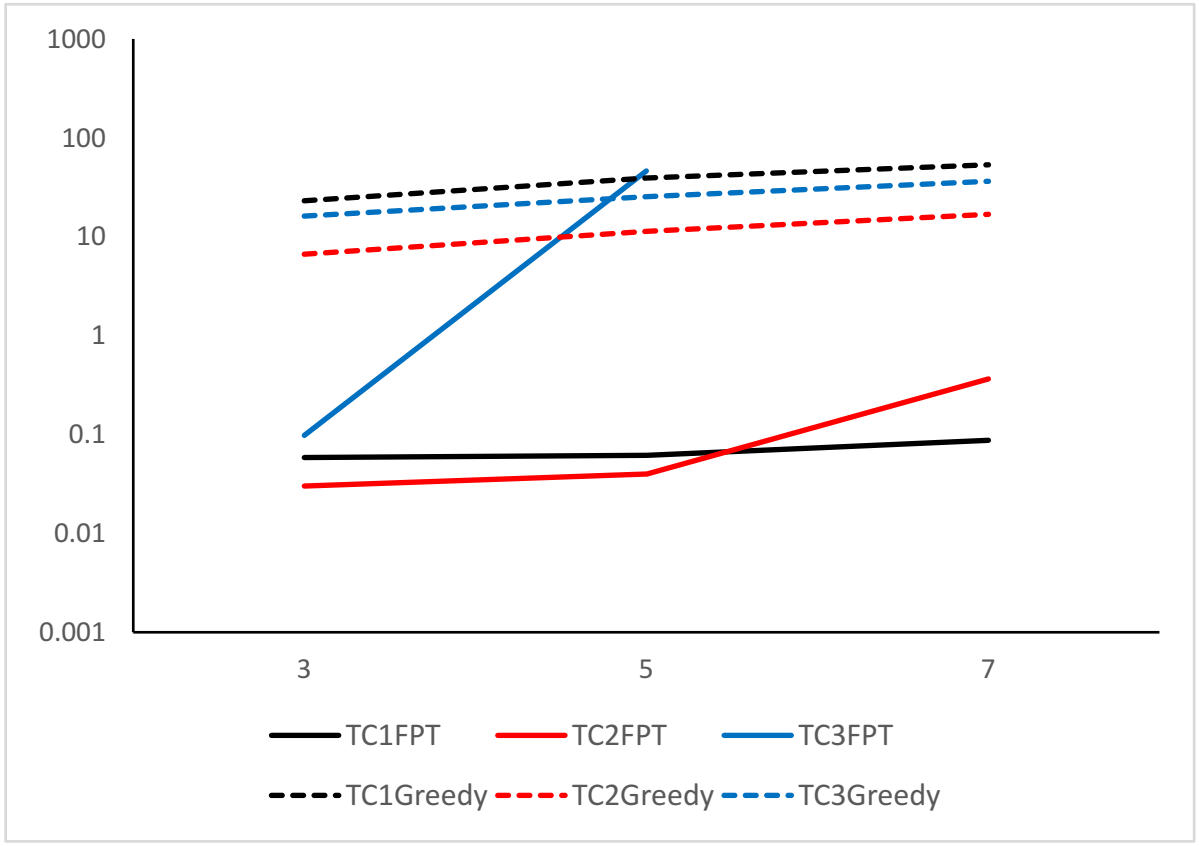
Experimentation

- Tested scalability as a function of $|C|, |S|, |P|$ and $Property\ Range$
  - Default values: $|S| = 8, |P| = 5, |C| = 100K, Property\ Range = 4$
- Types of synthetic datasets:
  - TC1 (random assignment)
    - Property values: assigned randomly using uniform distribution
    - Skills per candidate: randomly choosing between 1 and $|S|$ skills per candidate
  - TC2 (random assignment with 1 skill)
    - Property values: assigned randomly using uniform distribution
    - Skills per candidate: each candidate is given 1 random skill
  - TC3 (skewed distribution with 2 skills)
    - Property values and skills (2 skills per candidate) are assigned using a skewed distribution

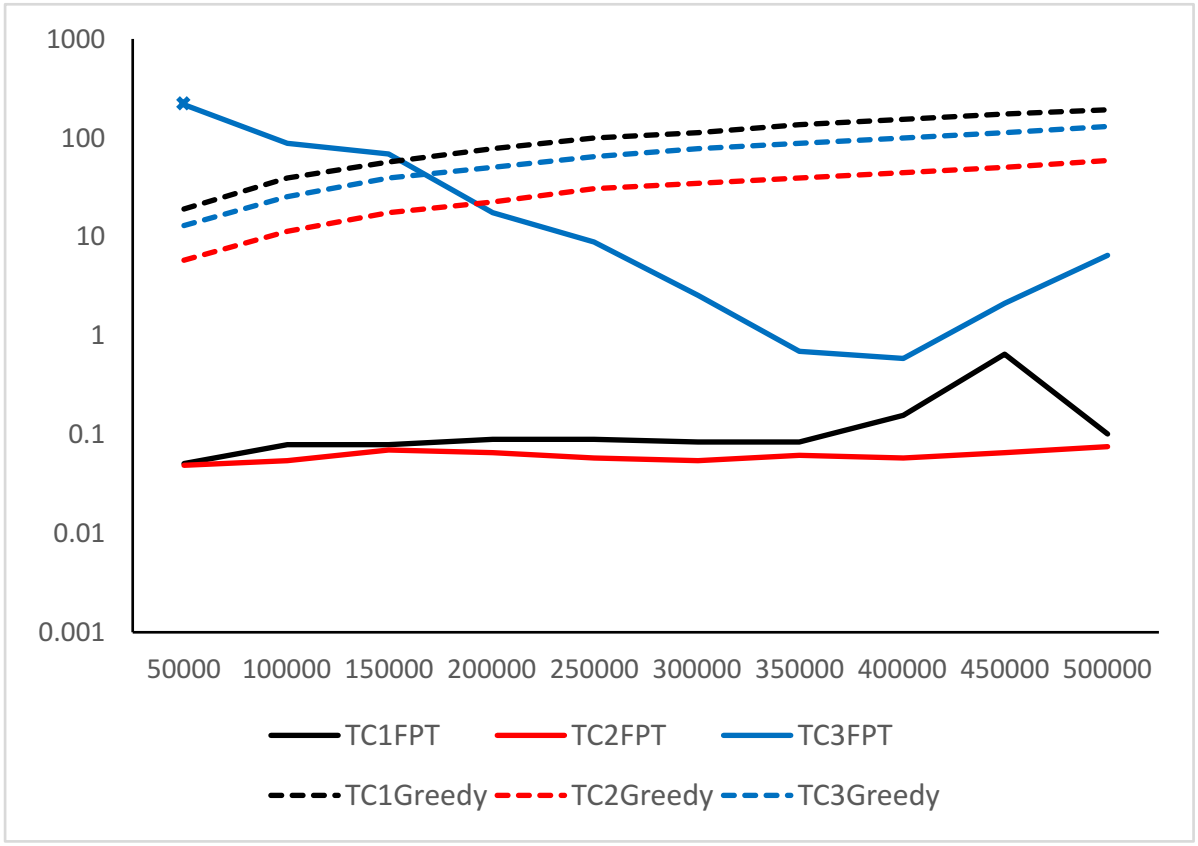
Experimentation: Varying number of skills



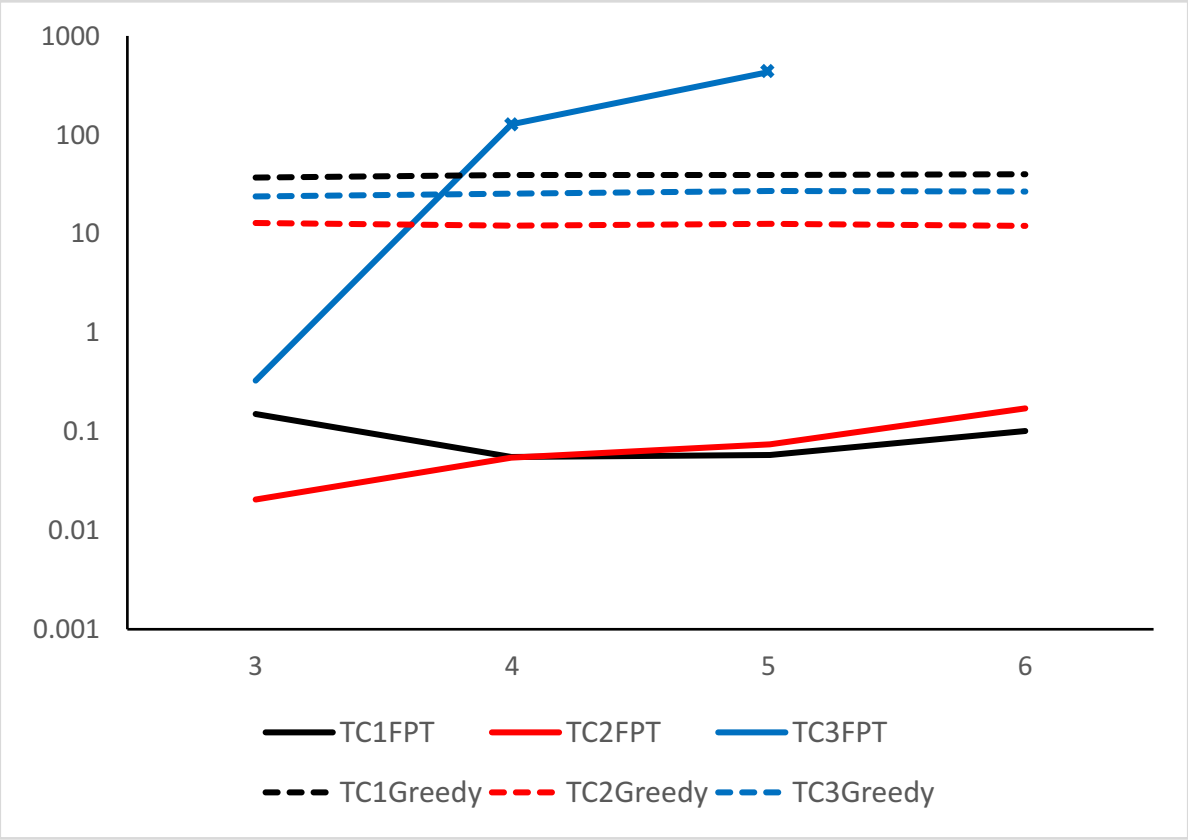
Experimentation: Varying number of properties



Experimentation: Varying number of candidates



Experimentation: Varying property range



Experimentation: Quality of Results (Greedy Vs. FPT)

	TC1	TC2	TC3
Max diff	0	0.25	0.5
Average over all test cases	0	0.01	0.11
Average over test cases in which greedy didn't return optimal result	0	0.25	0.29

Conclusions

- FPT Optimal Algorithm
  - Always returns an optimal result
  - Time increases exponentially with the number of skills, properties and property range
  - Increasing the number of candidates doesn’t impact running time (except when the data is skewed)
  - Might take long time to find the optimal solution (especially when the data is skewed)
  - Outperforms the Greedy Algorithm when there is little skew in the data

- Greedy Approximation Algorithm
  - Performs well under all types of data
  - Returns results close to optimal

Questions?