Worksheet 1: SMP

Rin Meng 51940633
Kevin Zhang 10811057
Mika Panagsagan 29679552

January 14, 2025

1 Build intuition through examples.

From this point onwards, we will read **Prefers**(x, y) as x prefers y over some other option.

1. Small and trivial instances of the problem, consider these instances:

   (a) Only one student s and one employer e:

   - Students: s_1
   - Employers: e_1
   - Preferences:
     A. **Prefers**(s_1, e_1)
     B. **Prefers**(e_1, s_1)
   - Triviality: Only one student and one employer, so there is only one match.

   (b) Equal preferences for all students and employers:

   i. Students: s_1, s_2
   ii. Employers: e_1, e_2
   iii. Preferences:
      A. **Prefers**$(s_1, e_1 \vee e_2)$
      B. **Prefers**$(s_2, e_1 \vee e_2)$
      C. **Prefers**$(e_1, s_1 \vee s_2)$

D. **Prefers**$(e_2, s_1 \vee s_2)$

iv. Triviality: All students and employers have indifferent preferences, so there are multiple stable matchings.

(c) Perfectly matched preferences between students and employers:

i. Students: s_1, s_2

ii. Employers: e_1, e_2

iii. Preferences:

A. **Prefers**(s_1, e_1)

B. **Prefers**(s_2, e_2)

C. **Prefers**(e_1, s_1)

D. **Prefers**(e_2, s_2)

iv. Triviality: All are perfectly matched, therefore there is only one stable matching.

2. Potential solutions to these instances:

   (a) Only one student s and one employer e:

   - The only stable matching is (s_1, e_1).
   - This solution is trivial but is optimal as well.

   (b) Equal preferences for all students and employers:

   - There are multiple stable matchings: $(s_1, e_1), (s_2, e_2)$ and $(s_1, e_2), (s_2, e_1)$.
   - This solution is optimal, but not unique.

   (c) Perfectly matched preferences between students and employers:

   - The only stable matching is $(s_1, e_1), (s_2, e_2)$.
   - This solution is optimal and unique.

   There are many ways to conclude if a solution is better. When we consider fairness and satisfaction, we may value **Instance b** more than others. However, when we consider uniqueness and optimality, we may value **Instance c** more than others.

2 Developing a Formal Problem Specification

1. Notation for describing the problem instance.

(a) Let $S = \{s_1, s_2, \dots, s_n\}$ be the set of students.

(b) Let $E = \{e_1, e_2, \dots, e_n\}$ be the set of employers.

(c) Student's preference list $P(s_i)$, which is a ranked list of employers from most preferred to least preferred.

(d) Employer's preference list $P(e_j)$, which is a ranked list of students from most preferred to least preferred.

(e) A matching is a bijection $M : S \to E$, or a set of pairs $M = \{(s, e)\}$ where s and e are matched uniquely.

(f) A blocking pair (s_i, e_j) is a pair of student-employer that prefer each other over their current match.

   i. s_i prefers e_j over $M(s_i)$.

   ii. e_j prefers s_i over $M(e_j)$.

2. Notation for describing a potential solution

   (a) A set of potential matches $M = \{(s_1, e_1), (s_2, e_2)\}$ of student-employer pairs.

   (b) Valid if and only if every student and employer is assigned exactly one (uniqueness).

3. Good solutions

   (a) Optimality

      i. A solution is student-optimal if it provides the best possible match for every student.

      ii. A solution is employer-optimal if it provides the best possible match for every employer.

   (b) Uniqueness

      i. If a unique stable matching exists, it is the only correct solution.

      ii. If multiple stable matchings exist, we will pick the one that maximizes a criterion (e.g., student happiness, fairness).

   (c) Stability

      i. No blocking pairs exist in the matching.

ii. A matching M is stable if there exists no student-employer pair (s_i, e_j) such that

A. s_i prefers e_j over $M(s_i)$, the employer currently matched with s_i.

B. e_j prefers s_i over $M(e_j)$, the student currently matched with e_j.

3 Identify similar problems. What are the similarities?

1. Marriage Problem (Stable Marriage Problem), where each men has their own primary choice, each women has their own primary choice, and each person has a preference list.

2. Admission-Student problem, where each student has their own primary choice for colleges, and each college has their own primary choice for students, each party also has a preference list.

4 Evaluate simple algorithmic approaches, such as brute force.

1. Brute force algorithm

   (a) Each student has a preference list of employers, and each employer has a preference list of students.

   (b) Employer-optimal choice: For all employers $E = e_1, e_2, \ldots, e_n$ we will access their preference list $P(e_i)$, add all the match to another list M. looping through all of the students $S = s_1, s_2, \ldots, s_n$ and checking each of their own preference list $P(s_i)$ add all the match to another list M.

   (c) Student-optimal choice: For all students $S = s_1, s_2, \ldots, s_n$ we will access their preference list $P(s_i)$, and match it by looping through all of the employers $E = e_1, e_2, \ldots, e_n$ and checking each of their own preference list $P(e_i)$, add all the match to another list M.

2. (a) Bound by the time $t(n)$ we can say that our worst-case scenario of our brute force algorithm is $O(n!)$.
    - The number of ways to match n employers and n students is $n!$.
    - For each employer, we must check each student's preference list and for each student, we must check each employer's preference list, and that is $O(n^2)$.
    - This results in a worst-case time complexity of $O(n! \cdot n^2)$.
    - Since $n!$ grows faster than n^2, we can simplify this to $O(n!)$.

   (b) Each potential solution is a perfect matching between n employers and n students, therefore the total number of potential solutions is $n!$.

   (c) Overall worst-case running time of the brute force will always be $O(n!)$.
    - We generate $n!$ potential solutions.
    - For each solution, we spend $O(n^2)$ time to check if it is stable.
    - Since generating a solution takes negligible time, compared to checking it, the dominant term is

$$O(n! \cdot n^2) = O(n^2)$$

    - Since $n!$ grows faster than n^2, we can simplify this to $O(n!)$.

   Therefore, the brute force algorithm has a factorial time complexity of $O(n!)$, which is extremely inefficient for large n.

5 Design a better algorithm.

We can do this by using the Gale-Shapley algorithm.

1. Input
   (a) Two preference lists, one for students and one for employers.
   (b) The number of students and employers, n.
2. Steps:

(a) Initialization: Create an empty list of matched pairs. All applicants are initially unmatched, and all employers are initially unmatched.

(b) Proposal Phase: Each unmatched applicant proposes to the first employer on their preference list who has not already rejected them.

(c) Employer's response:

   i. Each employer receives proposals and considers them:
      - If they are unmatched, they accept the proposal.
      - If they are already matched but prefer the new applicant over their current match, they reject the current match and accept the new proposal.
      - If they prefer their current match, they reject the new proposal.

(d) Repeat: Applicants who have been rejected by all employers or who haven't yet been matched will propose to the next employer on their list.

(e) Termination: The algorithm terminates when no applicants are left to propose or when everyone is matched. At this point, we have a stable matching.

3. Time complexity:

   (a) Time per proposal, each student proposes to at most n employers, and each employer receives at most n proposals, so the time per proposal is $O(n)$.

   (b) Total time complexity, since there are n students and n employers, the total time complexity is $O(n^2)$.

4. Walkthrough:

   (a) Let $n = 3$, S be the student set, and E be the employer set, M be the matching set, and P be the preference list.

   (b) $S = \{s_1, s_2, s_3\}$ and $E = \{e_1, e_2, e_3\}$, $M = \emptyset$.

   (c) $P(s_1) = \{e_1, e_2, e_3\}$, $P(s_2) = \{e_2, e_1, e_3\}$, $P(s_3) = \{e_3, e_2, e_1\}$.

   (d) $P(e_1) = \{s_1, s_2, s_3\}$, $P(e_2) = \{s_2, s_3, s_1\}$, $P(e_3) = \{s_3, s_1, s_2\}$.

   i. s_1 proposes to e_1, e_1 accepts.
   ii. s_2 proposes to e_2, e_2 accepts.
   iii. s_3 proposes to e_3, e_3 accepts.

(e) Terminate: we the most stable matching, which priotiizes the students.

$$M = \{(s_1, e_1), (s_2, e_2), (s_3, e_3)\}$$

5. Challenges:

   (a) Let $n = 3$, S be the student set, and E be the employer set, M be the matching set, and P be the preference list.

   (b) $S = \{s_1, s_2, s_3\}$ and $E = \{e_1, e_2, e_3\}$, $M = \emptyset$.

   (c) $P(s_1) = \{e_2, e_3\}$, $P(s_2) = \{e_1, e_3\}$, $P(s_3) = \{e_2, e_1\}$.

   (d) $P(e_1) = \{s_1, s_2, s_3\}$, $P(e_2) = \{s_2, s_3, s_1\}$, $P(e_3) = \{s_3, s_1, s_2\}$.

      i. s_1 proposes to e_2, e_2 accepts.
      ii. s_2 proposes to e_1, e_1 accepts.
      iii. s_3 proposes to e_2, e_2 accepts.

$$M = \{(s_1, e_2), (s_2, e_1), (s_3, e_2)\}$$

      iv. Now we have to goto the employer's preference list, since there are two primary choice for student s_1 and s_3, we have to reject one of them.
      v. e_1 stays the same, goes with s_2.
      vi. e_3 rejects s_1 and accepts s_2.
      vii. e_2 rejects s_2 and accepts s_3.

$$M = \{(s_1, e_2), (s_2, e_1), (s_3, e_3)\}$$