稳定匹配算法在高校招生录取系统中的应用

2311095，宋卓伦，计算机科学与技术

2025 年 6 月 10 日，南开大学计算机学院，天津

摘要

本研究应用 Gale-Shapley 算法于高校招生，实现了考生和高校主导的两种版本，在小规模数据集（50 名考生，5 所高校）上验证效果。结果显示，两算法均生成稳定匹配，考生主导更优于考生满意度，高校主导录取分数更高。未来计划测试大规模数据集，为招生优化提供理论与实践参考。

关键词：稳定匹配算法、Gale-Shapley 算法、高校招生、双边匹配

目录

1 引言

“十年寒窗苦读，理想即将实现。” 每年的 6 月份，全国各省市都开始了全国统一的考试——高考。通过这场考试，每一位考生都希望进入自己偏好的知名学府，每一所高校也希望录取优秀考生获得优质生源。

1962 年，Gale 和 Shapley 提出的稳定匹配算法（Gale-Shapley 算法）[1] 通过考虑双方偏好，生成稳定匹配，广泛应用于住院医师分配和学校选择等领域 [2; 3]。基于这套较为广泛应用的算法，本研究决定实现以考生和高校为主导的稳定匹配算法，基于小规模数据验证其在招生录取中的适用性，分析匹配效果，为优化招生系统提供参考。

2 问题描述

这里我们提出运用稳定匹配算法通过双向偏好解决资源配置问题，确保无不稳定对（即无考生和高校更偏好彼此而非当前匹配）的产生。

2.1 符号定义

为了行文规范，我们进行以下统一的符号定义（表1）：

符号	含义
S	考生集合，包含 n 个考生 $\{s_1, s_2, \ldots, s_n\}$
C	高校集合，包含 m 个高校 $\{c_1, c_2, \ldots, c_m\}$
s_i	第 i 个考生，$i = 1, 2, \ldots, n$
c_j	第 j 个高校，$j = 1, 2, \ldots, m$
q_j	高校 c_j 的招生名额
P_{s_i}	考生 s_i 对高校的偏好排序
P_{c_j}	高校 c_j 对考生的偏好排序
M	匹配结果，表示考生与高校的分配关系
$M(s_i)$	考生 s_i 在匹配 M 中分配的高校（若未匹配则为空）
$M^{-1}(c_j)$	高校 c_j 在匹配 M 中录取的考生集合
$\|M(s_i)\|$	考生 s_i 的匹配数量（0 或 1）
$\|M^{-1}(c_j)\|$	高校 c_j 的匹配考生数量（$\leq q_j$）
$c_j >_{s_i} M(s_i)$	考生 s_i 更偏好高校 c_j 而非其当前匹配的高校
$s_i >_{c_j} s_k$	高校 c_j 更偏好考生 s_i 而非考生 s_k

表 1: 形式化定义中的符号说明

2.2 形式化定义

考生集合 $S = \{s_1, \ldots, s_n\}$，高校集合 $C = \{c_1, \ldots, c_m\}$，高校 c_j 有名额 q_j。考生 s_i 和高校 c_j 有偏好排序 P_{s_i} 和 P_{c_j}。匹配 M 满足：

- $\forall s_i \in S, |M(s_i)| \leq 1$
- $\forall c_j \in C, |M^{-1}(c_j)| \leq q_j$

匹配稳定指无 (s_i, c_j) 满足 $c_j >_{s_i} M(s_i)$ 且 $|M^{-1}(c_j)| < q_j$ 或 $\exists s_k \in M^{-1}(c_j)$ 使 $s_i >_{c_j} s_k$。

3 算法设计

3.1 算法概述和数据结构

Gale-Shapley 算法通过申请方（考生或高校）提出请求，接收方根据偏好接受或拒绝，生成稳定匹配。考生主导算法对考生最优，高校主导算法对高校最优。

- **考生类（Student）**：该类包含 ID、姓名、分数、偏好高校列表、当前匹配和申请索引。
- **高校类（College）**：该类包含 ID、名称、名额、偏好考生列表（按分数降序）、当前录取名单。
- **算法类**：包含两种，管理考生和高校列表，执行匹配操作。

3.2 考生主导算法

下面是基于考生选择的主导算法（考生择校），符合考生择校时候的操作，志愿优先：

Algorithm 1 以考生为主导的 Gale-Shapley 算法

输入: 考生集合 S，高校集合 C，偏好 P_{s_i}，P_{c_j}，名额 q_j
输出: 稳定匹配 M

```
初始化：考生未匹配，高校名单为空
while 存在未匹配且有偏高的考生 s do
    s 向下一个偏好高校 c 申请
    if c 未满额 then
        c 接受 s
    else
        if c 更偏好 s 而非最低排名考生 s′ then
            c 接受 s，拒绝 s′
        else
            c 拒绝 s
        end if
    end if
end while
return M
```

3.3 高校主导算法

下面是基于学校选择的主导算法（高校筛选），符合高校筛选时候的操作，分数优先：

Algorithm 2 以高校为主导的 Gale-Shapley 算法

输入: 考生集合 S，高校集合 C，偏好 P_{s_i}，P_{c_j}，名额 q_j
输出: 稳定匹配 M

```
初始化：考生未匹配，高校名单为空
while 存在未满额且有未邀请考生的高校 c do
    c 向下一个未邀请考生 s 发出邀请
```

```
    if s 未匹配 then
        s 接受 c
    else
        if s 更偏好 c 而非当前匹配 c' then
            s 接受 c，拒绝 c'
        else
            s 拒绝 c
        end if
    end if
end while
return M
```

4 算法分析

本部分详细分析以考生为主导和以高校为主导的 Gale-Shapley 算法稳定性的证明，以及在时间复杂度和空间复杂度方面的表现，结合高校招生录取场景的特点，探讨算法效率及其实际应用中的影响因素。

4.1 稳定性证明

- $c_j >_{s_i} M(s_i)$，即考生 s_i 更偏好高校 c_j 而非其当前匹配 $M(s_i)$（若 $M(s_i) = \emptyset$，则任何 c_j 均满足）。
- $|M^{-1}(c_j)| < q_j$（高校 c_j 未满额），或 $\exists s_k \in M^{-1}(c_j)$ 使 $s_i >_{c_j} s_k$（c_j 更偏好 s_i）。

使用反证法，假设算法产生的匹配 M 不稳定，存在不稳定对 (s_i, c_j)，满足上述条件。我们分析 s_i 在算法执行过程中是否向 c_j 申请，推导矛盾。

在考生主导算法中，每个未匹配的考生 s_i 按偏好列表 P_{s_i} 依次向高校，直至匹配或耗尽偏好。考虑 s_i 和 c_j 的情况：

1. **情况 1：s_i 未向 c_j 申请**。若 s_i 未申请 c_j，说明 s_i 在申请更优先的高校 c_k（$c_k >_{s_i} c_j$）时已匹配，并停止申请后续高校。由于 $c_j >_{s_i} M(s_i)$，则 $M(s_i) \neq \emptyset$，且 $M(s_i) >_{s_i} c_j$（因为 s_i 匹配到更优先的高校）。这与假设 $c_j >_{s_i} M(s_i)$ 矛盾。因此，s_i 必然向 c_j 申请。
2. **情况 2：s_i 向 c_j 申请但被拒绝**。若 s_i 申请 c_j 被拒绝，说明：
   - c_j 已满额（$|M^{-1}(c_j)| = q_j$），且 c_j 的最低排名考生 $s_k \in M^{-1}(c_j)$ 满足 $s_k >_{c_j} s_i$（c_j 更偏好 s_k）。
   - 或 c_j 未满额但仍拒绝 s_i（因 s_i 排名低于 c_j 的偏好阈值）。

   分析两种子情况：

   - **子情况 2.1：c_j 已满额**。因 $|M^{-1}(c_j)| = q_j$，假设存在不稳定对，需 $\exists s_k \in M^{-1}(c_j)$ 使 $s_i >_{c_j} s_k$。但 s_i 被拒绝说明 $\forall s_k \in M^{-1}(c_j)$，$s_k >_{c_j} s_i$（算法总是保留更高排名的考生）。这与 $s_i >_{c_j} s_k$ 矛盾。
   - **子情况 2.2：c_j 未满额**。若 $|M^{-1}(c_j)| < q_j$，c_j 拒绝 s_i 仅因 s_i 不满足录取条件（例如，排名过低）。但不稳定对假设要求 $|M^{-1}(c_j)| < q_j$ 时 c_j 接受 s_i，这与算法拒绝 s_i 矛盾。

因此，s_i 被 c_j 拒绝时，不存在 $s_i >_{c_j} s_k$ 或 $|M^{-1}(c_j)| < q_j$，与不稳定对假设矛盾。

3. **情况 3：s_i 向 c_j 申请并被接受**。若 s_i 被 c_j 接受，则 $M(s_i) = c_j$。但不稳定对假设要求 $c_j >_{s_i} M(s_i)$，即 $c_j >_{s_i} c_j$，显然矛盾。因此，此情况不可能。

综合以上，假设存在不稳定对 (s_i, c_j) 总导致矛盾。因此，匹配 M 没有不稳定对，是稳定的。

对于以高校为主导的算法，稳定性证明类似。高校按偏好邀请考生，考生接受更优高校，若存在不稳定对 (s_i, c_j)，可类似分析 c_j 未邀请 s_i 或 s_i 拒绝 c_j 的情况，推导出矛盾。

在小规模实验（50 名考生，5 所高校，总名额 45）中，两种算法的匹配结果均无不稳定对，验证了理论证明的正确性。

4.2 复杂度分析

4.2.1 时间复杂度分析

Gale-Shapley 算法的时间复杂度取决于考生数、高校数以及偏好列表的处理过程。设考生集合规模为 n（即 $|S| = n$），高校集合规模为 m（即 $|C| = m$），以下分别分析两种算法的时间复杂度。

以考生为主导的算法　在以考生为主导的算法中，每个考生按照其偏好列表依次向高校提出申请，高校根据偏好和名额决定接受或拒绝。算法的核心步骤包括：

1. **初始化**：所有考生未匹配，高校录取名单为空，时间为 $O(1)$。
2. **申请循环**：每个考生 s_i 按偏好列表 P_{s_i}（最多包含 m 个高校）逐一申请：
   - 选择下一个偏好高校：$O(1)$。
   - 高校检查名额和偏好：
     - 若高校未满额，直接接受，$O(1)$。
     - 若高校已满额，比较申请考生与最低排名考生，需查找最低排名考生，时间为 $O(q_j)$，其中 q_j 为高校 c_j 的名额。
   - 更新匹配状态（接受或拒绝）：$O(1)$。
3. **终止条件**：所有考生要么匹配成功，要么耗尽偏好列表。

最坏情况下，每个考生申请所有 m 个高校，每次申请涉及 $O(1)$ 或 $O(q_j)$ 的操作。由于 q_j 通常远小于 n（例如，在小规模数据中，$q_j \leq 15$），可以近似认为每次申请的复杂度为 $O(1)$。因此：每个考生最多申请 m 次，总申请次数为 $O(n \times m)$；每次申请的处理时间为 $O(1)$（忽略 q_j 的影响）；总体时间复杂度为：

$$O(n \times m)$$

若考虑高校名额 q_j 的影响，比较最低排名考生的最坏复杂度为 $O(q_j)$，总复杂度可达 $O(n \times m \times \max q_j)$。但在高校招生场景中，$q_j$ 通常为常数级别（例如，5-15），因此仍可简化为 $O(n \times m)$。

以高校为主导的算法　在以高校为主导的算法中，高校按偏好列表向考生发出邀请，考生根据偏好决定接受或拒绝。核心步骤包括：

1. **初始化**：所有考生未匹配，高校名单为空，$O(1)$。
2. **邀请循环**：每个高校 c_j 按偏好列表 P_{c_j}（最多包含 n 个考生）逐一邀请：

- 选择下一个未邀请考生：$O(1)$。
- 考生检查偏好：
  - 若考生未匹配，直接接受，$O(1)$。
  - 若考生已匹配，比较当前高校与现有匹配高校，时间为 $O(1)$（偏好列表查找可优化为哈希表）。
- 更新匹配状态：$O(1)$。

3. **终止条件**：所有高校名额已满或耗尽偏好列表。

最坏情况下，每个高校邀请所有 n 个考生，总邀请次数为 $O(m \times n)$。每次邀请的处理时间为 $O(1)$。因此，总体时间复杂度为：

$$O(n \times m)$$

优化与实际性能 在实际高校招生场景中，偏好列表长度可能小于最大值（例如，考生平均偏好 k 所高校，$k \ll m$）。通过以下优化可降低复杂度：

- **预处理偏好列表**：使用哈希表存储偏好排名，查询时间从 $O(m)$ 或 $O(n)$ 降为 $O(1)$。
- **优先队列**：高校维护当前录取考生的优先队列，查找最低排名考生从 $O(q_j)$ 降为 $O(\log q_j)$。

优化后，时间复杂度可接近 $O(n \times k)$，其中 k 为平均偏好长度。在小规模实验（$n = 50, m = 5$）中，实际运行时间为 1-2 毫秒，验证了算法的高效性。

4.2.2 空间复杂度分析

空间复杂度主要由数据结构和匹配状态的存储需求决定，涉及考生和高校的偏好列表、匹配结果等。以下分析主要存储需求：

- **考生偏好列表**：每个考生 s_i 的偏好列表 P_{s_i} 包含至多 m 个高校，总存储为：
$$O(n \times m)$$
- **高校偏好列表**：每个高校 c_j 的偏好列表 P_{c_j} 包含至多 n 个考生，总存储为：
$$O(m \times n)$$
- **匹配状态**：
  - 考生匹配：每个考生记录当前匹配高校（或空），存储为 $O(n)$。
  - 高校匹配：每个高校 c_j 记录当前录取考生列表，最多 q_j 个考生，总存储为：
  $$O\left(\sum_{j=1}^{m} q_j\right)$$
  在高校招生中，$\sum q_j$ 通常为 $O(n)$（例如，小规模数据中总名额为 45，考生数为 50）。
- **辅助数据结构**：如申请队列或邀请记录，在最坏情况下为 $O(n)$ 或 $O(m)$。

综合以上，空间复杂度为：

$$O(n \times m + m \times q)$$

其中，$q = \max q_j$ 为最大名额，$n \times m$ 项主导存储需求。在实际场景中：若偏好列表较短（例如，平均长度 k），则偏好存储降为 $O(n \times k + m \times k)$。小规模实验（$n = 50, m = 5, q = 15$）中，空间需求约为 $50 \times 5 + 5 \times 15 = 325$ 个存储单元，内存占用可忽略。

4.3 实际应用中的考虑

- **规模影响**：在小规模数据（$n = 50, m = 5$）中，$O(n \times m)$ 复杂度表现高效，但在超大规模（如全国高考，$n \sim 10^7, m \sim 10^3$）中，需并行化实现或增量更新。
- **数据特性**：考生偏好集中于少数热门高校可能导致申请堆积，使用负载均衡可优化。
- **稳定性保证**：两种算法均保证稳定匹配，时间和空间复杂度均衡，适合招生系统。

5 实验设计

数据生成 实验使用小规模数据集：

- 考生：50 名，分数随机生成（120-750，正态分布，符合天津高考）。
- 高校：5 所，名额为 C0:11, C1:21, C2:11, C3:9, C4:10（总名额 45）。
- 考生偏好：全排列 5 所高校。
- 高校偏好：按分数降序排列。

数据存储在 'test_data.txt' 中，格式示例：

```
STUDENTS
S0 Student0 654.187 C3 C4 C1 C2 C0  // 学生编号，学生名，高考分数，意向学校
...
COLLEGES
C0 College0 11  // 大学编号，大学名，招生数目
...
```

实验环境 本次实验环境如下：

- 操作系统：Windows 11（PowerShell）和 Linux （Ubuntu 24.04）
- 编程语言：C++17（g++）

评价指标 这次实验，我们选取了以下指标判断算法好坏：

- 匹配率：成功匹配考生数/总考生数，越高证明大家都有学上；
- 名额使用率：匹配考生数/总名额，越高证明高校分配名额越合理，判断高校数量是否合理；
- 平均满意率：匹配高校在考生偏好中的平均排名，越高证明算法越好；
- 平均录取分数：匹配考生的平均分数，观察分数集中程度；
- 执行时间：算法运行时间，看性能。

指标	数值
考生数量	50
高校数量	5
总招生名额	45
平均偏好数	5
最少偏好数	5
最多偏好数	5

表 2: 小规模数据集

指标	考生主导	高校主导
匹配稳定	是	是
成功匹配数	45	45
匹配率	90.00%	90.00%
名额使用率	100.00%	100.00%
满意度排名	1.5-2.0	2.2-3.0
平均录取分数	433 左右	434 左右
执行时间（秒）	0.001-0.002	0.001-0.002

表 3: 两种算法结果

6 实验结果与分析

对于我们的实验结果，这里的数据如表2所示，规模较小：

接下来分析结果，根据上面的结果我们不难看到：

- 两种算法均产生稳定匹配；
- 考生主导算法的满意度排名较低（1.5-2.0），对考生更优；
- 高校主导算法的录取分数略高，因优先高分考生，也是实际需要上应该采用的方法；
- 名额使用率修复 $\leq 100\%$；
- 执行时间短（1-2 毫秒），适合小规模数据。

7 结论与展望

Gale-Shapley 算法经过上述检验，在一定程度上可以适用于高校招生，考生主导优化满意度，高校主导提升录取质量。C++ 实现高效，验证了小规模数据的有效性。

但是我们的方法也具有一定的局限性：我们的数据是模拟出来的，具体研究情况需要真实数据验证；对于选择的偏好是简化的，尚未考虑复杂场景（例如同分竞争等等），未来我们可以优化算法，进一步实现高级功能。高考是人生重要的大事，录取系统和算法马虎不得！

在完成上述局限性的改进之后，我们会测试真实招生数据。扩展算法支持不完全偏好，最后添加稳定性检查和可视化。

参考文献

[1] Gale, D., & Shapley, L. S. (1962). College Admissions and the Stability of Marriage. *The American Mathematical Monthly*, 69(1), 9-15. https://doi.org/10.2307/2312726

[2] Roth, A. E. (1984). The Evolution of the Labor Market. *Journal of Political Economy*, 92(6), 991-1016. https://doi.org/10.1086/261272

[3] Abdulkadiroğlu, A., & Sönmez, T. (2003). School Choice. *American Economic Review*, 93(3), 729-747. https://doi.org/10.1257/000282803322157061