arXiv:2601.07959v1 [cs.GT] 12 Jan 2026

鲁棒稳定匹配：应对偏好变化

Rohith Reddy Gangam[1], Tung Mai[2],Nitya Raju[*3]，以及 Vijay V. Vazirani[1]

[1]加利福尼亚大学欧文分校 [2]
Adobe Research[3]马里兰大学帕克分校

摘要

我们研究 Gale 和 Shapley 双边稳定匹配设定中对偏好变化具有鲁棒性的稳定匹配。 [GS62]给定两个实例A 和B ，它们涉及相同的一组参与者；如果某个匹配在两个实例下都稳定，则称其为鲁棒匹配。这一概念体现了匹配市场中理想的鲁棒性：在这类市场中，偏好可能随时间演变、被虚报，或受到不确定性的影响。经典稳定匹配理论揭示了单个实例丰富的格结构、算法结构和多面体结构，但当稳定性要求同时适用于多个实例时，这些性质中哪些仍然成立尚不清楚。我们的工作开启了对偏好变化模型日益一般化时，鲁棒稳定匹配的结构特性和计算行为的系统研究。

我们研究一系列扰动模型下的鲁棒性：1. 某个参与者的偏好列表中一次上移，2. 单个参与者进行任意排列变化，以及 3. 双方多个代理人进行任意偏好变化。对于每种情形，我们刻画以下问题何时成立：1. 鲁棒稳定匹配的集合构成子格；2. 鲁棒稳定匹配的格存在一种简洁的伯克霍夫偏序，从而支持高效枚举；3. 能否高效计算工人最优和企业最优鲁棒稳定匹配；4. 鲁棒稳定匹配多面体是否为整数的（通过研究其线性规划公式）。

我们给出明确的反例，展示这些结构性质和几何性质在何处失效，并以 XP 时间算法补充这些结果，其运行时间为$O(n^k)$ ，参数为k，即偏好发生变化的参与者数量。我们的结果精确划定了鲁棒稳定匹配问题中可处理与难处理情形之间的边界。

1 引言

匹配是市场设计中的核心主题，其起源可追溯至 Gale 和 Shapley 于1962年发表的开创性论文 [GS62]，该论文提出了稳定匹配问题，并

[*]这项工作是在作者就读于加利福尼亚大学欧文分校期间完成的。

开启了对基于匹配的市场设计这一更广泛领域的研究。经典的稳定匹配模型假设参与者报告固定的偏好排序，但现实市场很少满足这种不变性：随着新信息的出现、外部条件的变化，或包括虚报和串谋在内的策略行为的产生，偏好可能会发生变化。这些因素自然引出了这样一个问题：匹配在输入发生扰动时会如何表现。在本研究中，我们考察能够在多个偏好配置下保持稳定的匹配——具体而言，即在基础实例A 和经过扰动的实例B中均保持稳定的匹配，我们将其称为稳健稳定匹配。在日益一般化的扰动模型下，我们系统分析了这类匹配的结构、算法和几何性质。

稳定匹配问题的一个核心结构特征是，某个实例的稳定匹配集合构成一个有限分配格，其顺序由一方的偏好决定 [Knu97a，GI89]。该格可通过旋转紧凑地表示；旋转对应于稳定匹配之间的最小变换，也是许多基本算法和结构结果的基础。我们通过比较定义在同一参与者集合上的两个实例所对应的格，考察偏好变化如何影响格结构。具体而言，我们研究这两个实例的稳定匹配的交集是否仍保留经典情形中稳定匹配所具有的格结构。

我们首先考察最简单的非平凡扰动，即一位参与者的偏好列表发生单次移动，并对由此产生的稳定匹配空间给出完整的结构刻画。接着，我们研究单个参与者进行任意置换的情况，随后将分析扩展到多个工人和企业同时修改偏好的实例。这些模型逐步削弱了稳健稳定匹配的格结构，并揭示了该结构得以保留或消失的确切条件。

除了这些结构性结果，我们还研究了鲁棒稳定匹配的算法与几何性质。我们探讨了判定两个实例的稳定匹配交集是否非空的复杂性、如何高效计算该交集中的工人最优匹配和雇主最优匹配，以及是否存在简洁的伯克霍夫表示，以支持对所有鲁棒稳定匹配进行高效枚举。从几何角度来看，我们分析了分数鲁棒稳定匹配多面体，并准确确定其何时具有整数性。我们证明，整数性的丧失与格结构的消失相一致，从而为结构性质和几何性质提供了一个明确且统一的阈值。

本文结果的初步版本曾发表于三篇早期论文 [MV18、GMRV22、GMRV26], ，这些工作分别在逐步更一般的偏好扰动模型下探讨鲁棒性。文献 [MV18]研究了单个代理人的偏好列表发生一次上移时的鲁棒性，并证明了格结构及多项式时间算法的存在。文献 [GMRV22] 将这一研究扩展到单个代理人进行任意置换的情形，同时高效计算相应的伯克霍夫偏序。文献 [GMRV26], 则研究了最一般的情形，即双方的多个代理人都可以任意改变偏好；该文确定了格结构和多面体整性成立的尖锐阈值，并针对一般情形开发了参数化算法。本文将这些结果整合、完善并拓展为一个连贯统一的框架。

1.1 我们的贡献

本文始终考虑包含 n 名工人和 n 家企业的稳定匹配实例，双方的偏好列表均为严格且完备。对于实例 A，令 $\mathcal{M}_A$ 表示其稳定匹配集合，令 $\mathcal{L}_A$ 表示这些匹配在通常的支配序下构成的有限分配格（详见第3.2节）。我们从最简单的变化开始：新实例 B 由 A 经由一名代理人的偏好列表发生一次上移而得到。在此情形下，我们完整描述了该上移操作如何将稳定匹配格划分开来：$\mathcal{M}_A \cap \mathcal{M}_B$ 和 $\mathcal{M}_A \setminus \mathcal{M}_B$ 均构成 $\mathcal{L}_A$的子格。

接下来，我们将分析扩展到一种更一般的模型：在该模型中，单个代理人（无论是工人还是企业）都可以在实例 A 和 B之间任意调整其偏好列表的排列顺序。在这种情形下，尽管 $\mathcal{M}_A \cap \mathcal{M}_B$ 仍是一个子格，但集合 $\mathcal{M}_A \setminus \mathcal{M}_B$ 却不再必然构成 $\mathcal{L}_A$的子格。尽管如此，某些结构仍然存在：它构成 $\mathcal{L}_A$的一个半子格。这一关键性质使我们能够通过高效刻画旋转上的伯克霍夫偏序，找出并简洁地描述在两个实例中都保持稳定的所有匹配（定义见 第3.3节）。

这些结果的一个核心主题是两个实例的格之间的结构关系。这自然引出了以下问题。稳定匹配格在所有有限分配格中具有普适性，这凸显了该问题的重要性：对于每个有限分配格 $\mathcal{L}$，都存在一个稳定匹配实例 A，其格 $\mathcal{L}_A$ 与 $\mathcal{L}$同构。因此，如果以下问题的答案是肯定的，就会为整个类别带来一项新的基本性质。

问题1。 $\mathcal{M}_A \cap \mathcal{M}_B$ 在任意偏好变化下总是 $\mathcal{L}_A$ 和 $\mathcal{L}_B$ 的子格吗？

我们证明，一般而言，答案是“否”。考虑一个鲁棒稳定匹配实例(A,B)，其中p 名工人和q 家企业（共n 名工人和n 家企业）将偏好从A 改为B。在此情形下，我们确定了一个精确的结构边界：$\mathcal{M}_A \cap \mathcal{M}_B$ 恰好在$p \leq 1$ 或$q \leq 1$时，才是 $\mathcal{L}_A$ 和$\mathcal{L}_B$ 的子格；我们还给出一个例子，表明从$(p, q) = (2, 2)$开始，这一性质便不再成立。

除了这些结构性结果，我们还探讨了有关鲁棒稳定匹配的若干基本计算问题和几何问题：

1. 我们能否高效判定 $\mathcal{M}_A \cap \mathcal{M}_B \neq \emptyset$？ 2. 如果 $\mathcal{M}_A \cap \mathcal{M}_B$ 构成一个格，能否高效计算工人最优和雇主最优的鲁棒稳定匹配（定义见 第3.2节）？ 3. $\mathcal{M}_A \cap \mathcal{M}_B$ 是否存在简洁的伯克霍夫偏序，并支持高效枚举？ 4. 分数鲁棒稳定匹配多面体是否为整数的，从而能够采用基于线性规划（LP）的算法？

对于所有满足子格性质的情形，我们还提供了一种基于延迟接受的算法，用于计算鲁棒格中的工人最优匹配和企业最优匹配。我们证明，当 $p \leq 1$或 $q \leq 1$ 时，分数鲁棒稳定匹配多面体仍然是整数的。对于 $(p、q) = (2,2)$，我们给出了一个多面体包含分数顶点的示例

。这产生了一个尖锐阈值，恰好对应于格结构失效的边界，并排除了最一般情形下采用基于线性规划的方法的可能性。作为补充，我们给出了一种以$O(n^{p+q+2})$ 时间运行的 XP 算法，用于判定是否存在鲁棒稳定匹配；若存在，则构造一个鲁棒稳定匹配，并以相同的延迟界枚举所有此类匹配。我们的结果总结见表 1。

(p, q)	**计算 (P?)**	**结构 (格？)**	**几何 (整数的？)**
(0,0)	P [GS62]	是 [IL86]	是 [TS98]
(0,1)	P [定理 23]	是 [定理 4]	是[定理 24]
(0,n)	P [定理 23]	是 [定理 4]	是 [定理 24]
(1,1)	P [定理 23]	是 [定理 19]	是 [定理 24]
(1,n)	P [定理 23]	是 [定理 19]	是 [定理 24]
(2,2)	P [定理 27]	否 [定理 18]	否 [定理 25]
(p,q)	XP:$O(n^{p+q+2})$ [定理。 27]	-	-
(n,n)	NP完全 [MO19]	无 [定理 18]	无 [定理 25]

表 1：稳健稳定匹配的结果总结。p 名工人和q 家企业将其实例A 中的偏好列表置换为B。

“计算”列讨论判定问题 $\mathcal{M}_A \cap \mathcal{M}_B \neq \varnothing$。“结构”列回答 $\mathcal{M}_A \cap \mathcal{M}_B$ 和 $\mathcal{L}_B$ 是否为 $\mathcal{L}_A$ 的子格。
“几何”列回答稳健分数稳定匹配多面体是否为整数的。在$(0,1)$ 和$(0, n)$ 情形下，寻找 Birkhoff偏序有所不同，因为 $\mathcal{M}_A \setminus \mathcal{M}_B$ 在前一种情形下构成半子格，而在后一种情形下不构成半子格。对于$(1,1)$ 和$(1, n)$ 情形，显式构造 Birkhoff偏序的问题仍未解决；然而，我们证明了一种情形的多项式时间可解性意味着另一种情形也具有多项式时间可解性。

2 相关工作

稳定匹配问题由 Gale 和 Shapley 提出 [GS62], 二人还提出了延迟接受（DA）算法，该算法能产生一种稳定匹配，使市场一方获得最优结果，另一方则获得最差结果。多年来，围绕这一问题发展出了丰富的结构理论、算法理论和博弈论理论，进而产生了高效算法以及住院医师分配和学校择校系统等影响深远的应用 [GI89，Rot16，APR09，APRS05]。Knuth [Knu76] 证明了所有稳定匹配构成一个有限分配格。这一奠基性成果为此后诸多进展奠定了基础。包括乡村医院定理和延迟接受算法的激励相容性在内的其他博弈论性质，已在 [Rot82，DF81]中得到确立。关于稳定匹配的几何性质，Teo和Sethuraman [TS98] 证明了稳定匹配多面体的整性

面体的另一种表述见 [Van89，Rot85]。这一领域的奠基性工作意义重大，Roth 和 Shapley 因此荣获2012年诺贝尔经济学奖 [RS12]，；如今，稳定匹配问题已成为算法博弈论的基石；全面论述见 [Knu97b，GI89，Man13，EIV23]。

还有一些研究探讨了更一般或对抗性偏好变化下的鲁棒性和稳定性。Chen等人 [CSS21] 研究了一种基于偏好配置之间交换距离的鲁棒性概念，定义了d-鲁棒匹配，即在有界扰动下仍保持稳定的匹配，并分析了稳定性与社会福利之间的权衡。在代理人可以任意改变其偏好的更一般情形下，Miyazaki和Okamoto [MO19] 考虑了如何寻找一个在两个实例中均稳定的匹配，并将其称为联合稳定匹配；他们还证明，判定是否存在此类匹配是 NP 困难。这些结果凸显了在不受限制的偏好变化下要求鲁棒性所带来的计算挑战。

当市场中只有一方改变偏好时，这种情形便与弱偏好或部分有序偏好下的稳定匹配密切相关。在这一背景下，研究者探讨了若干种稳定性的细化形式，包括弱稳定性、强稳定性和超强稳定性。Irv- ing [Irv94] 证明了弱稳定匹配的存在性，并给出了计算此类匹配的多项式时间算法。Spieker [Spi95] 证明了超稳定匹配集合构成一个分配格，而Manlove [Man02] 将这一结果推广到了强稳定情形。Kunysz等人 [KPG16] 给出了所有强稳定匹配的简洁偏序表示。

另一个相关研究方向关注偏好中的不确定性或部分信息。Aziz等人 [ABG+16, ABF+17] 提出了若干不确定性模型，并研究了在所有实现下均保持稳定的匹配。Genc等人 [GSOS17, GSSO17] 引入了(x,y)-超匹配，这种匹配在发生扰动事件后仍易于修复。尽管这些概念并非本文所指意义上的鲁棒，但它们刻画了互补的韧性特性。

除了鲁棒性之外，稳定匹配框架的多种推广也已得到探索。Chen等人 [CNS18] 研究了多准则（多模态）偏好，而 Menon 和 Larson [ML18] 则考察了如何在弱序的所有补全中，最小化阻塞对的最大数量。增量偏好变化也受到研究：Bredereck等人 [BCK+20] 和 Boehmer 等人 [BHN22a, BHN22b] 分析了随着偏好随时间演变而维持稳定性的算法和结构性质。

受欢迎匹配的背景下也研究了鲁棒性。由G¨ardenfors [G¨ar75], 引入的受欢迎性通过多数投票比较匹配。Bullinger等人近期研究了鲁棒版本。 [BGS24], 他们基于偏好变化的程度，得出了多项式时间结果和困难性结果。Cs´aji [Cs´a24] 进一步将这些思想扩展到了具有多模态偏好和不确定偏好的情形。

最后，计算社会选择领域对输入变化的鲁棒性研究由来已久。大量研究探讨了投票中的微小扰动如何影响选举结果 [FR16，SYE13， BFK+21]，通常采用基于交换的距离度量 [EFS09]。类似思想也被应用于稳定匹配：Boehmer 等人 [BBHN21] 分析了如何通过策略性改变强制实现或阻止特定匹配，而 B´erczi 等人 [BCK24] 则研究了如何利用偏好修改，确保存在具有理想结构特征的稳定匹配

性质。

3 模型与预备知识

3.1 稳定匹配问题与鲁棒稳定匹配问题

一个稳定匹配问题实例由一组n 名工人 $\mathcal{W} = \{w_1,w_2,\ldots,w_n\}$以及一组$n$ 家企业 $\mathcal{F} = \{f_1,f_2,\ldots,f_n\}$组成，统称为参与者。每个参与者$a \in \mathcal{W} \cup \mathcal{F}$都对另一类型的参与者持有一个严格全序（偏好列表），记作 $>_a$。例如，$w^i <_f w^j$ 表示企业f 严格偏好w^j 而非w_i；工人偏好也以类似方式表示。

匹配 M 是 $\mathcal{W}$与 $\mathcal{F}$之间的一一对应。对于每一对 $(w, f) \in M$，我们记 $M(w) = f$、$M(f) = w$，并称他们为 M 下的配偶。若一对 $(w, f) \notin M$ 满足 $f >_w M(w)$ 且 $w >_f M(f)$，则称其为 M 的阻塞对。若一个匹配不存在阻塞对，则称其为稳定匹配。

一个鲁棒稳定匹配实例由 $k \geq 2$ 个定义在相同代理集合 $\mathcal{W}$和 $\mathcal{F}$上的稳定匹配问题实例 $A_1, A_2, \ldots, A_k$ 组成。若匹配 M 在每个 A_i 下都稳定，则称 M 在 $(A_1, \ldots, A_k)$ 下是鲁棒稳定的。我们主要关注 $k = 2$ 的情形，并记作 (A, B)。对于任意实例 I，令 $\mathcal{M}_I$表示稳定匹配的集合，$\mathcal{L}_I$ 表示 I 下的稳定匹配格（将在第3.2节中正式定义）。我们研究的主要对象是鲁棒稳定匹配的集合 $\mathcal{M}_A \cap \mathcal{M}_B$。

我们还将使用以下概念来描述实例之间的“距离”。如果一对（A、B）属于（p、q）型，则表示B 中的偏好与A 中的偏好至多在p 个工人和q 家企业处不同，而在其余 $2n - p - q$ 个参与者处完全相同。（结果关于p 和q是对称的，因此我们只考察 $0 \leq p \leq q \leq n$ 的情形。）

3.2 稳定匹配格

设M 和M' 是在同一实例下的两个稳定匹配。如果每个工人都弱偏好M 中的配偶，而非M'中的配偶，我们称M 支配 M'，记作$M \preceq M'$。等价地，在支配序中，M 是M' 的前驱。

如果稳定匹配M 满足$M \preceq M_1$ 且$M \preceq M_2$，则称M 是M_1 和M_2 的共同前驱。如果它是共同前驱，且不存在其他严格支配它的共同前驱，则称它为最低共同前驱。类似地，可以定义后继和最高共同后继。

稳定匹配上的支配偏序构成一个格 [GI89]。对于任意两个稳定匹配 M_1, M_2，它们的交 $M_1 \wedge M_2$（最低共同前驱）和并 $M_1 \vee M_2$（最高共同后继）都是唯一的。此外，$M_1 \wedge M_2$ 是通过为每位工人分配其在 M_1 和 M_2, 中更偏好的配偶得到的，而 $M_1 \vee M_2$ 是通过为每位工人分配其在 M_1 和 M_2中较不偏好的配偶得到的；两者都是稳定匹配。这些运算满足分配律：对于任意稳定匹配 M, M′、M″,

$$M \vee (M' \wedge M'') = (M \vee M') \wedge (M \vee M'')$$

$$M \wedge (M' \vee M'') = (M \wedge M') \vee (M \wedge M'').$$

该格包含一个唯一的匹配 M_0，它支配所有其他匹配；还包含一个唯一的匹配 M_z，它被所有其他匹配支配。M_0 是工人最优（雇主最劣）稳定匹配，而M_z是雇主最优（工人最劣）稳定匹配。

3.3 伯克霍夫定理与旋转

定义 1。偏序集的闭集（或下集）是一个子集 S，满足：若 $x \in S$ 且 $y \prec x$，则 $y \in S$。

对于偏序集 Π，其闭集族 Π 对并集和交集运算封闭，并构成一个分配格，记作 $L(\Pi)$，其中并和交分别由并集和交集给出。伯克霍夫定理 [B+37, Sta96] 指出，对于任意有限分配格 $\mathcal{L}$，存在一个偏序集 Π，使得$L(\Pi) \cong \mathcal{L}$；我们称 Π 生成 $\mathcal{L}$。

定理 2（伯克霍夫 [B+37]）。每个有限分配格 $\mathcal{L}$ 都同构于某个有限偏序集 Π 的 $L(\Pi)$。

对于稳定匹配格，生成偏序集可以取为由 Irving 引入的旋转 偏序集 [Irv85，IL86]；另见 [GI89]。旋转是一种最小循环变换，可将一个稳定匹配映射到另一个稳定匹配。

设 M 是一个稳定匹配。对于工人 w，令 $s_M(w)$ 表示 w 的偏好列表中满足以下条件的第一个企业 f：相较于其在 M 中的配偶，f 严格偏好 w。令 $next_M(w)$ 表示企业 $s_M(w)$ 在 M 中的配偶。在 M 中暴露的一个旋转 ρ 是一个有序对列表 $\rho = \{(w_0, f_0), (w_1, f_1), \ldots, (w_{r-1}, f_{r-1})\}$，满足对于每个 i，都有 $w_{i+1} = next_M(w_i)$，其中下标按模 r 计算。定义 M/ρ 为这样得到的匹配：保持不在 ρ 中的所有配对不变，并将每个 w_i 与 $f_{i+1} = s_M(w_i)$ 匹配。已知 M/ρ 也是稳定匹配，而从 M 到 M/ρ 的变换称为消去 ρ。

引理 1 ([GI89], 定理 2.5.4)。在从 M_0到 M_z 的任何消去序列中，每个旋转都恰好出现一次。

设 $\rho = \{(w_0, f_0), \ldots, (w_{r-1}, f_{r-1})\}$ 是一个旋转。对于 $0 \le i \le r-1$，我们称 ρ 将 w_i 从 f_i 移至 f_{i+1}，并将 f_i 从 w_i 移至 w_{i-1}。如果企业 f 是 f_i，或者在 w_i 的偏好列表中严格位于 f_i 和 f_{i+1} 之间，我们称 ρ 将 w_i 移至 f 的下方。类似地，如果 w 是 w_i，或者在 f_i 的偏好列表中位于 w_i 和 w_{i-1} 之间，我们称 ρ 将 f_i 移至 w 的上方。

3.4 旋转偏序集

如果在从 M_0 到任一使 ρ 暴露的稳定匹配的每一种消去序列中，旋转 ρ' 都先于另一个旋转 ρ 被消去，则称 ρ' 先于 ρ，记作 $\rho' \prec \rho$。这种优先关系在旋转集合上定义了一个偏序，称为旋转偏序集，记作 Π。

引理 2 ([GI89], 引理 3.2.1)。对于任意工人 w 和企业 f，至多存在一个旋转将 w 配对至 f、将 w 移至 f 的下方，或将 f 移至 w 的上方。此外，如果 ρ_1 将 w 配对至 f，且 ρ_2 将 w 从 f 移开，则$\rho_1 \prec \rho_2$。

引理 3 ([GI89], 引理 3.3.2)。旋转偏序集 Π 至多包含 $O(n^2)$ 个旋转，且可在多项式时间内计算。

稳定匹配与 Π 的闭子集之间存在一一对应：给定一个闭集 S，从 M_0 开始，按照与顺序 Π 一致的任意拓扑序消除 S 中的旋转，即可得到一个稳定匹配 $M(S)$，且每个稳定匹配都可由此得到。我们称 S 生成 $M(S)$ ，并称 Π 生成稳定匹配格。

偏序集的 哈斯图 是一个有向图，其中每个元素对应一个顶点；如果 $x \prec y$ ，且不存在满足 $x \prec z \prec y$ 的 z ，则存在一条边$x \to y$ 。

3.5 子格与半子格

设 $\mathcal{L}$ 是一个具有并与交运算 $\vee$ 和 $\wedge$ 的分配格。

子格 $\mathcal{L}' \subseteq \mathcal{L}$ 是一个子集，使得对于任意 x、$y \in \mathcal{L}'$，$x \vee y \in \mathcal{L}'$ 和 $x \wedge y \in \mathcal{L}'$ 均成立。并半子格 $\mathcal{L}_j \subseteq \mathcal{L}$ 是一个子集，使得对于任意 x、$y \in \mathcal{L}_j$，$x \vee y \in \mathcal{L}_j$。类似地，交半子格 $\mathcal{L}_m \subseteq \mathcal{L}$ 是一个子集，使得对于任意 x、$y \in \mathcal{L}_m$，$x \wedge y \in \mathcal{L}_m$。注意， $\mathcal{L}'$ 是子格，当且仅当它既是并半子格又是交半子格。

3.6 线性规划形式

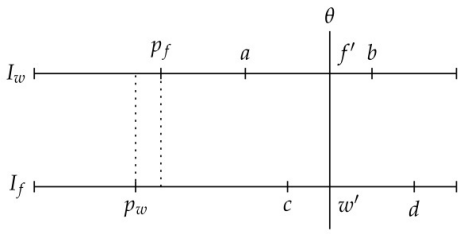


图1：从分数解中求出整数稳定匹配（图片来源： [EIV23]）

关于稳定匹配问题，一个众所周知的结果是，它存在一种线性规划形式，其中所有顶点都是整数的。这提供了一种高效计算稳定匹配的方法。线性规划公式如下，记为 LP（1）。

第三个约束排除了阻塞对：对于任意工人—企业对(w, f)，它确保如果w与其偏好低于f的企业（分数地）匹配，那么f 必须与其偏好高于w的工人匹配。其余约束确保x 定义了一个分数完美匹配。

$$
\begin{aligned}
\text{maximize} \quad & 0 \\
\text{subject to} \quad & \sum_{w} x_{wf} = 1 && \forall f \in \mathcal{F}, \\
& \sum_{f} x_{wf} = 1 && \forall w \in \mathcal{W}, \\
& \sum_{f >_w^A f'} x_{wf'} - \sum_{w' >_f^A w} x_{w'f} \le 0 && \forall w \in \mathcal{W},\ \forall f \in \mathcal{F}, \\
& x_{wf} \ge 0 && \forall w \in \mathcal{W},\ \forall f \in \mathcal{F}.
\end{aligned} \tag{1}
$$

一个x 所对应的线性规划 (1) 的解，可以按如下方式转换为整数完美匹配。构造 $2n$ 个单位区间，每个区间对应一个代理人。对于每个工人w，将其区间I_w 划分为n 个子区间，长度分别为x^{wf}（注意，$\sum_f x_{wf} = 1$），并按照w从最偏好的企业到最不偏好的企业的顺序排列。对每个企业f的区间I_f也采用相同的处理方式，但将子区间按照f从最不偏好的工人到最偏好的工人的顺序排列。

从 $[0,1]$，中均匀随机地选取一个值 θ，并确定每个代理人的区间中包含 θ 的子区间。θ 恰好落在边界上的概率为零，因此我们可以忽略这种情况。

令 $\mu_\theta \colon \mathcal{W} \to \mathcal{F}$ 表示每个工人区间中包含 θ 的子区间所对应的企业，并令 $\mu'_\theta \colon \mathcal{F} \to \mathcal{W}$ 表示企业对应的类似映射。这些函数定义了一个完美匹配，且互为逆函数。此外，对于任意 $\theta \in [0,1]$，它们都是稳定的。

Lemma 4 ([TS98]). *If $\mu_\theta(w) = f$ then $\mu'_\theta(f) = w$.*

引理5 ([TS98])。对于每个 $\theta \in [0,1]$，匹配 μ_θ 都是稳定的。

定理3 ([Van89])。由线性规划 (1) 定义的稳定匹配多面体具有整数顶点；也就是说，它是稳定匹配的凸包。

4 结果

本节介绍我们的结果。我们用 $\mathcal{M}_X$ 表示实例 X 的稳定匹配集合，用 $\mathcal{L}_X$ 表示相应的格。我们主要研究两个稳定匹配实例下的鲁棒稳定匹配，不过这些结果也适用于任意 $k \geq 2$ 个实例。

考虑两个定义在 A 和 B 上的稳定匹配实例，它们的参与者集合包含 n 名工人，$\mathcal{W} = \{w_1, w_2, \ldots, w_n\}$，以及 n 家企业，$\mathcal{F} = \{f_1, f_2, \ldots, f_n\}$。我们首先考虑定理 4 中的情形，该定理阐明了当 A 和 B 仅在一侧的参与者偏好上存在差异时，鲁棒稳定匹配的结构。

定理4。设 A 是一个稳定匹配实例，B 是由 A 改变一侧参与者（工人或企业，但不能两侧同时改变）的偏好列表而得到的另一个实例。那么，$\mathcal{M}_A \cap \mathcal{M}_B$ 中的匹配构成两个格各自的一个子格。

证明。只需证明 $\mathcal{M}_A \cap \mathcal{M}_B$ 是 $\mathcal{L}_A$的子格。设 $|\mathcal{M}_A \cap \mathcal{M}_B| > 1$，并令$M_1$ 和 M_2 是 $\mathcal{M}_A \cap \mathcal{M}_B$中的两个不同匹配。
令 $\vee_A$ 和 $\vee_B$ 分别表示实例 A 和B下的并运算。同理，令$\wedge_A$ 和 $\wedge_B$ 分别表示A 和B下的交运算。

根据并运算的定义（见第3.2节），$\mathrm{M}_1 \vee_A \mathrm{M}_2$ 是这样得到的匹配：按照实例A，为每个工人分配其在 M_1 和 M_2 中较不偏好的配偶。不失一般性，假设 B 是通过仅改变企业的偏好列表而从 A 得到的。由于每个工人在 A 和 B 中的偏好列表相同，因此在这两个实例中，其来自 M_1 和 M_2 的较不偏好的配偶相同。因此，$\mathrm{M}_1 \vee_A \mathrm{M}_2 = \mathrm{M}_1 \vee_B \mathrm{M}_2$。类似的论证表明，$\mathrm{M}_1 \wedge_A \mathrm{M}_2 = \mathrm{M}_1 \wedge_B \mathrm{M}_2$。

因此，$M_1 \vee_A M_2$ 和 $M_1 \wedge_A M_2$ 都属于 $\mathcal{M}_A \cap \mathcal{M}_B$，如所要求的。 □

推论 1. 设 A 是一个稳定匹配实例，$B_1,\ldots,B_k$ 是由 A 得到的实例，每个实例只改变一方参与者的偏好列表，即工人或企业，但不能同时改变双方。则 $\mathcal{M}_A \cap \mathcal{M}_{B_1} \cap \cdots \cap \mathcal{M}_{B_k}$ 中的匹配构成 $\mathcal{L}_A$的一个子格。

证明。假设 $|\mathcal{M}_A \cap \mathcal{M}_{B_1} \cap \cdots \cap \mathcal{M}_{B_k}| > 1$，且 M_1 和 M_2 是该交集中的两个不同匹配。对于每个 $1 \leq \mathrm{i} \leq \mathrm{k}$，$\mathrm{M}_1$ 和 M_2 都属于 $\mathcal{M}_A \cap \mathcal{M}_{B_i}$。根据定理4，每个$\mathcal{M}_A \cap \mathcal{M}_{B_i}$都是 $\mathcal{L}_A$ 的子格。因此，对每个 i，$\mathrm{M}_1 \vee_A \mathrm{M}_2$ 和 $\mathrm{M}_1 \wedge_A \mathrm{M}_2$ 都属于 $\mathcal{M}_A \cap \mathcal{M}_{B_i}$，进而属于它们的交集。 □

这些结果表明，在某些情况下，鲁棒稳定匹配集合构成原始实例的稳定匹配格的一个子格。这促使我们进一步刻画稳定匹配格的子格。在第 4.2.2 节中，我们证明，对于通过置换单个工人或单个企业的偏好列表而得到的任意实例B，集合 $\mathcal{M}_A \setminus \mathcal{M}_B$构成一个半子格 $\mathcal{L}_A$ （见引理21）。特别地，若置换工人的偏好列表， $\mathcal{M}_A \setminus \mathcal{M}_B$ 构成 $\mathcal{L}_A$的并半子格；若置换企业的偏好列表，则它构成 $\mathcal{L}_A$ 的交半子格。在这两种情况下， $\mathcal{M}_A \cap \mathcal{M}_B$ 都是 $\mathcal{L}_A$ 和 $\mathcal{L}_B$的子格，如定理4所示。

基于这一动机，我们首先研究第 4.1 节中的伯克霍夫子格定理。我们建立分配格及其底层偏序的性质，并探讨如何通过压缩偏序来生成子格。当格可以分解为两个子格，或分解为一个子格和一个半子格时，我们设计了能够高效描述这些结构的算法。随后，我们在第 4.2 节中分析日益复杂的模型下稳健稳定匹配的结构性质。 第 4.3 节 介绍了当稳健稳定匹配构成子格时，用于计算工人最优和企业最优稳定匹配的算法。第 4.4 节 研究稳健稳定匹配多面体的几何性质。最后，第 4.5 节 给出了一种 XP 时间算法，用于在最一般的情形下计算和枚举稳健稳定匹配；在此情形中，任意数量的参与者都可能在A 与B之间通过任意置换改变其偏好。

4.1 伯克霍夫关于子格的定理

设 Π 是一个有限偏序集。为简化记法，本文假定 Π 必须有两个虚拟元素 s 和t；其余元素称为真元素，而“元素”一词既指真元素，也指虚拟元素。元素s 先于其他所有元素，而t 后于 Π中的所有其他元素。 Π 的真闭集是指任何包含s 但不包含t的闭集。容易看出， Π 的所有真闭集在集合交与并集运算下构成一个分配格。我们将此格记为$L(\Pi)$。以下定理也称为有限分配格基本定理。

定理 5。 （伯克霍夫 *[B^+37]*）每个有限分配格 $\mathcal{L}$ 都同构于 $L(\Pi)$，其中 Π是某个有限偏序集。

我们应用伯克霍夫定理来研究有限分配格的子格。首先，在定义 2中，我们陈述偏序集压缩偏序集压缩这一关键运算。

定义 2。 给定一个有限偏序集 Π，首先将其元素进行划分；每个子集称为一个元元素。定义元元素之间如下的优先关系：如果 x、y 是 Π 的元素，

其中 x 属于元元素 X，y 属于元元素 Y，且 x 优先于 y，则 X 优先于 Y。假设这些优先关系在元元素上构成一个偏序，记为 Q（否则，该划分对我们的目的而言没有用处）。令 Π' 为元元素上的任意偏序，使得 Q 的优先关系是 Π' 的优先关系的子集。则 Π' 称为 Π 的压缩。令 A_s 和 A_t 分别表示包含 s 和 t 的 Π' 的元元素。

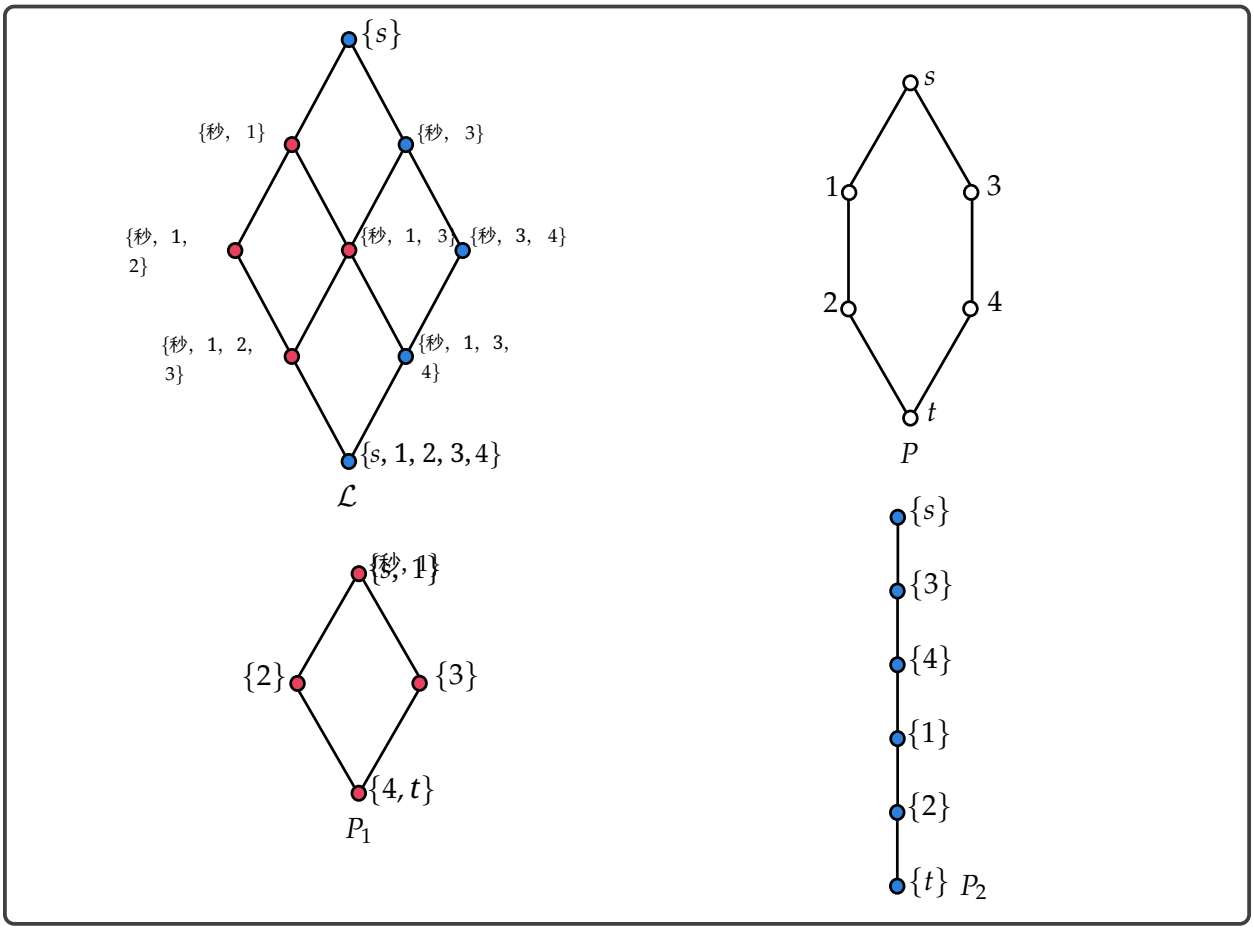


图 2：压缩的两个示例。格 $\mathcal{L} = L(P)$。P_1 和P_2 是P的压缩，它们分别生成 $\mathcal{L}$中由红色和蓝色元素构成的子格。

关于压缩的示例，请参见图 2。显然，A_s 在 Π'中先于所有其他元元素，且A_t 在 Π'中后于所有其他元元素。再一次，我们所说的 Π' 的适当闭集，是指 Π' 中包含A_s 且不包含A_t的闭集。那么，由 Π' 的所有适当闭集构成的格记作$L(\Pi')$。在子格的语境下，压缩非常有用，这归功于定理 6。附录A提供了该定理在稳定匹配语境下的证明。

定理 6. Π 的压缩与 $L(\Pi)$ 的子格之间存在一一对应关系，使得如果 $L(\Pi)$ 的子格 $\mathcal{L}'$ 对应于压缩 Π'，那么 $\mathcal{L}'$ 由 Π'生成。

4.1.1 压缩的另一种视角

本节给出偏序集压缩的另一种定义；本文其余部分将使用这一定义。该定义的优点在于，对于后文介绍的应用而言，它更易于处理。其缺点是，若干不同的边集可能产生相同的压缩。因此，不适合用这一界定来陈述 $\mathcal{L}$ 的子格与 Π的压缩之间存在一一对应关系。最后，我们将证明，利用第一种定义得到的任意压缩 Π' 也可以通过第二种定义得到，反之亦然（命题 7），从而表明这两种定义对于我们的目的而言是等价的。

给定一个稳定匹配实例的偏序集 Π；令 $\mathcal{L}$ 为它生成的格。用$H(\Pi)$ 表示 Π 的哈斯图。考虑以下操作以导出新的偏序集 Π'：选择一组有向边 E，将其添加到$H(\Pi)$ 中，并令H_E 表示所得的图。令H' 为将H_E的强连通分量收缩后得到的图；每个强连通分量都将成为 Π' 的一个元旋转。未被收缩的边将在这些强连通分量上构成一个有向无环图H'。这些边给出了偏序集 Π' 中元旋转之间的优先关系。

令 $\mathcal{L}'$ 为 $\mathcal{L}$ 由 Π'生成的子格。我们称边集 E 定义了 $\mathcal{L}'$。可以看出，每组 E 都唯一确定一个子格 $L(\Pi')$；然而，可能有多组边集定义同一个子格。注意，给定 Π的一个压缩 Π'，可以轻易得到一组边集 E，它定义了 $L(\Pi')$。边集定义子格的示例见图3。

命题 7。偏序集压缩的两种定义是等价的。

证明。设 Π' 是使用第一种定义对 Π 进行压缩所得的结果。显然，对于 Π' 中的每个元旋转，我们都可以向 Π 添加边，使得生成的强连通分量恰好就是该元旋转。在不可比较的元旋转之间引入的任何额外优先关系，也可以通过添加适当的边来引入。

反方向则更简单，因为可以将每个强连通分量定义为一个元旋转，并且可以通过引入新的优先约束来模拟额外添加的边。

□

对于有向边 $e = uv \in E$，u 称为尾，v 称为头。设I 是 Π 的一个闭集。那么我们定义如下：

- I 分离一条边 $uv \in E$，当且仅当 $v \in I$ 且 $u \notin I$。
- I 跨越一条边 $uv \in E$，当且仅当 $u \in I$ 且 $v \notin I$。

如果I 既不分离也不穿过任何边$uv \in E$，则I 称为相对于E的分裂集。

引理 6。设 $\mathcal{L}'$ 是 $\mathcal{L}$ 的子格，E 是定义 $\mathcal{L}'$的一组边。当且仅当生成匹配 M 的闭子集 I 不分离任何边 $uv \in E$ 时，匹配 M 属于 $\mathcal{L}'$。

证明。令 Π' 为与 $\mathcal{L}'$对应的压缩。根据 定理 6，$\mathcal{L}'$ 中的匹配是通过消除 Π'的闭子集中的旋转生成的。

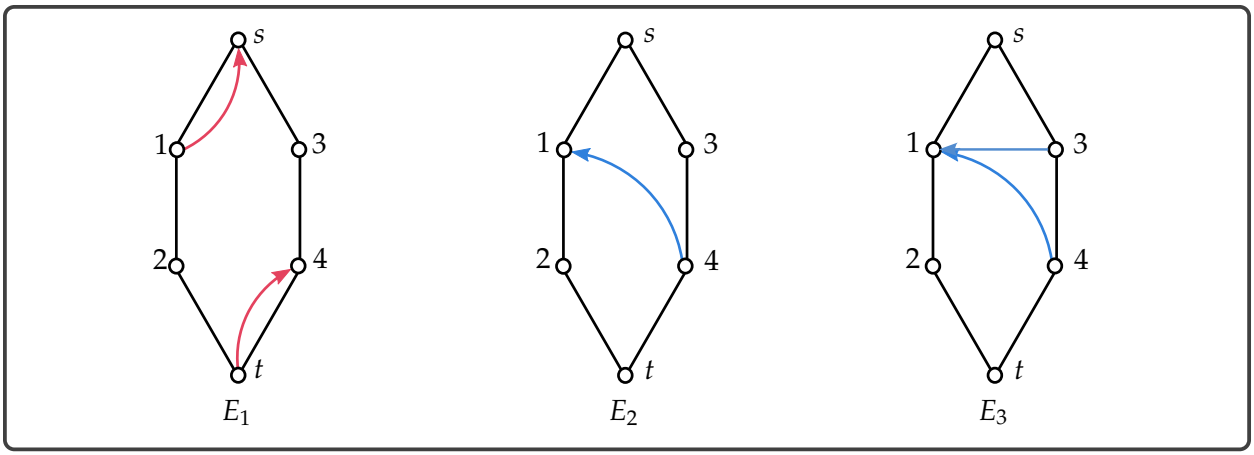


图3： E_1 （红色边）和 E_2 （蓝色边）分别定义了 图 2中的红色和蓝色元素子格。 E_2 和 E_3 定义了相同的压缩，并表示相同的子格。 E_1,E_2 和 E_3 中的所有黑色边均从上向下定向（图中未示出）。

首先，假设 I 将 $uv \in E$分开。此外，为了导出矛盾，假设 $M \in \mathcal{L}'$ ，并令I' 为与 Π' 对应的闭子集M。令U 和 V 分别为包含u 和v 的元旋转。注意，I 和 I' 中的旋转集合相同。因此，$V \in I'$且 $U \notin I'$。由于 $uv \in E$，在U 和 V 之间存在一条边，且该边位于 H'中。因此，I' 不是Π'的闭子集。

接下来，假设 I 不分离任何 uv $\in$ E。我们证明，I 中的旋转可以划分为 Π' 的闭子集 I′ 中的元旋转。如果 I 不能划分为元旋转，则必定存在一个元旋转 A，使得 A ∩ I 是 A 的非空真子集。由于 A 由 H_E 的某个强连通分量中的旋转组成，因此 H_E 中必定存在一条从 A \ I 指向 A ∩ I 的边 uv。因此，I 分离 uv。由于 I 是一个闭子集，uv 不可能是 H 中的一条边。因此，uv $\in$ E，这就产生了矛盾。剩下要证明的是，将 I 划分得到的元旋转集合是 Π' 的闭子集。反设如此，则存在元旋转 U $\in$ I′ 和 V $\notin$ I′，使得 H′ 中存在一条从 U 指向 V 的边。因此，存在 u $\in$ U、 v $\in$ V，且 uv $\in$ E，这就产生了矛盾。

□

注 8。 不失一般性地，我们可以假设定义 $\mathcal{L}'$ 的集合 E 在以下意义下是极小的：不存在边 $uv \in E$ ，使得没有 Π 中的任何闭集将uv 分离。注意，如果存在这样一条边，那么$E \setminus \{uv\}$ 定义出相同的子格 $\mathcal{L}'$。类似地，不存在边$uv \in E$ ，使得每个分离uv 的闭集也会分离E中的另一条边。

定义 3。 对于偏序集 Π 中的元素 v，我们定义 Π 的四个有用子集 :

$$
\begin{aligned}
I_v &= \{r \in \Pi : r \prec v\} \\
J_v &= \{r \in \Pi : r \preceq v\} = I_v \cup \{v\} \\
I'_v &= \{r \in \Pi : r \succ v\} \\
J'_v &= \{r \in \Pi : r \succeq v\} = I'_v \cup \{v\}
\end{aligned}
$$

注意，I_v、J_v、$\Pi \setminus I'_v$、$\Pi \setminus J'_v$ 都是闭集。

引理 7。 对于每个 $uv \in E$，J_v 和 $\Pi \setminus J'_u$ 都将 uv 分离。

证明。 由于 uv 属于 E，u 不可能属于 J_v；否则，不存在能够分隔 uv的闭子集，这与 备注 8矛盾。因此，J_v 对所有 uv 都分隔 uv ，其中 uv属于 E。
同理，由于 uv 属于 E，v 不可能属于 J'_u。因此， $\Pi \setminus J'_u$ 包含 v 但不包含 u，因而分隔 uv。

□

有了这些定义和性质，我们现在可以考察定义子格 L_1 的 L的边集性质，具体取决于 L_1 如何划分 L。在 第 4.1.2 节中，我们考虑 L 被划分为两个子格的情形；在 第 4.1.3 节中，我们考虑 L 被划分为一个子格和一个半子格的情形。

4.1.2 设定 I

在设定 I 下，给定的格 $\mathcal{L}$ 具有子格 $\mathcal{L}_1$ 和 $\mathcal{L}_2$ ，使得 $\mathcal{L}_1$ 和 $\mathcal{L}_2$ 划分 $\mathcal{L}$。此设定的主要结构性事实是：

定理 9。 设 $\mathcal{L}_1$ 和 $\mathcal{L}_2$ 是 $\mathcal{L}$ 的子格，且 $\mathcal{L}_1$ 和 $\mathcal{L}_2$ 划分 $\mathcal{L}$。则存在边集 E_1 和 E_2 ，分别定义 $\mathcal{L}_1$ 和 $\mathcal{L}_2$ ，使它们构成一条从 t 到 s 的交替路径。

我们将在稳定匹配的背景下证明该定理。设 E_1 和 E_2 是分别定义 $\mathcal{L}_1$ 和 $\mathcal{L}_2$ 的任意两个边集。我们将证明，可以对 E_1 和 E_2 进行调整，使它们构成一条从 t 到 s 的交替路径，同时不改变相应的压缩。

引理 8。 必定存在一条从 t 到 s 的路径，该路径由 E_1 和 E_2 中的边组成。

证明。令 R 表示从 t 出发、沿 E_1 和 E_2 中的边可达的顶点集合。反设 R 不包含 s。考虑由$\Pi \setminus R$ 中的旋转生成的匹配 M。不失一般性，设 $M \in \mathcal{L}_1$。根据引理 6， $\Pi \setminus R$ 将一条边 $uv \in E_2$ 分隔开。因此，$u \in R$ 且 $v \in \Pi \setminus R$。由于 $uv \in E_2$, v 也可从 t 沿 E_1 和 E_2 中的边通过一条路径到达。

□

根据引理 8，设 Q 是一条从 t 到 s 的路径。将 Q 划分为子路径 $Q_1, \ldots, Q_k$，使得每个 Q_i 仅包含 E_1 或 E_2 中的边，并且对于所有 $1 \leq i \leq k-1$，$E(Q_i) \cap E(Q_{i+1}) = \emptyset$。令 r_i 为 Q_i 末端的旋转，但 $i = 0$ 时除外，此时 $r_0 = t$。具体而言，Q 中有 $t = r_0 \to r_1 \to \ldots \to r_k = s$。我们将证明，每个 Q_i 都可以替换为一条从 r_{i-1} 到 r_i 的直接边，此外，还可以移除所有不在 Q 中的边。

引理 9. 设 Q_i 仅包含 E_α $(\alpha = 1$ 或 $2)$ 中的边。Q_i 可以替换为一条从 r_{i-1} 到 r_i 的边，其中 $r_{i-1}r_i \in E_\alpha$。

证明。 分隔 $r_{i-1}r_i$ 的闭子集必定会分隔 Q_i 中的一条边。此外，任一闭子集必定恰好分隔 $r_0r_1, \ldots, r_{k-2}r_{k-1}, r_{k-1}r_k$ 中的一条边。因此，分隔 E_1 （或 E_2 ）中边的闭子集集合保持不变。

□

引理 10. $E_1 \cup E_2$ 中但不在 Q 中的边可以删除。

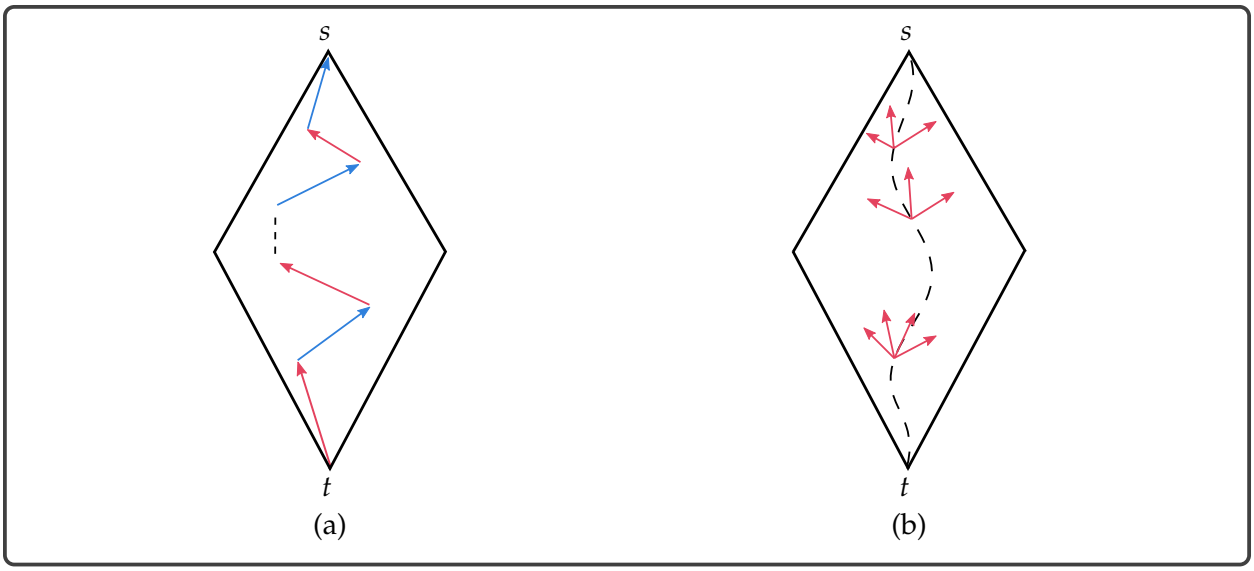


图4：示例：（a）典范路径；（b）花束。

证明。设 e 是 $E_1 \cup E_2$ 中但不在 Q 中的一条边。假设 $e \in E_1$。设 I 是一个将 e 分隔开的闭子集。根据引理 6，由 I 生成的匹配属于 $\mathcal{L}_2$。由于 e 不在 Q 中，且 Q 是一条从 t 到 s 的路径，I 必须还会分隔另一条位于 Q 中的边 e'。根据引理 6，I 不可能同时分隔 E_1 和 E_2 中的边。因此，e' 也必须位于 E_1 中。于是，从 E_1 中移除 e 后，由 I 生成的匹配仍属于 $\mathcal{L}_2$。这一论证适用于所有将 e 分隔开的闭子集。□

根据引理 9和引理 10，$r_0r_1, \ldots, r_{k-2}r_{k-1}, r_{k-1}r_k$ 均为E_1 和E_2 中的边，且它们在E_1 和E_2之间交替。因此，我们得到定理 9。这类路径的示意图见图4（a）。

命题10。存在一个旋转序列 $r_0, r_1, \ldots, r_{2k}, r_{2k+1}$，使得闭子集生成匹配 $\mathcal{L}_1$ 当且仅当它包含 r_{2i} 但不包含 r_{2i+1}，其中某个 $0 \leq i \leq k$。

4.1.3 情形 II

在情形 II 下，给定的格 $\mathcal{L}$ 可以划分为一个子格 $\mathcal{L}_1$ 和一个半子格$\mathcal{L}_2$。我们假设 $\mathcal{L}_2$ 是一个并半子格。显然，将 $\mathcal{L}$中的次序反转，也涵盖了交半子格的情形。下面的定理推广了定理 9，并给出了定义边集E $\mathcal{L}_1$的充分刻画。

定理 11. 存在一个边集 E 定义子格 $\mathcal{L}_1$，使得：

1. E 中各边的尾端集 T_E 构成 Π中的一条链。
2. 不存在由 E 中的边组成的长度为二的路径。
3. 对每个 $r \in T_E$，令

$$F_r = \{v \in \Pi : rv \in E\}.$$

那么，F_r 中的任意两个旋转都不可比较。

4. 对于任意 $r_i, r^j \in T_E$，其中 $r_i \prec r^j$，存在一个分裂集，包含 $F_{r_i} \cup \{r_i\}$中的所有旋转，且不包含 $F_{r_j} \cup \{r_j\}$中的任何旋转。

满足定理 11 的集合 E 称为花束。对于每个 $r \in T_E$，令 $L_r = \{rv \mid v \in F_r\}$。那么 L_r 称为一朵花。注意，花束 E 划分为若干朵花。这些概念如图 4(b) 所示。黑色路径从 s 指向 t，是定理 11 中提到的链；红色边构成 E。注意，边集 E 的边尾端位于这条链上。对于每个这样的尾端，从其出发的 E 中的边构成一朵花。

令E 为定义 $\mathcal{L}_1$的任意边集。我们将证明可以修改E，使其满足定理 11中的条件。令S 为 Π的一个分裂集。换言之，S 是一个闭子集，并且对所有$uv \in E$，u、v 要么都属于S，要么u、v 都属于 $\Pi \setminus S$。

引理 11。$T_E \cap S$ 中存在唯一的极大旋转。

证明。假设 $T_E \cap S$ 中至少有两个极大旋转 $u_1, u_2, \ldots u_k (k \geq 2)$。设 $v_1, \ldots v_k$ 是包含 $u_1, u_2, \ldots u_k$ 的边的头。对于每个 $1 \leq i \leq k$，令 $S_i = J_{u_i} \cup J^{v_j}$，其中 j 是任意满足 $j \neq i$ 的指标。由于 u_i 和 u^j 不可比较，因此 $u_j \notin J^{u_i}$。此外，根据引理 7，$u_j \notin J^{v_j}$。因此，$u_j \notin S_i$。由此可知，S_i 包含 u_i，并将 $u_i v^j$ 分隔开。由于 S_i 将 $u_i v^j \in E$ 分隔开，根据引理 6，由 S_i 生成的匹配属于 $\mathcal{L}_2$。

由于 $\bigcup_{i=1}^{k} S_i$ 包含 $T_E \cap S$ 中的所有极大旋转，并且 S 不会分隔 E 中的任何边，$\bigcup_{i=1}^{k} S_i$ 也不会分隔 E 中的任何边。因此，由 $\bigcup_{i=1}^{k} S_i$ 生成的匹配属于$\mathcal{L}_1$，因而不属于 $\mathcal{L}_2$。这与 $\mathcal{L}_2$ 是并半子格这一事实矛盾。 □

记 r 为 $T_E \cap S$ 中唯一的极大旋转。令

$$R_r = \{v \in \Pi : \text{存在一条从 } r \text{ 到 } v \text{ 的路径，其边属于 } E\}, E_r = \{uv \in E : u, v \in R_r\}, G_r = \{R_r, E_r\}。$$

注意，$r \in R_r$。对于每个 $v \in R_r$，都存在一条从 r 到 v 的路径，且 $r \in S$。由于 S 不会跨越路径中的任何边，v 也必须属于 S。因此，$R_r \subseteq S$。

引理 12. 设 $u \in (T_E \cap S) \setminus R_r$，使得对于 $x \in R_r$，有 $u \succ x$。则我们可以将每条 $uv \in E$ 替换为 rv。

证明。我们将说明 分隔 E 中一条边的闭子集集合保持不变 .

Let I 是分隔uv的闭子集。那么I 也必须分隔rv，因为$r \succ v$。

现在假设I 是分隔rv的闭子集。我们分两种情况讨论：

- 如果$u \in I$，那么I 必须包含x，因为$u \succ x$。因此，I 分隔从r 到x的路径中的一条边。
- 如果$u \notin I$，那么I 分隔uv。

□

根据引理 12 持续替换边，直到不存在$u \in (T_E \cap S) \setminus R_r$ 使得对于某个$x \in R_r$，$u \succ x$。

引理 13. 设

$$X = \{v \in S : v \succeq x \text{ for some } x \in R_r\}.$$

1. $S \setminus X$ 是一个闭子集。*2.* 对于每个 $u \in (T_E \cap S) \setminus R_r$，$S \setminus X$ 都包含 u。*3.* $(S \setminus X) \cap R_r = \emptyset$。*4.* $S \setminus X$ 是一个分裂集。

证明。 该引理由以下断言得出：

断言 1。 $S \setminus X$ 是一个闭子集。

证明。 设v 是$S \setminus X$ 中的一个旋转，u 是v的前驱。由于S 是一个闭子集，$u \in S$。注意，如果一个旋转属于X，则它的所有后继都必须包含在其中。因此，由于$v \notin X$，$u \notin X$。所以，$u \in S \setminus X$。 □

断言 2。 $S \setminus X$ 对$u \in (T_E \cap S) \setminus R_r$中的每个$u$ 都包含。

证明。 根据引理 12替换边后，对于每个$u \in (T_E \cap S) \setminus R_r$，$u$ 都不会后继于任何$x \in R_r$。因此，根据X的定义，$u \notin X$。 □

断言 3. (集合 $\setminus X) \cap R_r = \emptyset$。

证明 $R_r \subseteq X$，(集合 $\setminus X) \cap R_r = \emptyset$。 □

断言 4. 集合 $\setminus X$ 不会分离E中的任何边。

证明。 假设集合 $\setminus X$ 分离了$uv \in E$中的边。那么$u \in X$ 且$v \in$ 集合 $\setminus X$。根据断言 2，u 不可能是尾顶点，这就产生了矛盾。 □

断言 5. 集合 $\setminus X$ 不会与E中的任何边交叉。

证明。 假设集合 $\setminus X$ 与$uv \in E$中的边交叉。那么$u \in$ 集合 $\setminus X$ 且$v \in X$。设J 为分离uv的一个闭子集。于是$v \in J$ 且$u \notin J$。
由于 $uv \in E$ 且 $u \in S$，$u \in T_E \cap S$。因此，$r \succ u$ 根据 引理 11。由于 J 是一个闭子集，$r \notin J$。

由于 $v \in X$，$v \succeq x$ 对 $x \in R_r$成立。同样，由于 J 是一个闭子集，$x \in J$。

因此，J 将从 r 到 x 的 G_r中的路径上的一条边分隔开。因此，所有分隔uv 的闭子集也必须分隔 E_r 中的另一条边。这与 注 8中的假设矛盾。 □

□

引理 14。 E_r 可以替换为以下边集：

$$E'_r = \{rv : v \in R_r\}.$$

证明。 我们将证明，分隔E_r 中一条边的闭子集，与分隔E'_r 中一条边的闭子集相同。

考虑一个分隔$rv \in E'_r$中一条边的闭子集I 。由于$v \in R_r$，I 必须分隔从r 到v的路径上的E 中的一条边。根据定义，该边属于E_r。
现在设 I 是一个分隔$uv \in E_r$中的一条边的闭子集。由于 $uv \in E$，$u \in T_E \cap S$。根据引理 11，$r \succ u$。因此，I 也必须分隔 $rv \in E'_r$。 □

现在我们可以证明定理 11。

定理 11 的证明。首先，令 $S_1 = \Pi$ ，并令 r_1 为根据引理 11 得到的唯一极大旋转。然后，我们可以根据引理 12 和引理 14 替换边。替换后，r_1 是 G_{r_1} 中唯一的尾顶点。根据引理 13，存在一个集合 X，使得 $S_1 \setminus X$ 不包含 R_{r_1} 中的任何顶点，并包含 T_E 中除 r_1之外的所有其他尾顶点。此外，$S_1 \setminus X$ 是一个分裂集。因此，我们可以令 $S_2 = S_1 \setminus X$ 并重复此过程。

设 $r_1, \ldots, r_k$ 是上述过程中找到的旋转。由于对于所有 $1 \le i \le k$ 和 $S_1 \supset S_2 \supset \ldots \supset S_k$，$r_i$ 是 $T_E \cap$ 中唯一的极大旋转，因此我们有 $r_1 \succ r_2 \succ \ldots \succ r_k$。根据引理 14，对于每个 $1 \le i \le k$，E_{r_i} 由边 $r_i v$（其中 $v \in R_{r_i}$）组成。因此，不存在由 E 中的边构成的长度为二的路径，条件 2 得以满足。此外，$r_1, \ldots, r_k$ 恰好是 T_E 中的尾顶点，这满足条件 1。

设 r 是 T_E 中的一个旋转，并考虑 u、$v \in F_r$。此外，假设 $u \prec v$。分离 rv 的闭子集 I 包含 v，但不包含 r。由于 I 是闭子集且 $u \prec v$，I 包含 u。因此，I 也分离 ru，这与注 8 中的假设矛盾。当 $v \prec u$ 时，同样的论证也适用。因此，u 和 v 不可比较，正如条件 3 所述。

最后，令r_i、$r_j \in T_E$，其中$r_i \prec r_j$。根据上述构造，$S_j \supset S_{j-1} \supset$。..... $\supset S_i$，$R_{r_j} \subseteq S_j \setminus S_{j-1}$，并且$R_{r_i} \subseteq S_i$。因此，$S_i$ 包含R_{r_i} 中的所有旋转，但不包含R_{r_j}中的任何旋转，从而满足条件 4。 □

定理 11 自然引出了命题 12，该命题解释了设定 II 中定义子格的唯一边集的结构。

命题 12。 存在一个旋转序列 $r_1 \prec \ldots\ldots \prec r_k$，并且对于每个 $1 \le i \le k$，都存在一个集合 F_{r_i}，使得闭子集生成匹配 $\mathcal{L}_1$，当且仅当该闭子集一旦包含 F_{r_i}中的某个旋转，就必须也包含 r_i。

4.1.4 寻找花束的算法

本节给出一种寻找花束的算法。设 $\mathcal{L}$ 为一个分配格，可以划分为子格 $\mathcal{L}_1$ 和半子格 $\mathcal{L}_2$。给定偏序集 Π 的 $\mathcal{L}$ 以及一个成员关系预言机（用于确定 $\mathcal{L}$ 的匹配是否属于 $\mathcal{L}_1$），该算法将返回定义 $\mathcal{L}_1$的花束。

根据定理 11，尾端 T_E 在 Π中构成一条链 C。我们的算法见算法 1，其思想是按照花在 C 中的顺序寻找它们。具体而言，算法维护一个分裂集 S，使得在任意时刻，S 外的所有花都已被找到。开始时，S 被设为 Π，并随着算法进行而逐渐变小。步骤 2 检查 M_z 是否为 $\mathcal{L}_1$ 中的匹配。如果 $M_z \notin \mathcal{L}_1$，则根据引理 6，闭子集 $\Pi \setminus \{t\}$ 将 E 中的一条边分隔开。因此，C 上的第一个尾端必定是 t。否则，算法跳转到步骤 3 以寻找第一个尾端。每当找到一个尾端 r 时，步骤 5 会立即找到与 r 对应的花 L_r。随后更新分裂集 S，使其不再包含 L_r，但仍包含尚未找到的花。接下来，我们的算法继续在更新后的 S 中寻找下一个尾端。如果找不到尾端，则终止。

FindBouquet(Π)：输入：一个偏序集 Π。输出：由边构成的集合 E，用于定义花束 $\mathcal{L}_1$。1. 初始化：令 $S = \Pi$，$E = \emptyset$。2. 如果 M_z 属于 $\mathcal{L}_1$：转到步骤 3。否则：$r = t$，转到步骤 5。3. r = FindNextTail(Π, S)。4. 如果 r 不为空值：转到步骤 5。否则：转到步骤 7。5. F_r = FindFlower(Π, S, r)。6. 更新：(a) 对每个 $u \in F_r$：$E \leftarrow E \cup \{ru\}$。(b) $S \leftarrow S \setminus \bigcup_{u \in F_r \cup \{r\}} J'_u$。(c) 转到步骤 3。7. 返回 E。

算法 1： 寻找花束的算法。

首先，我们证明一个简单的观察结果。

引理 15。 设 v 是 Π 中的一个旋转。设 $S \subseteq \Pi$，使得 S 和 $S \cup \{v\}$ 都是闭子集。如果 S 在 $\mathcal{L}_1$ 中生成一个匹配，且 $S \cup \{v\}$ 在 $\mathcal{L}_2$ 中生成一个匹配，那么 v 是 E 中一条边的头。如果 S 在 $\mathcal{L}_2$ 中生成一个匹配，且 $S \cup \{v\}$ 在 $\mathcal{L}_1$ 中生成一个匹配，那么 v 是 E 中一条边的尾。

证明。 假设 S 在 $\mathcal{L}_1$ 中生成一个匹配，且 $S \cup \{v\}$ 在 $\mathcal{L}_2$中生成一个匹配。根据引理 6，S 不会分离 E中的任何边，而 $S \cup \{v\}$ 分离边 $e \in E$的情况只有在 u 是 e的头时才会发生。

对于第二种情况，可以给出类似的论证。 □

引理 16。 给定一个分裂集 S，FindNextTail(Π, S) (算法 2) 返回 S 中最大的尾顶点；若 S 中没有尾顶点，则返回Null 。

证明。 令 r 为 S 中最大的尾顶点。

首先，我们证明 $r \in V$。根据 定理 11，E 中边的尾部构成 Π中的一条链。因此 $\Pi \setminus I'_r$ 包含 S中的所有尾端。因此， $\Pi \setminus I'_r$ 不会分隔任何尾端

FindNextTail(Π, S)：**输入：** 一个偏序集 Π，一个分裂集 S。**输出：** S 中最大的尾顶点；如果 S 中没有尾顶点，则返回空值。1. 计算 S 中满足以下条件的旋转 v 的集合 V：

- $\Pi \setminus I'_v$ 在 $\mathcal{L}_1$ 中生成一个匹配。
- $\Pi \setminus J'_v$ 在 $\mathcal{L}_2$ 中生成一个匹配。2. 若 $V \neq \emptyset$ 且 v 是 V 中唯一的极大元素：返回 v。否则：返回空值。

算法 2： 用于寻找下一个尾端的子程序。

属于 S。由于 S 是一个分裂集，$\Pi \setminus I'_r$ 不会分离尾端位于 $\Pi \setminus S$中的任何边。因此，根据 引理 6，$\Pi \setminus I'_r$ 在 $\mathcal{L}_1$中生成一个匹配。根据 引理 7，$\Pi \setminus J'_r$ 必定会分离 E中的一条边，因此根据 $\mathcal{L}_2$ 引理 6生成一个匹配。

根据 引理 15，V 中的任意旋转都必须是 E中某条边的尾端。因此，根据 定理 11，它们全都是 r 的前驱。 □

FindFlower(Π, S,r)：输入：一个偏序集 Π、尾顶点 r，以及一个包含 r 的分裂集 S。输出：集合 $F_r = \{v \in \Pi : rv \in E\}$。1. 计算 $X = \{v \in I_r : J_v$ 在 $\mathcal{L}_1\}$ 中生成一个匹配。2. 令 $Y = \bigcup_{v \in X} J_v$。3. 若 $Y = \emptyset$ 且 $M_0 \in \mathcal{L}_2$：返回 $\{s\}$。4. 计算旋转集合 V，其中 v 属于 S，满足：

- $Y \cup I_v$ 在 $\mathcal{L}_1$ 中生成一个匹配。
- $Y \cup J_v$ 在 $\mathcal{L}_2$ 中生成一个匹配。5. 返回V。

算法 3： 用于寻找花的子程序。

引理 17。 给定一个尾顶点 r 和一个包含 r 的分裂集 S，FindFlower(Π，S，r) *(算法 3)* 能正确返回 F_r。

证明。 首先，我们给出集合Y的两个关键性质。根据定理 11，E 中边的尾部构成一条链 C，位于 Π中。

断言 1。 Y 包含 r 在C中的所有前驱。

证明。假设 C 中至少有一个 r 的前驱，并将 r′ 的直接前驱记为。只需证明 $r' \in Y$。根据定理 11，存在一个分裂集 I，使得 $R_{r'} \subseteq I$ 且 $R_r \cap I = \emptyset$。令 v 为 $C \cap I$ 中的极大元素。那么 v 是 I 中所有尾顶点的后继。因此，J_v 不会分离 I 内 E 中的任何边。因此，$v \in X$。由于 $J_v \subseteq Y$，Y 包含 C 中 r 的所有前驱。 □

断言 2。 Y 不包含F_r中的任何旋转。

证明。 由于Y 是由 $\mathcal{L}_1$ 中生成匹配的闭子集的并集，Y 也会在 $\mathcal{L}_1$ 中生成匹配。根据 引理 6，Y 不会分离 E 中的任何边。由于 $r \notin Y$，Y 必定不包含 F_r 中的任何旋转。

□

根据断言 1，如果$Y = \emptyset$，r 是 C 中找到的最后一个尾端。因此，如果 $M_0 \in \mathcal{L}_2$，s 必定属于 F_r。根据定理 11，F_r 中的头是不可比较的。因此，s 是 C 中唯一的旋转。FindFlower在步骤 3 中正确地返回 $\{s\}$ 。假设这种情况不会发生，我们将证明返回的集合是 F_r。

断言 3。$V = F_r$。

证明。 设 v 是 V中的一个旋转。根据 引理 15，v 是某条边 e 在 E中的头。由于Y 包含 r 在 C中的所有前驱，边 e 的尾必定是 r。因此，$v \in F_r$。
设 v 是 F_r中的一个旋转。由于 Y 包含 r 在 C 中的所有前驱，$Y \cup I_v$ 无法分离尾端为 r 的前驱的任何边。此外，根据定理 11，F_r 中的头是不可比较的。因此，I_v 不包含 F_r中的任何旋转。由于根据上述断言，Y 不包含 F_r 中的任何旋转，$Y \cup I_v$ 不会分离 E 中的任何边。因此，$Y \cup I_v$ 生成一个匹配于 $\mathcal{L}_1$中。最后，$Y \cup J_v$ 显然分离 rv，因此在 $\mathcal{L}_2$中生成一个匹配。因此，$v \in V$，如所需。

□

□

定理 13。 FindBouquet(Π)（见算法 *1*）返回一个定义边集的集合 $\mathcal{L}_1$。

证明。 根据引理 16 和引理 17，只需证明 S 在步骤 6(b) 中得到了正确更新。更准确地说，我们需要

$$S \setminus \bigcup_{u \in F_r \cup \{r\}} J'_u$$

仍然必须是一个分裂集，并且包含所有尚未找到的花。这由引理 13 可得，注意到

$$\bigcup_{u \in F_r \cup \{r\}} J'_u = \{v \in \Pi : v \succeq u \text{ for some } u \in R_r\}.$$

□

显然， $\mathcal{L}$ 的子格也必须是半子格。因此，可以使用 FindBouquet来寻找第 4.1.2 节中所述的典范路径。相同的算法还可用于检查 $M_A \cap M_B = \emptyset$。令 E 为 FindBouquet 算法给出的边集，令 H_E 为将 E 添加到原旋转偏序集的哈斯图Π 中 $\mathcal{L}_A$后得到的相应图。如果 H_E 只有一个强连通分量，则该压缩 Π' 只有一个元元素，并表示空格。

定理14。 如果一个格 $\mathcal{L}$ 可以划分为一个子格 $\mathcal{L}_1$ 和一个半子格 $\mathcal{L}_2$，并且存在一个多项式时间预言机，能够确定任意 $x \in \mathcal{L}$ 是否属于 $\mathcal{L}_1$ 或 $\mathcal{L}_2$，那么可以在多项式时间内找到定义$\mathcal{L}_1$ 的边集。

4.2 鲁棒稳定匹配的结构

本节研究两个稳定匹配实例 A 和 B 的鲁棒稳定匹配集合 $\mathcal{M}_A \cap \mathcal{M}_B$。我们展示该集合如何分解格 $\mathcal{L}_A$，并将分析扩展到多个实例的情况。我们从最基本的情形入手：只有一个代理人从实例 A 到实例 B 的偏好发生变化。首先，我们将这种变化限定为一种称为 上移 的特定操作。接着，我们将其推广到代理人对其偏好列表进行任意排列的情形，最后讨论多个代理人在 A 和 B 之间以任意方式改变偏好的情形。

4.2.1 单个代理：上移

我们从最简单的非平凡偏好变化模型入手，其中单个代理人的偏好通过 上移 操作进行修改。简单来说，上移是指将某个选项在代理人的偏好列表中向前提升，同时保持所有其他选项的相对顺序不变。

形式化地，设A 是一个稳定匹配实例，a 是一个代理人。如果存在两个代理人x 和y ，使得<style id='a 在A 中严格偏好x 而非y ，且a在A中的偏好列表里y 紧邻x 的上方，同时<style id='a列表中所有其他代理人的相对顺序保持不变，则称实例B 由A 通过对代理人a的偏好列表进行上移 得到。A 和B中所有其他代理人的偏好列表均相同。

在这种情况下，定理4 已蕴含鲁棒稳定匹配的集合 $\mathcal{M}_A \cap \mathcal{M}_B$构成 $\mathcal{L}_A$的一个子格。现在，我们通过证明在A 中稳定、但在B中不稳定的匹配的补集 $\mathcal{M}_A \setminus \mathcal{M}_B$也构成 $\mathcal{L}_A$的一个子格，进一步加强这一结构性结果。综合这些结果可知，上移会将稳定匹配格清晰地划分为两个子格。

我们首先证明一个简单的观察结论。

Lem引理 18. 设 $M \in \mathcal{M}_A \setminus \mathcal{M}_B$。在实例 B 下，M 唯一的阻塞对是 (w, f) .

证明。 由于 $M \notin \mathcal{M}_B$，存在一个阻塞对 $(x, y) \notin M$，它是在 B 下的。假设 $(x, y) \neq (w, f)$；我们证明 (x,y) 在 A 下也必定是一个阻塞对。
令 y' 为 x 在 M 中的配偶，并令 x' 为 y 在 M 中的配偶。由于 (x,y) 在 B 下是一个阻塞对，因此有 $x \succ_y^B x'$ 且 $y \succ_x^B y'$。
x 的偏好列表从 A 到 B保持不变，因此 $y \succ_x^A y'$。现在考虑两种情况。

- 如果$y \neq f$，那么y 的偏好列表在A 和B之间没有变化，这意味着$x \succ_{Ay} x'$。因此，(x, y) 是A 下的一个阻塞对。

- 如果$y = f$，那么由于$(x, y) \neq (w, f)$，我们有$x \neq w$。对于所有$x \neq w$的工人，$x \succ_{Bf} x'$意味着$x \succ_{Af} x'$，因为只有w在f的列表中被上移。因此，(x, y)仍然是A下的一个阻塞对。

在这两种情况下，(x, y)在A下阻塞M，这与A实例中M的稳定性相矛盾。 □

回想一下，$w_1 \geq_f w_2 \geq_f \ldots \geq_f w_k$是$k$名位于$w$上方的工人，在$f$的偏好列表中，且$w$的位置向上移动到了$w_1$在$B$中的位置上方。根据引理 18，我们可以据此刻画集合 $\mathcal{M}_A \setminus \mathcal{M}_B$。

引理 19. *$\mathcal{M}_A \setminus \mathcal{M}_B$是$A$中所有稳定匹配的集合，其中$f$与其偏好列表中位于$w_1$和$w_k$之间的配偶匹配，而$w$与其偏好列表中位于$f$下方的配偶匹配。*

证明。假设M是A中的一个稳定匹配，其中包含$w_i f$，且$1 \leq i \leq k$，并且wf'满足$f >_w f'$。在B中，f更喜欢w而非w_i，因此wf是一个阻塞对。因此，M在B下不稳定，并且$M \in \mathcal{M}_A \setminus \mathcal{M}_B$。

为证明另一个方向，设 M 是 $\mathcal{M}_A \setminus \mathcal{M}_B$ 中的一个匹配。根据引理 18，w f 是 M 在 B 中唯一的阻塞对。要出现这种情况，$\mathrm{p}_M(\mathrm{w}) <_{\mathrm{w}}^{B} \mathrm{f}$ 且 $\mathrm{p}_M(\mathrm{f}) <_{\mathrm{f}}^{B} \mathrm{w}$。我们将证明，对于 $1 \leq \mathrm{i} \leq \mathrm{k}$，$\mathrm{p}_M(\mathrm{f}) = \mathrm{w}_i$。反设 $\mathrm{p}_M(\mathrm{f}) <_{\mathrm{f}}^{B} \mathrm{w}_k$，那么 $\mathrm{p}_M(\mathrm{f}) <_{\mathrm{f}}^{A} \mathrm{w}$。因此，w f 是 A 中的一个阻塞对，这就产生了矛盾。 □

令 $\mathcal{L}_A$ 为由 $\mathcal{M}_A$ 构成的工人最优格（即顶元素为工人最优稳定匹配的格）。

定理 15. 集合 $\mathcal{M}_A \setminus \mathcal{M}_B$ 是 $\mathcal{L}_A$ 的子格。

证明。假设 $\mathcal{M}_A \setminus \mathcal{M}_B$ 非空。令 M_1 和 M_2 为 $\mathcal{M}_A \setminus \mathcal{M}_B$ 中的两个匹配。根据引理 19，M_1 和 M_2 都将 f 与其列表中介于 w_1 和 w_k 之间的配偶匹配，并将 w 与其列表中排在 f 之后的配偶匹配。由于 $\mathrm{M}_1 \wedge \mathrm{M}_2$ 是这样得到的匹配：每个工人选择更偏好的配偶，每家企业选择最不偏好的配偶，因此 $\mathrm{M}_1 \wedge \mathrm{M}_2$ 也属于引理 19 所刻画的集合。类似的论证也适用于 $\mathrm{M}_1 \vee \mathrm{M}_2$ 的情况。因此 $\mathcal{M}_A \setminus \mathcal{M}_B$ 是 $\mathcal{L}_A$ 的子格。 □

关于上移模型的更详细结构性结果可见 [MV18]。由于上移是偏好列表任意置换的一种特殊情况，接下来我们转向这一更一般的设定。

4.2.2 单个代理：A 与B之间的任意排列变化

设 A 是一个稳定匹配实例，B 是通过任意置换单个工人或单个企业的偏好列表而得到的实例。引理20 给出了此类置换的一个示例，在该示例中，$\mathcal{M}_A \setminus \mathcal{M}_B$ 并非 $\mathcal{L}_A$的子格，这表明，在第 4.1.2 节 中针对向上移动所获得的结构性保证并不适用于这一更一般的情形。

尽管如此，仍有一些结构得以保留。对于所有此类实例 B，引理21 表明 $\mathcal{M}_A \setminus \mathcal{M}_B$始终构成 $\mathcal{L}_A$的半子格。结合定理4，这意味着，要刻画任意单个代理的偏好变化下的鲁棒稳定匹配，理解半子格既是必要的，也是充分的。

1	b	a	c	d
2	a	b	c	d
3	d	c	a	b
4	c	d	a	b

A中的企业偏好

1	c	a	b	d
2	a	b	c	d
3	d	c	a	b
4	c	d	a	b

B中的企业偏好

a	1	2	3	4
b	2	1	3	4
c	3	1	4	2
d	4	3	1	2

两个实例中的工人偏好实例

图5：一个 $\mathcal{M}_A \setminus \mathcal{M}_B$ 不是 $\mathcal{L}_A$子格的例子。

下一个引理涉及图5所示的例子，其中工人集合为 $\mathcal{W}$ ={a、b、c、d}，企业集合为 $\mathcal{F} = \{1, 2, 3, 4\}$。实例$B$ 由实例A 通过置换企业1的偏好列表得到。

引理20。 $\mathcal{M}_A \setminus \mathcal{M}_B$ 不是 $\mathcal{L}_A$的子格。

证明。$M_1 = \{1a, 2b, 3d, 4c\}$ 和 $M_2 = \{1b, 2a, 3c, 4d\}$ 是相对于实例A的稳定匹配。显然，$M_1 \wedge_A M_2 = \{1a, 2b, 3c, 4d\}$ 在A下也是一个稳定匹配。

从A 变为B时，工人b 和c 在企业1的列表中的位置互换。在B下，$1c$ 是M_1 的阻塞对，$1a$ 是M_2的阻塞对。因此，M_1 和M_2 都属于$\mathcal{M}_A \setminus \mathcal{M}_B$。然而，$M_1 \wedge_A M_2$ 是B下的一个稳定匹配，因此不属于 $\mathcal{M}_A \setminus \mathcal{M}_B$。因此， $\mathcal{M}_A \setminus \mathcal{M}_B$ 在 $\wedge_A$ 操作下不封闭。

□

引理21。 对于通过置换一名工人或一家企业的偏好列表而得到的任意实例 B，$\mathcal{M}_A \setminus \mathcal{M}_B$ 构成 $\mathcal{L}_A$ 的一个半子格。

证明。 假设一家企业f 的偏好列表发生了置换。我们将证明 $\mathcal{M}_A \setminus \mathcal{M}_B$ 是 $\mathcal{L}_A$的一个并半子格。交换工人和企业的角色后，置换一名工人的偏好列表将使 $\mathcal{M}_A \setminus \mathcal{M}_B$ 成为 $\mathcal{L}_A$的一个交半子格。

设M_1 和M_2 是 $\mathcal{M}_A \setminus \mathcal{M}_B$中的两个匹配。因此，它们都不属于 $\mathcal{M}_B$。换言之，在实例B下，它们各自都有一个阻塞对。

设w 是$M_1 \vee_A M_2$中f 的配偶。那么，在M_1 或M_2 中（也可能两者皆是），<style id='w 也必须与f 匹配。不妨假设在M_1中，w 与f 匹配。

设 xy 是 B 下 M_1 的一个阻塞对。我们将证明，在 B 下，xy 也必定是 $M_1 \vee_A M_2$ 的一个阻塞对。首先，企业 y 必须是 f，因为其他偏好列表均未改变。由于在 B 下，x f 是 M_1 的一个阻塞对，因此 x $>_f^B$w。同理，f $>_x$f′，其中 f ′是 x 在 M_1 中的配偶。设 f ″是 x 在 $M_1 \vee_A M_2$ 中的配偶。那么，f′ $\geq_x$f ″。由此可得，f $>_x$f ″。由于 x $>_f^B$w 且 f $>_x$f″，因此在 B 下，x f 必定是 $M_1 \vee_A M_2$ 的一个阻塞对。

□

命题 16。 可以高效地计算出一组边，用于定义由匹配组成的 $\mathcal{M}_A \cap \mathcal{M}_B$ 中的子格 $\mathcal{L}'$。

证明。 我们有 $\mathcal{L}'$ 和 $\mathcal{M}_A \setminus \mathcal{M}_B$ 划分 $\mathcal{L}_A$，其中 $\mathcal{M}_A \setminus \mathcal{M}_B$ 是$\mathcal{L}_A$的半子格，依据 引理21。因此，FindBouquet(Π) 根据 $\mathcal{L}'$，通过 定理 13 找到定义该对象的一组边。

根据引理 3，输入 Π 到 FindBouquet 可以在多项式时间内计算得出。显然，也可以高效实现成员关系预言机，以检查某个匹配是否属于 $\mathcal{L}'$。由于 Π 具有$O(n^2)$ 个顶点（引理 3），FindBouquet 的每一步都能在多项式时间内完成。 □

由此得到以下定理 17，它将在第 4.2.3 节中用于处理参与者偏好在A 和B之间发生变化的情况。

定理 17. 存在一种算法，可在时间$O(|S|p(n))$内检查是否存在相对于任意集合 S 的完全鲁棒稳定匹配，其中 p 是一个多项式函数，前提是 S 中的所有实例相对于某个实例 $A \in S$ 都是 *(0,1)*。此外，若答案为是，则所有此类匹配构成 $\mathcal{L}_A$ 的一个子格，并且我们的算法能找到生成该子格的一个偏序。

证明。 设 A、B_1,.....，B_k 是S中的多项式多个实例。令E_i 为定义$\mathcal{M}_A \cap \mathcal{M}_{B_i}$ 的边集，对所有 $1 \leq i \leq k$均如此。根据推论1， $\mathcal{L}' = \mathcal{M}_A \cap \mathcal{M}_{B_1} \cap \ldots\ldots \cap \mathcal{M}_{B_k}$ 是 $\mathcal{L}_A$的一个子格。
引理 22。 $E = \bigcup i\, E_i$ 定义了 $\mathcal{L}'$。

证明。 根据 引理 6，只需证明对于任意闭子集 I，I 不分离 E 中的一条边，当且仅当 I 在 $\mathcal{L}'$ 中生成一个匹配。
I 不分离 E 中的一条边，当且仅当 I 对所有 E_i 中的 $1 \leq i \leq k$ 均不分离任何边；当且仅当由 I 生成的匹配属于 $\mathcal{M}_A \cap \mathcal{M}_{B_i}$ 中的所有 $1 \leq i \leq k$，根据 引理 6。 □

根据 引理 22，可以从 Π′ 生成 $\mathcal{L}'$ 的压缩可按照第 E 4.1.1 节所述的方法由 构造。根据 命题 16，我们可以计算每个 E_i，从而高效地计算 Π′。显然，可以用 Π′来检查是否存在完全鲁棒稳定匹配。具体而言，存在完全鲁棒稳定匹配，当且仅当存在 Π′ 的真闭子集。当且仅当 s 和 t 属于 Π′ 中不同的元旋转；这是一个易于检查的条件。因此，我们得到 定理 17。 □

4.2.3 多个代理：任意排列变化

现在我们考虑最一般的鲁棒性模型，即（p，q）-鲁棒稳定匹配。形式上，我们允许p 名工人和q 家企业在两个实例A 和B之间任意置换其偏好列表；这两个实例定义在相同的代理集合 $\mathcal{W} = \{w_1, w_2, \ldots\ldots、w_n\}$ 以及 $\mathcal{F} = \{f_1, f_2, \ldots\ldots、f_n\}$上。我们假设 $0 \leq p \leq q \leq n$，因为所有结果对于p 和q都是对称的。

这一设定下的核心问题是，鲁棒稳定匹配的集合 $\mathcal{M}_A \cap \mathcal{M}_B$ 是否构成 $\mathcal{L}_A$ 和 $\mathcal{L}_B$的子格。如定理 4所示，当偏好变化仅限于市场的一侧时——也就是说，当（A，B）属于（0，n）型或（n，0）型时——交集 $\mathcal{M}_A \cap \mathcal{M}_B$ 确实是两个格的子格。

当双方的参与者都可以改变自己的偏好时，这一论证便不再适用。尤其是，$\mathcal{M}_A \setminus \mathcal{M}_B$ 未必构成半子格，不像在(0, 1) 情形下那样；而由 $\mathcal{L}_A$ 和 $\mathcal{L}_B$ 诱导的并与交运算也可能不同。因此，对于匹配M、$M' \in \mathcal{M}_A \cap \mathcal{M}_B$，可能会出现这样的情况：在实例$A$中，一些工人更偏好$M$，而

在实例B中则更偏好M'，而M 是另一种选择——当变化仅限于一方时，不会出现这种现象。

尽管存在这些复杂情况，但事实证明，在许多实例中，交集 $\mathcal{M}_A \cap \mathcal{M}_B$ 仍然是 $\mathcal{L}_A$ 和 $\mathcal{L}_B$的子格。在这种情况下，同一组匹配在这两个实例下具有不同的格结构：虽然底集相同，但格运算$(\vee_A,\ \wedge_A)$和 $(\vee_B,\ \wedge_B)$ 不再相同。图中展示了一个说明这一现象的例子：图 6。

1	a	b	c	d	e	f
2	b	a	c	d	e	f
3	c	d	a	b	e	f
4	d	c	a	b	e	f
5	e	f	a	b	c	d
6	f	e	a	b	c	d

A中的工人偏好

a	2	1	3	4	5	6
b	1	2	3	4	5	6
c	4	3	1	2	5	6
d	3	4	1	2	5	6
e	6	5	1	2	3	4
f	5	6	1	2	3	4

A中的企业偏好

1	b	a	c	d	e	f
2	a	b	c	d	e	f
3	c	d	a	b	e	f
4	d	c	a	b	e	f
5	e	f	a	b	c	d
6	f	e	a	b	c	d

B中的工人偏好

a	1	2	3	4	5	6
b	2	1	3	4	5	6
c	4	3	1	2	5	6
d	3	4	1	2	5	6
e	6	5	1	2	3	4
f	5	6	1	2	3	4

B中的企业偏好

图6：多个代理改变偏好的特殊示例。工人和企业的偏好按从最喜欢到最不喜欢的顺序展示。

实例 B 由实例 A 通过置换工人 1 和 2 以及企业 a 和 b的偏好得到。可以看出，对 $\mathcal{M}_A \cap \mathcal{M}_B$中的任意两个匹配取并（或交），都会得到交集中的另一个匹配，无论考虑的是哪个实例。然而，在 A中，工人 1 和 2 更偏好 M_2 而非 M_1；而在 B中，他们更偏好 M_1 而非 M_2。图 7 展示了 $\mathcal{M}_A \cap \mathcal{M}_B$中匹配的偏序如何随底层实例而变化。

$\mathrm{M}_1 = \{1\mathrm{b}, 2\mathrm{a}, 3\mathrm{d}, 4\mathrm{c}, 5\mathrm{e}, 6\,\mathrm{f}\}$ 和 $\mathrm{M}_2 = \{1\mathrm{a}, 2\mathrm{b}, 3\mathrm{c}, 4\mathrm{d}, 5\,\mathrm{f}, 6\mathrm{e}\}$ 在 A 和 B 中均稳定。$\mathrm{X}_1 = \mathrm{M}_1 \wedge_A \mathrm{M}_2 = \{1\mathrm{a}, 2\mathrm{b}, 3\mathrm{c}, 4\mathrm{d}, 5\mathrm{e}, 6\,\mathrm{f}\}$ 和 $\mathrm{X}_2 = \mathrm{M}_1 \vee_A \mathrm{M}_2 = \{1\mathrm{b}, 2\mathrm{a}, 3\mathrm{d}, 4\mathrm{c}, 5\,\mathrm{f}, 6\mathrm{e}\}$。$\mathrm{Y}_1 = \mathrm{M}_1 \wedge_B \mathrm{M}_2 = \{1\mathrm{b}, 2\mathrm{a}, 3\mathrm{c}, 4\mathrm{d}, 5\mathrm{e}, 6\,\mathrm{f}\}$ 和 $\mathrm{Y}_2 = \mathrm{M}_1 \vee_B \mathrm{M}_2 = \{1\mathrm{a}, 2\mathrm{b}, 3\mathrm{d}, 4\mathrm{c}, 5\,\mathrm{f}, 6\mathrm{e}\}$。
然而，正如定理 18 所示，交集 $\mathcal{M}_A \cap \mathcal{M}_B$ 并不总是子格。图 8给出了一个具体示例。因此，虽然子格性质对$(0,n)$成立，但对$(2,\,2)$则不再成立。

定理 18. 当$(A、B)$ 属于(p,q) 类型且 p、$q\ \geq 2$时，集合 $\mathcal{M}_A \cap \mathcal{M}_B$ 并不总是 $\mathcal{L}_A$ 和 $\mathcal{L}_B$的子格。

证明。图 8 中的例子表明，在$(2,\,2)$中，交集不一定是子格。

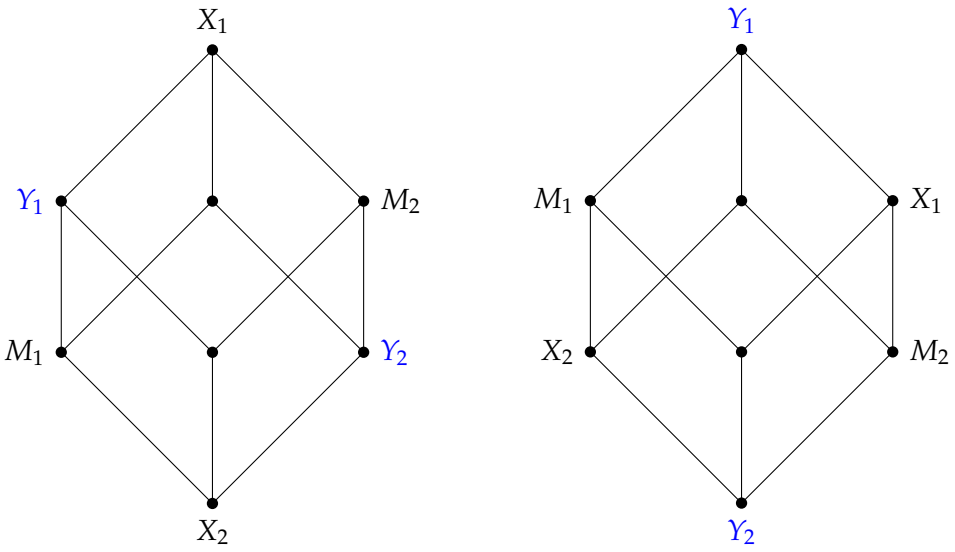


图 7：格 $\mathcal{L}_A$ 和 $\mathcal{L}_B$ 例如见图6

实例B 与A 的区别在于工人 3 和 4 以及企业c 和d的偏好。

考虑匹配 $M_1 = \{1a、2b、3c、4d\}$ 和 $M_2 = \{1b、2a、3d、4c\}$，它们在 A 和B下均稳定。然而，$M = M_1 \wedge_A M_2 = \{1a、2b、3d、4c\}$ 在B下并不稳定，因为$a >_4 c$ 和 $4 >_a 1$，因此 $4a$是B下M 的一个阻塞对。

因此，如果每一侧至少有两名参与者在A 和B中的偏好不同，$\mathcal{M}_A \cap \mathcal{M}_B$ 就不是 $\mathcal{L}_B$的子格。

□

这引出了一个问题：子格性质会在何处失效？我们证明，只要同一侧改变偏好的参与者不超过一名，子格性质仍然成立。具体而言，对于$(\mathbf{1},n)$，交集 $\mathcal{M}_A \cap \mathcal{M}_B$ 仍然是一个子格。

设A 和B 是一个（$\mathbf{1},n$）鲁棒稳定匹配实例，其中只有w_1 以及所有企业从A 到B改变了偏好。

定理 1　**9.** 若 A 和 B 属于（$1,n$）类型，则 $\mathcal{M}_A \cap \mathcal{M}_B$ 是 $\mathcal{L}_A$ 和 $\mathcal{L}_B$ 的子格。

证明。 不失一般性，假设 $|\mathcal{M}_A \cap \mathcal{M}_B| > 1$，并令$M_1$ 和M_2 是 $\mathcal{M}_A \cap \mathcal{M}_B$中的两个不同匹配。令 $\vee_A$ 和 $\vee_B$ 分别表示在A 和B下的并运算。同样，令 $\wedge_A$ 和 $\wedge_B$ 分别表示在A 和B下的交运算。

并$M_1 \vee_A M_2$ 通过依据实例A，将每位工人与M_1和M_2，中其较不偏好的配偶匹配而得到。由于工人w_2,w_3,. . . , w_n 在A 和B中的偏好相同，他们在A 和B 下较不偏好的配偶也相同。因此，对所有$w \neq w_1$,，都有：

$$(M_1 \vee_A M_2)(w) = (M_1 \vee_B M_2)(w).$$

由于 $M_1 \vee_A M_2$ 和 $M_1 \vee_B M_2$ 分别是各自实例下的稳定匹配，因此它们也是完美匹配。因此，剩下的工人 w_1 必须与同一家企业匹配

1	a	b	c	d
2	b	c	a	d
3	d	c	a	b
4	c	d	a	b

A中的工人偏好

a	2	4	1	3
b	1	2	3	4
c	3	4	2	1
d	4	3	2	1

A中的企业偏好

1	a	b	c	d
2	b	c	a	d
3	c	d	a	b
4	d	a	c	b

B中的工人偏好

a	2	4	1	3
b	1	2	3	4
c	4	2	3	1
d	3	4	1	2

B中的企业偏好

图 8: $\mathcal{M}_A \cap \mathcal{M}_B$ 不是 $\mathcal{L}_A$的子格。偏好列表按降序排列。偏好从 A 变为 B 的参与者以红色标出。所有示例均使用此记法。

在两种匹配中:

$$(M_1 \vee_A M_2)(w_1) = (M_1 \vee_B M_2)(w_1).$$

因此，$M_1 \vee_A M_2 = M_1 \vee_B M_2$。类似的论证表明，$M_1 \wedge_A M_2 = M_1 \wedge_B M_2$。因此，在实例$A$ 和 B 下，并与交运算等价，并且$M_1 \vee_A M_2$和$M_1 \wedge_A M_2$ 均属于 $\mathcal{M}_A \cap \mathcal{M}_B$。定理得证。

□

因此，当实例的类型为 $(1,n)$ 时，定理 6 保证存在一个生成相应子格的偏序集。然而，与$(0,1)$ 情形不同，定理14 不适用于类型为$(0,n)$的实例。

引理23。 如果 A 和 B 的类型为$(0,n)$，那么 $\mathcal{M}_A \setminus \mathcal{M}_B$ 并不总是 $\mathcal{L}_A$的半子格。

证明。考虑图 9 所示的实例 A 和 B。工人集合为 $\mathcal{W} = \{1,2,3,4,5\}$，企业集合为 $\mathcal{F} = \{$a、b、c、d、e$\}$。实例 B 是通过置换企业 b 和 c 的列表，从实例 A 得到的。考虑匹配 $M_1 = \{$1b、2a、3c、4d、5e$\}$ 和 $M_2 = \{$1a、2b、3d、4c、5e$\}$，它们相对于实例 A 都是稳定匹配。相对于实例 B，M_1 存在阻塞对 (5, c)，而 M_2 存在阻塞对 (4, b)。因此，M_1 和 M_2 都属于 $\mathcal{M}_A \setminus \mathcal{M}_B$。然而，相对于 A 的并，即 $M_1 \vee_A M_2 = \{$1b、2a、3d、4c、5e$\}$，相对于 A 和 B 都是稳定的。

同样，我们可以看到，$M_3 = \{1b、2c、3d、4a、5e\}$ 和$M_4 = \{1d、2a、3b、4c、5e\}$ 相对于A是稳定的，但在B下分别有阻塞对$(1, a)$ 和$(4,d)$ 。然而，它们的交，$M_3 \wedge_A M_4 = \{1b、2a、3d、4c、5e\}$，相对于 A 和B均是稳定的。

因此， $\mathcal{M}_A \setminus \mathcal{M}_B$ 不是 $\mathcal{L}_A$的半子格。 □

注意，由于A 和B 在图 9 中实际上是$(0, 2)$，这为$\mathcal{M}_A \setminus \mathcal{M}_B$ 是 $\mathcal{L}_A$的半子格时提供了一个紧界。

1	a	b	d	c	e
2	b	a	c	d	e
3	c	d	b	a	e
4	b	d	c	a	e
5	c	e	a	b	d

在A 和B中的工人偏好

a	4	2	1	3	5
b	3	1	2	4	5
c	2	4	3	1	5
d	1	3	4	2	5
e	5	1	2	3	4

在A中的企业偏好

a	4	2	1	3	5
b	1	4	2	3	5
c	4	5	3	1	2
d	1	3	4	2	5
e	5	1	2	3	4

在B中的企业偏好

图 9: $\mathcal{M}_A \setminus \mathcal{M}_B$ 不是 $\mathcal{L}_A$的半子格。

这表明定理14 不能直接用于在$(0, n)$ 情形下求出 Birkhoff 偏序。为解决这一问题，我们定义n 个混合实例B_1、……、B_n，使得对于每个$i \in \{1, 2, 、……、n\}$，数对(A, B_i) 都属于$(0, 1)$类型。具体而言，实例B_i 中代理人x 的偏好概况定义如下：

$$>_x^{B_i} = \left\{ \begin{array}{ll} >_x^B, & \text{if } x = f_i \\ >_x^A, & \text{if } x \neq f_i \end{array} \right\}$$

也就是说，实例B_i 与实例A完全相同，只是企业f_i 的偏好被替换为实例B中的偏好。

如此定义后，定理20 表明，同时在A 和B 下稳定的匹配，恰好是在A 以及所有混合实例B_1, B_2、…、B_n下稳定的匹配。

定理20。 $\mathcal{M}_A \cap \mathcal{M}_B = \mathcal{M}_A \cap \mathcal{M}_{B_1} \cap \mathcal{M}_{B_2} \cap \ldots \cap \mathcal{M}_{B_n}$.

证明。 我们将证明：对于完美匹配而言，在实例 A 或实例 B 的偏好列表下，任何阻塞对都是实例 A 或$B_1, \ldots$、B_n中至少一个实例的偏好列表下的阻塞对；反之亦然。

设 M 是一个完美匹配，在 A 或 B 的偏好列表下存在一个阻塞对。如果它在 A 下存在阻塞对，那么显然 M 不属于 $\mathcal{M}_A \cap \mathcal{M}_{B_1} \cap \ldots \cap \mathcal{M}_{B_n}$。假设 (w, f) 是 B 下的一个阻塞对，且 f $>_w^B$ M(w)，因此 w $>_f^B$M(f)。所有实例中的工人偏好列表都相同，所以在每个实例 B_i中，w 都更喜欢 f 而非 M(w)。企业 f 在 B 和 B_k 中的偏好列表相同，其中某个 k $\in \{1, \ldots, n\}$。于是 w $>^{B_k}$f M(f)，且 (w, f) 是实例 B_k 的阻塞对，因此 M 不是 $\mathcal{M}_A \cap \mathcal{M}_{B_1} \cap \ldots \cap \mathcal{M}_{B_n}$ 中的稳定匹配。

设 M 是一个完美匹配，在实例 A、$B_1, \ldots$、B_n之一的偏好列表下存在阻塞对(w, f) 。与前一种情况一样，如果(w, f) 在A的偏好列表下是阻塞对，那么M 在 $\mathcal{M}_A \cap \mathcal{M}_B$ 中就不是稳定匹配。否则，存在某个实例B_k，使(w, f) 成为阻塞对。我们将证明(w, f) 必定是B的阻塞对。

假设f 不是在B 和B_k中偏好列表相同的代理人。那么w 和f在A 和B_k中的偏好列表都相同，因此这是A的一个阻塞对。否则，必定是f 在B_k 中更改了其偏好列表，并且在B中具有相同的更改后列表。于是(w, f) 也是B 的一个阻塞对，因为$f >_w^B M(w)$ 且$w >_f^B M(f)$。由此可得，这两个集合相等。

□

定理 21。 如果 A 和 B 属于$(0,n)$类型，那么可以高效地计算生成 *Birkhoff* 偏序的 $\mathcal{M}_A \cap \mathcal{M}_B$，并且还可以通过多项式延迟枚举该集合中的匹配。

证明。 由于每个 B_i 与实例 A 恰好在一个代理人的偏好上不同，根据 命题 16，可以高效地计算定义E_i 的边集 $\mathcal{M}_A \cap \mathcal{M}_{B_i}$。根据引理 22 和定理 17，并集$E = \bigcup_i E_i$ 定义了子格 $\mathcal{M}_A \cap \mathcal{M}_B$，并可用于高效枚举该交集中的所有匹配。

□

对于$(A、B)$ 属于$(1，n)$类型的情形，也可以尝试采用类似的技巧。令$w_1 \in \mathcal{W}$和f_1、……、$f_n \in \mathcal{F}$分别表示偏好在A 与B之间发生变化的工人与企业。对于任意子集$X \subseteq \mathcal{W} \cup \mathcal{F}$，令$B_X$ 表示这样一个实例：X 中的每个参与者都具有与B中相同的偏好，而所有其他参与者都具有与A中相同的偏好。于是：

定理 22。 $\mathcal{M}_A \cap \mathcal{M}_B = \mathcal{M}_A \cap \mathcal{M}_{B_{\{w_1,f_1\}}} \cap \mathcal{M}_{B_{\{w_1,f_2\}}} \cap \cdots \cap \mathcal{M}_{B_{\{w_1,fn\}}}$。

证明。 $(\supseteq)$ 令 $\mu /\in \mathcal{M}_A \cap \mathcal{M}_B$。则相对于$A$ 或B， μ 有一个阻塞对。如果是A，那么它显然不在 $\mathcal{M}_A \cap \mathcal{M}_{B_{\{w_1,f_1\}}} \cap \mathcal{M}_{B_{\{w_1,f_2\}}} \cap \cdots \cap \mathcal{M}_{B_{\{w_1,fn\}}}$ 中。因此，假设相对于B，它有一个阻塞对$(w、f)$。

假设$w = w_1$且$f \in \{f_1,……、f_n\}$。因此$f >_w^B \mu(w)$且$w >_f^B \mu(f)$。但这正是$B_{\{w,}$中w和$f\}$的偏好，因此 $\mu /\in \mathcal{M}_A \cap \mathcal{M}_{B_{\{w_1,f_1\}}} \cap \mathcal{M}_{B_{\{w_1,f_2\}}} \cap \cdots \cap \mathcal{M}_{B_{\{w_1,fn\}}}$。
否则，如果$w \neq w_1$且$f \in \{f_1,……、f_n\}$，那么w在A和B中具有相同的偏好；在每个$B_{\{w_1,f\}}$中也是如此，而f在B和$B_{\{w_1f\}}$中具有相同的偏好。因此，$(w、f)$是实例$B_{w_1、f}$的阻塞对。

$(\subseteq)$ </style}同样，考虑 /，$_{f_1} \cap \cdots \cap \mathcal{M}_{B_{w_1,\ fn}}$。那么 μ 有一个阻塞对 $(w，f)$。与前一种情况一样，假设该阻塞对并非相对于 A。那么它相对于某个实例 $B_{\{w_1f_i\}}$。

有三种情况：

1. w $\neq$ w_1 且 f $=$ f_i：那么 f_i 在 B 和 $B_{w_1、}$ f_i 中的列表相同，w 在 A、B 和 $B_{w_1、}$ f_i 中的列表相同；因此这相对于 B 是一个阻塞对。2. w $=$ w_1 且 f $\neq$ f_i：那么 w_1 在 A、B 和 $B_{\{w_1,f_i\}}$ 中的列表相同，并且 f 在 A、B 和 $B_{\{w_1,f_i\}}$ 中的列表相同；因此这相对于 B 是一个阻塞对。3. w $=$ w_1 且 f $=$ f_i：那么 w_1 在 B 和 $B_{\{w_1,f_i\}}$ 中的列表相同，并且 f 在 B 和 $B_{\{w_1,f_i\}}$ 中的列表相同；因此这相对于 B 是一个阻塞对。

另一种关于$w \neq w_1$ 和$f \neq f_i$ 的情况不会出现，因为那样一来，(w,f) 就会是A的一个阻塞对，但这不可能，因为 $\mu \in \mathcal{M}_A$。
因此，(w,f) 是相对于B的一个阻塞对，并且 $\mu /\in \mathcal{M}_B$；所以它不属于 $\mathcal{M}_A \cap \mathcal{M}_B$。
我们得出结论：这些集合相同。

□

推论 2. 对于$(1,n)$ 情形，计算 *Birkhoff* 偏序属于 $\mathcal{P}$，当且仅当计算$(1,1)$ 情形的偏序属于 $\mathcal{P}$。

与(0,1) 情形不同，根据 $\mathcal{M}_A \setminus \mathcal{M}_B$，引理24表明，当$(A, B)$ 属于(1,1)类型时，集合并不总是半子格。因此，定理14 无法应用，而如何刻画子格结构仍是一个开放问题。

引理24 . 如果 A 和 B 属于(1,1)类型，那么 $\mathcal{M}_A \setminus \mathcal{M}_B$ 并不总是 $\mathcal{L}_A$ 的半子格 .

证明。考虑图10中给出的实例 A 和 B。工人集合为 $\mathcal{W} = \{1, 2, 3, 4, 5\}$，企业集合为 $\mathcal{F} = \{$a、b、c、d、e$\}$。实例 B 是通过对企业 c 和工人 3 的偏好列表进行置换，从实例 A 得到的。考虑匹配 $M_1 = \{$1b、2c、3a、4d、5e$\}$ 和 $M_2 = \{$1a、2b、3c、4e、5d$\}$，它们相对于 A 都是稳定的。对于实例 B，M_1有阻塞对 (3, d)，而 M_2 有阻塞对 (4, c)。因此，M_1 和 M_2 都属于$\mathcal{M}_A \setminus \mathcal{M}_B$。然而，相对于 A 的并 $M_1 \vee_A M_2 = \{$1a、2b、3c、4d、5e$\}$，对于 A 和 B 都是稳定的。

同样，我们看到，在B下，$M_3 = \{1c、2a、3b、4d、5e\}$和$M_4 = \{1b、2c、3a、4e、5d\}$相对于A是稳定的，但分别存在阻塞对$(3, d)$和$(4,c)$。然而，它们的交，$M_3 \wedge_A M_4 = \{1c、2a、3b、4e、5d\}$，相对于$A$和$B$都是稳定的。
因此，$\mathcal{M}_A \setminus \mathcal{M}_B$ 不是 $\mathcal{L}_A$的半子格。 □

1	a	b	c	d	e
2	b	c	a	d	e
3	c	a	b	d	e
4	d	c	e	a	b
5	e	d	a	b	c

A中的工人偏好

a	2	3	1	4	5
b	3	1	2	4	5
c	1	2	3	4	5
d	5	3	4	1	2
e	4	5	1	2	3

A中的企业偏好

1	a	b	c	d	e
2	b	c	a	d	e
3	c	d	a	b	e
4	d	c	e	a	b
5	e	d	a	b	c

B中的工人偏好

a	2	3	1	4	5
b	3	1	2	4	5
c	1	4	3	2	5
d	5	3	4	1	2
e	4	5	1	2	3

B中的企业偏好

图10： $\mathcal{M}_A \setminus \mathcal{M}_B$ 不是 $\mathcal{L}_A$的半子格，即使(A, B) 为(1，1)类型也是如此。

4.3 计算工人最优和雇主最优鲁棒稳定匹配的算法

定理 19 表明，当实例(A, B) 为(1，n)类型时，鲁棒稳定匹配集合构成 $\mathcal{L}_A$ 和 $\mathcal{L}_B$的子格。这一结构性质意味着工人最优和企业最优鲁棒稳定匹配的存在性是明确的。本节将介绍用于计算这些极端匹配的算法。

旋转偏序集通过刻画匹配之间的最小变换，为遍历稳定匹配格提供了强有力的描述。然而，它本身并不能

生成鲁棒稳定匹配。因此，必须先使用显式算法计算至少一个鲁棒稳定匹配；获得这样的匹配后，其余鲁棒稳定匹配便可通过应用旋转来生成。

对于$(0,n)$情形——当偏好变化仅限于市场一侧时——可以对文献中已有的算法进行调整，以计算工人最优和雇主最优的稳健稳定匹配。我们在附录B中讨论这些调整。然而，这些方法无法扩展到$(1,n)$情形，因为市场双方都涉及其中。针对这种情形，我们基于 Gale 和 Shapley 的延迟接受（DA）算法设计了新的算法 [GS62]。

延迟接受算法（算法 4）通过迭代进行。每次迭代中，执行以下步骤：(a) 提议方（例如工人）向其最偏好的、尚未拒绝自己的企业提出提议。(b) 每家企业暂时接受该轮收到的提议中自己最偏好的一个，并拒绝其余提议。(c) 每位工人从自己的偏好列表中划去拒绝自己的企业。

延迟接受算法(A)：**输入：**稳定匹配实例A**输出：**完美匹配M每位工人都有一份按优先级从高到低排列的所有企业列表。1. 直到所有企业都收到提议，执行：(a) $\forall w \in W$，w 向其最优且尚未划去的企业提出提议。(b) $\forall f \in F$，f 暂时接受其最优提议并拒绝其余提议。(c) $\forall w \in W$，若w 被企业f拒绝，则从列表中划去f。2. 输出完美匹配M。

算法 4： 工人提出的延迟接受算法，由 [GS62](出自 [EIV23])

该过程持续进行，直到形成完美匹配，此时算法将其作为稳定匹配输出。关键思想是，每当发生拒绝时，对应的工人-企业配对永远不可能属于任何稳定匹配。我们修改该算法，以计算交集 $\mathcal{M}_A \cap \mathcal{M}_B$ 中 $(1,n)$ 情形下的工人最优和雇主最优稳定匹配。

对于 $(1,n)$ 情形，工人最优的 算法 5 在多个房间中同时运行延迟接受算法，其中每个房间的偏好列表对应一个独立实例。参与者按照各房间中的偏好行动：每次迭代中，工人提出提议，企业暂时接受最优提议并拒绝其余提议。不过，拒绝会跨所有房间生效。例如，如果某工人在房间 $\mathcal{R}_A$ 中被一家企业拒绝，他们就会将该企业从所有房间的列表中删除。这种同步机制确保，只要返回了完美匹配，该匹配就相对于所有实例都是稳定的。主要的技术难点在于，算法可能在不同房间中产生不同的完美匹配。引理 25表明这种情况不会发生，这一结论源于交集 $\mathcal{M}_A \cap \mathcal{M}_B$ 在两个格中具有相同的偏序。

雇主最优算法 6以类似方式运行：由企业提出申请，工人暂时接受。一个关键区别在于，工人会发出抢先拒绝：在每个房间中，工人会拒绝所有在其当前暂定匹配中排名靠后的企业——包括尚未提出申请的企业——这些判断基于他们的本地偏好列表。

这确保最终得到的完美匹配在所有房间中都完全相同。

定理 23。 设 A 和 B 是两个属于 $(1,n)$ 型的稳定匹配实例。那么 算法 *5* 和算法 *6* 分别找出工人最优和企业最优鲁棒稳定匹配，或者正确报告不存在此类匹配。

WorkerOptimal(A, B)：输入：定义在代理人 W ∪F 上的稳定匹配实例 A 和 B。输出：完美匹配 M，或 ⊠ （不存在鲁棒稳定匹配时）。假设有两个房间， $\mathcal{R}_A$ 和 $\mathcal{R}_B$ 分别对应实例 A 和 B。每个工人都有一个列表，初始化为所有企业；他们在提议时查看该列表。1. 在任意房间中都没有拒绝，或某个工人被所有企业拒绝之前，重复执行：(a) ∀I ∈ {A, B}、∀w ∈ W：w 在房间 $\mathcal{R}_I$中向其在 I 中尚未划去的最优企业提出提议。(b) ∀I ∈ {A, B}、∀f ∈ F：f 在房间 $\mathcal{R}_I$中暂时接受其在 I 中最优的提议，并拒绝其余提议。(c) ∀w ∈W：如果 w 在任意房间中被企业 f 拒绝，则将 f 从其列表中划去。2. (a) 如果某个工人被所有企业拒绝，则输出 ⊠。(b) 否则，这些接受关系在每个房间中定义一个完美匹配。如果所有房间中的匹配都相同，则输出该完美匹配 (M)。

算法 5： 寻找工人最优稳定匹配的算法。注意，提议者（工人）在所有房间中共用一个列表（或集合）。他们可以在每个房间向不同的企业提出申请，但会根据任何房间的拒绝同步更新列表。

定理 23 的证明 分为两部分，分别见 第 4.3.1 节 和 第 4.3.2 节。

4.3.1 算法正确性证明 算法5

我们分析该算法在（1,n） 情形下的工人提议版本，其中恰有一名工人在实例A 与B之间改变偏好。该算法在一组适当定义的实例上运行延迟接受算法，并在 $\mathcal{M}_A \cap \mathcal{M}_B$中返回工人最优鲁棒稳定匹配（如果存在）。令$w_1$ 表示偏好列表在A 与B之间不同的唯一工人。

引理 25。 算法 *5* 总会终止，并返回完美匹配或失败符号 ⊠。

证明。 该算法总会终止，因为在最后一次迭代之前，每次迭代至少会发生一次拒绝。

假设算法终止时得到两个不同的完美匹配， μ_A 在房间A 中， μ_B在房间B中。那么必定存在一名工人 $w \neq w_1$ ，其在A 和B 中的偏好相同，但在 μ_A 和 μ_B中的匹配对象不同。这样的工人必定存在，因为只有w_1 改变了偏好，而这些匹配都是完美匹配。一般性不失，假设 $\mu_A(w) >^{A,w}{}^{B} \mu_B(w)$。由于$w$ 在A 和B中的偏好列表相同，因此w 必定在房间B中向 $\mu_A(w)$ 提出过提议，遭到拒绝，并

FirmOptimal(A, B)：输入：参与者为 W ∪F 的稳定匹配实例 A 和 B。输出：完美匹配 M，或 ⊠（不存在鲁棒稳定匹配时）。假设有两个房间，$\mathcal{R}_A$ 和 $\mathcal{R}_B$，分别对应实例 A 和 B。每家企业都有一个列表，初始时包含所有工人；企业提议时会查看该列表。1. 直到任一房间中都没有发生拒绝，或某家企业被所有工人拒绝为止，执行：(a) ∀I ∈ {A, B}, ∀f ∈ F：f 向 $\mathcal{R}_I$中尚未划掉且在 I 中排名最高的工人提出提议。(b) ∀I ∈ {A, B}, ∀w ∈ W：w 在房间 $\mathcal{R}_I$ 中暂时接受 I 中自己最偏好的提议（称为 f Iw），并拒绝其余提议。(c) ∀I ∈ {A, B}, ∀w ∈ W：w 向所有尚未拒绝、且排名低于其当前选择的企业 f′ 发送先发拒绝，即 f Iw $\succeq_w^I$ f′。(d) ∀f ∈ F：如果 f 在某个房间被工人 w 拒绝，就从其列表中划掉 w。2. (a) 如果某家企业被所有工人拒绝，则输出 ⊠。(b) 否则，接受情况会在每个房间中形成完美匹配。如果所有房间中的匹配都相同，则输出该完美匹配(M)。

算法 6：寻找公司最优稳定匹配的算法。注意，在步骤 1(c) 中，工人可能会向尚未向其提出过申请的企业发送拒绝。

他们已将其从名单中划掉。然而，根据算法，w 也应将 $\mu_A(w)$从其在房间 A中的名单中划掉，因此 μ_A 不可能是在房间 A中得到的完美匹配。 □

引理 26。如果企业 f 在任一房间拒绝了 w，那么在 $\mathcal{M}_A \cap \mathcal{M}_B$ 的任何匹配中，w 和 f 都不能匹配 .

证明。我们使用单个实例的延迟接受算法的性质，A 该性质指出，如果企业 f 拒绝工人 w，那么 (w, f) 在相对于A [GS62]的任何稳定匹配中都不会匹配。

不失一般性，假设 f 在房间 A中拒绝了 w 。那么 (w, f) 在 $\mathcal{M}_A$ 的任何稳定匹配中都不会匹配，因此它们在 $\mathcal{M}_A \cap \mathcal{M}_B$ 的任何稳定匹配中也都不会匹配。

□

引理 27. 如果 A 和 B 存在鲁棒稳定匹配，那么 算法 5 就能找到一个。

证明. 假设存在这样的稳定匹配，但算法终止时输出⊠。这种情况只会发生在某个工人被所有企业拒绝时。然而，根据 引理 26，这意味着该工人在任何稳定匹配中都没有可行伴侣，这就产生了矛盾。

□

引理 28. 如果 算法 5 返回一个匹配，那么它在 A 和 B 下都必须是稳定的。

证明。 设 μ 是返回的完美匹配。不失一般性，假设相对于 A，存在阻塞对 (w, f)。也就是说，$f >_w^A \mu(w)$，且 $w >_f^A \mu(f)$。那么，w必定在算法的某次迭代中向f 提出过申请，并在房间A 中被拒绝，因为某位w' 工人受到f的青睐，胜过w；最终匹配为 $\mu(f)$。由于企业只会随着算法推进而获得更好的

匹配，因此必然有 $\mu(f) >_f w$，所以(w, f) 不是阻塞对。

□

Le引理 29。 如果 算法 *5* 返回一个匹配，则它必定是工人最优匹配 .

证明。 否则，某个工人$w \neq w_1$ 在工人最优匹配中会有更好的配偶。但这样一来，该工人就会在每个房间中都向那位配偶提出申请，并至少在其中一个房间遭到拒绝，从而导致矛盾。

□

4.3.2 算法 6正确性证明

现在，我们分析该算法在$(1,n)$ 情形下的企业提案版本。在这种情形中，实例A 和B之间恰好有一名工人的偏好发生变化。该算法在一组经过适当定义的实例上运行延迟接受算法，并在存在企业最优鲁棒稳定匹配时返回该匹配（ $\mathcal{M}_A \cap \mathcal{M}_B$）。令$w_1$ 表示在A 和B之间偏好列表不同的唯一工人。

引理30。 算法 算法 *6* 会终止，并且总是返回一个完美匹配，或者 ☒ .

*Pro*明。 该算法会终止，因为每次迭代至少会发生一次拒绝。

假设算法终止时，房间 A 中的 μ_A 与房间 B 中的 μ_B 是不同的完美匹配。那么存在某个工人 w $\neq$ w_1 ，他在 A 和 B 中的偏好列表相同，但匹配对象不同：μ_A(w) 和 μ_B(w)。不失一般性，假设 μ_A(w) $>_{w}^{A,B}$ μ_B(w)。那么，在某次迭代中，w 必定收到过来自 μ_A(w) 的提议，因为工人只有收到企业的提议，才能被暂时匹配。工人会拒绝在所有房间中都不如当前对象受偏好的企业，因此 w 必定也在房间 R_A 中拒绝了 μ_B(w)，并在房间 R_B 中拒绝了它。因此， μ_B 不可能是房间 R_B 中的最终完美匹配——这就产生了矛盾。 □

引理 31. 如果企业 f 在任一房间中被工人 w 拒绝，那么在 $\mathcal{M}_A \cap \mathcal{M}_B$ 的任何强稳定匹配中，w 和 f 都不能匹配。

证明。 我们利用单个实例 A的延迟接受算法的性质，该性质指出，如果企业 f 被工人 w拒绝，那么 (w, f) 在关于 A [GS62]中不会匹配。

我们还注意到一个额外事实：工人一旦拒绝某家企业，从那时起，他们只会与自己严格更偏好的合作方匹配。这等同于预先拒绝所有排名低于f的企业——我们将在此算法中沿用这一思路。

不失一般性，假设w 在A房间中拒绝了f。那么，在 $\mathcal{M}_A$的任何稳定匹配中，(w, f) 都不会匹配，因此在$\mathcal{M}_A \cap \mathcal{M}_B$的任何稳定匹配中，他们也不可能匹配。

□

Lemma 32. 如果 A 和 B 存在鲁棒稳定匹配，那么 算法 *6* 会找到一个。

证明。 假设存在这样的稳定匹配，但算法终止时输出⊠。这只会在某个企业被所有工人拒绝时发生。然而，根据 引理 31，这意味着该企业在任何稳定匹配中都没有可行伴侣，这就产生了矛盾。□

引理 33。 如果 算法 *6* 返回一个匹配，那么该匹配在 A 和 B 下都必须是稳定的。

证明。 假设算法返回一个完美匹配 μ，并且不失一般性地假设存在一个相对于 A 的阻塞对 (w, f)。也就是说，$w >_f^A \mu(f)$，且 $f >_w^A \mu(w)$。那么，f 必定在算法的某次迭代中向 w 提出过申请，而后在房间 R_A 中被某个企业拒绝；w 在 A 下更喜欢该企业。最终，w 与 $\mu(w)$ 配对，这是它最喜欢的。因此，$\mu(w) >_w^A f$，这与 (w, f) 是阻塞对的假设矛盾。□

引理 34。 如果 算法 *6* 返回一个匹配，那么它必须是企业最优匹配。

证明。 否则，某个企业 f 在企业最优匹配中会有一个更好的配偶。但这样一来，该企业就会在所有房间中都向该配偶提出申请，并在至少一个房间中遭到拒绝，从而导致矛盾。□

这些结果共同证明了算法 5 和6 分别求得工人最优和企业最优的鲁棒稳定匹配，从而完成了定理 23的证明。

4.4 多面体的整性

在第 3.6 节中，我们介绍了一种线性规划形式，其可行域为稳定匹配多面体，且其顶点均为整数的。现在，我们将这一框架扩展到稳健稳定匹配，通过定义一个恰当的线性规划，使其可行域涵盖在多个实例下均稳定的匹配。我们的目标是精确刻画稳健稳定匹配多面体何时具有整性，因为整性能够支持高效的基于线性规划的算法来计算稳健稳定匹配。

我们证明，当 (A, B) 的类型为 $(n, 1)$ 或 $(1, n)$ 时，稳健稳定匹配多面体是整数的，进而可在这些情形下通过高效算法计算稳健匹配。
鲁棒稳定匹配实例 (A, B) 的线性规划如下所示。前两个约束和最后一个约束确保任意可行解 x 都是分数完美匹配，而中间两个约束则分别防止实例 A 和 B 中出现阻塞对。

$$
\begin{aligned}
\text{maximize} \quad & 0 \\
\text{subject to} \quad & \sum_{w} x_{wf} = 1 && \forall f \in F, \\
& \sum_{f} x_{wf} = 1 && \forall w \in W, \\
& \sum_{f >_w^A f'} x_{wf'} - \sum_{w' >_f^A w} x_{w'f} \leq 0 && \forall w \in W,\ \forall f \in F, \\
& \sum_{f >_w^B f'} x_{wf'} - \sum_{w' >_f^B w} x_{w'f} \leq 0 && \forall w \in W,\ \forall f \in F, \\
& x_{wf} \geq 0 && \forall w \in W,\ \forall f \in F.
\end{aligned} \tag{2}
$$

线性规划 (2) 的一个解对应于一个在 A 和 B下都稳定的分数匹配。与单实例情形（见 第 3.6 节）类似，令 $\theta \in [0,1]$ 均匀随机选取。分别使用实例 A 和 B中的偏好序执行基于区间的舍入过程，构造两个整数完美匹配 μ_θ^A和 μ_θ^B。

引理35。 对于每个 $\theta \in [0,1]$，匹配 μ_θ^A 和 μ_θ^B 都相同，即 $\mu_\theta^A = \mu_\theta^B$。

证明。 μ_θ^A 和 μ_θ^B 是完美匹配，并且我们知道，在 A 和 B 中，$n-1$ 名工人w_1,.....w_{n-1} 的偏好相同。这意味着在这两个匹配中，这些 $n-1$ 名工人分别与 $n-1$ 家不同的企业配对。由于它们是完美匹配，故剩下的 n^{th} 名工人必须与剩余的企业配对。因此，这两个匹配相同。 □

定理24。 如果 (A,B) 属于 $(1,n)$ 类型，则 μ_θ^A 在 A 和 B 下均稳定，并且稳健分数稳定匹配多面体具有整数最优顶点。

证明。 根据 引理35， μ_θ^A和 μ_θ^B 相同。因此，由于它是整数的，并且是线性规划（2）的一个解，它在 A 和 B 下均稳定。这意味着，所有分数稳健稳定匹配的集合都可以写成整数解的凸组合，这与 定理3类似。这证明了该多面体是整数的。 □

因此，在 $(n,1)$ 和 $(1,n)$ 情形下，线性规划 (2) 提供了一种高效的方法来计算鲁棒稳定匹配。然而，这种方法不适用于 (2,2) 实例——引理 35 的证明无法推广；事实上，多面体甚至可能不是整数的，见图11中的例子。

定理25。 如果$(A,\ B)$ 属于$(2,2)$ 类型，那么鲁棒稳定匹配多面体并非总是整数的。

证明。 考虑图11所示的例子，其中实例 A 和 B 定义在4名工人 $\{1,2,3,4\}$ 和4家企业 $\{a、b、c、d\}$之上。只有工人1和2以及企业 a 和 b 的偏好从 A 变为 B。

实例 A 恰好有两个稳定匹配：

$$M_A^1 = \{(1,a),(2,b),(3,c),(4,d)\}, \quad M_A^2 = \{(1,b),(2,a),(3,d),(4,c)\}.$$

实例 B 也恰好有两个稳定匹配：

$$M_B^1 = \{(1,b),(2,a),(3,c),(4,d)\}, \quad M_B^2 = \{(1,a),(2,b),(3,d),(4,c)\}.$$

因此，对于这对A 和B，不存在鲁棒稳定匹配。

现在，考虑分数匹配 $M = \frac{1}{2}(M_A^1 + M_A^2) = \frac{1}{2}(M_B^1 + M_B^2)$。由于 M 是在 A 和 B下稳定匹配的凸组合，因此它是线性规划（2）的一个可行解。此外，M 是该实例中该线性规划的唯一解。

因此，鲁棒稳定匹配多面体是一个单点分数点，这表明对于一般的(2,2) 实例，该多面体并非整数的。 □

1	a	c	b	d
2	b	a	c	d
3	c	d	a	b
4	d	c	a	b

在A中的工人偏好

a	2	1	3	4
b	1	2	3	4
c	4	1	3	2
d	3	1	4	2

在A中的企业偏好

1	b	d	a	c
2	a	b	c	d
3	c	d	a	b
4	d	c	a	b

在B中的工人偏好

a	1	2	3	4
b	2	1	3	4
c	4	1	3	2
d	3	1	4	2

B中的企业偏好

图 11：此(2, 2) 实例的稳健稳定匹配多面体不是整数的 .

本节的所有结果都可以推广到两个以上的实例。设$(A_1, A_2, \ldots, A_k)$ 是一个(1,n)稳健稳定匹配实例，即在所有实例中，有$n-1$ 名工人的偏好配置都相同。那么：

定理 26。 对于一个$(1,n)$ 稳健稳定匹配实例$(A_1, A_2, \ldots, A_k)$，其中有 $k \geq 2$名工人在各实例中的偏好发生变化，集合 $\mathcal{L}' = \mathcal{M}_{A_1} \cap \mathcal{M}_{A_2} \cap \ldots \cap \mathcal{M}_{A_k}$ 构成每个格 $\mathcal{L}_{A_i}$的子格。线性规划公式可用于求出 $\mathcal{L}'$中的匹配，并且可以在多项式时间内计算出工人最优和企业最优鲁棒稳定匹配。此外，如果这些实例也是(0,n)实例，则生成 $\mathcal{L}'$ 的 *Birkhoff* 偏序可以高效计算，并且其中的匹配可以用多项式延迟枚举。

证明。 $\mathcal{L}'$ 是每个 $\mathcal{L}_{A_i}$ 的子格，这可由定理 19推得。线性规划方法可用于寻找交集中的稳定匹配：新的表述将包含 k 组阻挡对条件——每个实例 A_i对应一组——以确保不存在工人—企业对 (w, f) 在该实例下对 x 构成阻塞对。 算法 5 和 6 可用于寻找工人最优匹配和企业最优匹配。为每个实例定义一个房间，并根据相应实例的偏好在每个房间中运行延迟接受算法。如果相同的完美

匹配 M 在每个房间中都被得到，则由算法输出该匹配，并且它相对于所有实例都是稳定的。

当实例为 $(0,n)$ 时，通过识别每个实例 $\mathcal{L}'$ 中表示 $\mathcal{M}_{A_1} \cap \mathcal{M}_{A_i}$ 的边集 A_i，即可求得 Birkhoff 偏序；根据引理 22，这些边集的并集定义了 $\mathcal{L}'$。这提供了一种高效枚举匹配的方法。

□

4.5 一般情形下的稳健稳定匹配算法

第 4.2 节至 4.4 节证明了，对于 $(1,n)$ 实例，可以在多项式时间内找到稳健稳定匹配；而 [MO19] 则表明，当(A,B) 属于(n,n)类型时，该问题变为 NP 困难。现在我们考虑完全一般的设定，在这种设定下，实例A 与B 在参与者子集$S \subseteq \mathcal{W} \cup \mathcal{F}$的偏好上可以存在任意差异，其中 $|S \cap \mathcal{W}| = p$ 且 $|S \cap \mathcal{F}| = q$。

算法 7 可在存在鲁棒稳定匹配时找到一个。根据定理 27，该算法的运行时间为$O(n^{p+q+2})$ ，且可对其进行修改，以相同的渐近延迟枚举所有鲁棒稳定匹配。这得到了一个 XP 时间算法，其参数为两个实例中偏好不同的参与者总数。

直观来说，枚举 S 中参与者所有可能的伴侣分配，最多有 $O(n^{p+q})$ 种可能性。每一种分配都定义了 S 中参与者的一个部分匹配 M，算法会尝试确定 M 能否扩展为一个在 A 和 B 下都稳定的完整匹配。首先，算法会验证 $T = S \cup M(S)$ 内部不存在阻塞对。若不存在阻塞对，则在剩余参与者 $U = (W \cup F) \setminus T$ 上定义截断实例 X。在 X 中，缩短 U 中参与者的偏好列表，以排除涉及 T 中参与者的潜在阻塞对。然后在 X 上计算一个稳定匹配 M'。如果 M' 是 U 上的完美匹配，则可以证明，组合匹配 $M^\star = M \cup M'$ 是 A 和 B 中全体参与者之间的稳定匹配。

RobustStableMatching(A, B)：输入：参与者为 $W \cup F$ 的稳定匹配实例 A 和 B。输出：完美匹配 $M^\star$ 或 ⊠（若不存在鲁棒稳定匹配）。$S \subseteq W \cup F$ 是在 A 和 B 中偏好不同的参与者。

1. 对于 S 中每一种能够构成有效部分匹配的伴侣分配：(a) 令 M 为由此得到的部分匹配，并令 $T := S \cup M(S)$。(b) 如果存在 $w \in T \cap W$ 和 $f \in T \cap F$，使得 (w, f) 在 A 或 B 中相对于 M 构成阻塞对，则继续下一次迭代。(c) 移除 T 中所有参与者，得到定义在 $U := (W \cup F) \setminus T$ 上的实例 X；(d) 对每个 $a \in T$ 和 $b \in U$，若在 A 或 B 中 $b \succ_a M(a)$，则在 X 中将 b 的偏好列表截断至 a（即移除 a 以及排名低于 a 的所有参与者）。(e) 在 X 中求稳定匹配 M'。(f) 若 M' 是 U 上的完美匹配，则返回 $M^\star := M \cup M'$。

2. **返回** ⊠。

算法 7： 一种用于为较小的 (p,q) 寻找鲁棒稳定匹配的 XP 时间算法。

定理 27. 对于类型为 $(1,n)$ 的实例 A 和 B，算法 7 可在 $O\left(n^{p+q+2}\right)$ 时间内找到一个鲁棒稳定匹配，或正确报告不存在此类匹配。并且，鲁棒稳定匹配集合可通过 $O\left(n^{p+q+2}\right)$ 延迟枚举。

证明。 设 A 和B 是定义在工人W 和企业F上的两个实例。设S 是A 和B中偏好不同的参与者集合。在算法 7的一次固定迭代中，令M 表示步骤 1(a) 中选取的S 上的部分匹配，令$T := S \cup M(S)$，并令$U := (W \cup F) \setminus T$。在步骤 1(c)–(d) 中得到定义在$U$ 上的截断实例X 。由于所有偏好不同的参与者都已被移除，U 中参与者在A 和B中的偏好完全相同。

引理 36。 如果存在鲁棒稳定匹配 $M^\star \in M_A \cap$ ，则M_B，那么 算法 7 会返回一个完美匹配。

证明。 考虑选取部分匹配$M := M^\star|_S$ 的那次迭代，该匹配是在步骤 1(a) 中选取的。令$M' := M^\star|_U$。

首先注意，在任一实例A或B下，w和$f \in T$ 都不存在阻塞对(w, f) ，否则$M^\star$ 在相应实例中就不会稳定。因此，算法在此次迭代中不会在步骤 1(b) 中止。

接下来，注意到 M' 在 X 中也稳定。否则，M' 在 X 中的任何阻塞对，都对应于 A 和 B中的阻塞对，这与 $M^\star$的稳定性矛盾。此外，由于 $M^\star$ 是 $W \cup F$上的完美匹配，M' 必定是 U上的完美匹配。

由于 M' 是 X中的完美稳定匹配，乡村医院定理意味着 X 中的每个稳定匹配都是完美匹配。因此，算法必须在本次迭代中输出一个完美匹配。

□

引理 37。 如果 算法 7 返回一个完美匹配 $M^\star$，那么 $M^\star \in \mathcal{M}_A \cap \mathcal{M}_B$。

证明。 令 $M := M^\star|_S$ 为 S上的部分匹配，并令 $M' := M^\star|_U$ 为算法返回 $M^\star$的那次迭代中得到的 X 中的完美稳定匹配。固定任意一个实例$I \in \{A, B\}$。在 I 下，任何潜在的阻塞对 (a, b) 必属于以下某种情况：

1. 两个代理人都在 T 中。这种情况会被步骤 1(b) 排除：如果存在任何这样的阻塞对在任一实例中阻塞 M，该步骤就会丢弃此次迭代。2. 一个代理人在 T 中，另一个在 U 中。不失一般性，设 a $\in$ T，b $\in$ U。若 (a, b) 在 I 下是阻塞对，则 b $\succ_a^I$ M(a)，且 a $\succ_b^I$ M′(b)。然而，由于 b $\succ_a^I$ M(a)，在步骤 1(d) 中，实例 X 会从 b 的偏好列表中移除代理人 a 以及所有排名低于 a 的代理人。因此，b 不可能与排名低于 a 的代理人匹配，这与 a $\succ_b^I$ M′(b) 矛盾。3. 两个代理人都在 U 中。若 (a, b) 是阻塞对，则 b $\succ_a^I$ M′(a)，且 a $\succ_b^I$ M′(b)。注意，我们只截去每个代理人偏好列表的末尾。特别是，如果 x $\in$ U 与 M′(x) 匹配，那么在实例 I 中，x 比 M′(x) 更偏好的每个代理人仍必须出现在实例 X 中 x 的偏好列表里。因此，(a, b) 在实例 X 下也必定是阻塞对，这与步骤 1(e) 返回的 M′ 为稳定匹配这一事实矛盾。

因此，$M^\star$ 在 I 中不存在阻塞对；又因为 $I \in \{A、B\}$ 是任意的，所以 $M^\star \in \mathcal{M}_A \cap \mathcal{M}_B$。 □

引理 38. 算法 7 可在 $O\left(n^{p+q+2}\right)$时间内判定是否存在鲁棒稳定匹配。

证明。 我们分别估算每次迭代的工作量和迭代总次数：

1. 迭代次数。 步骤 1 遍历针对 $|S| = p + q$参与者的所有单射伴侣指派，因此有$O(n^{p+q})$ 种可能。
2. 每次迭代的运行时间。 步骤 1(b) 检查T内部是否存在阻塞对；步骤 1(c)–(d) 构造截断实例 X；步骤 1(f) 在X中求出一个稳定匹配；步骤 1(g) 验证M' 是否是U上的完美匹配。使用 Deferred Acceptance 算法$O(n^2)$ 时间实现中的标准数据结构，可以将上述每个步骤的运行时间控制在$O(n^2)$时间内。

因此，总运行时间为$O(n^{p+q+2})$。正确性由 引理 36 和 37得出。 □

引理 39。 实例 A 和 B 的鲁棒稳定匹配集合可以以 $O(n^{p+q+2})$的延迟进行枚举。

证明。 算法 7可以修改为如下方式，以枚举所有鲁棒稳定匹配：不再于步骤 1(f) 中返回单个稳定匹配，而是使用一种已知的、能够枚举其稳定匹配格结构的算法，枚举截断实例 X 的所有稳定匹配，每个匹配之间的延迟为$O(n^2)$ 。

由于步骤 1 的外层循环遍历 $O(n^{p+q})$ 个在 S 上的部分匹配选择，并且每次迭代执行$O(n^2)$ 的工作（或在步骤 1(b) 处提前终止），因此任意两个输出之间的延迟至多为$O(n^{p+q+2})$。

因此，所有鲁棒稳定匹配都可以以$O(n^{p+q+2})$ 的延迟进行枚举。 □

综上，前述引理证明了定理 27。

□

推论 3。 如果有常数数量的参与者改变其偏好，则可以在多项式时间内判定是否存在鲁棒稳定匹配，并找到一个鲁棒稳定匹配。所有此类匹配都可以以多项式延迟进行枚举。

定理 27意味着推论 3成立，因为当偏好发生变化的参与者数量为常数时，运行时间是多项式的。鲁棒稳定匹配的判定和枚举问题都可以高效求解。这表明，鲁棒稳定匹配的集合可以表示为$O(n^{p+q})$ 个稳定匹配格的并集，每个格对应于S中参与者的一种一致配偶分配。

然而，定义这些格的偏序可能存在显著差异。目前尚不清楚，这种分解是否能为计算 Birkhoff 偏序提供一种高效方法，即使是在$(1, 1)$ 情形下也是如此；根据推论 2，在$(1, n)$ 情形下同样如此。这些观察自然引出了以下开放问题。

备注 28. 开放问题。 计算$(1, 1)$ 情形下的 Birkhoff 偏序，以及由此扩展到$(1,n)$ 情形下的 Birkhoff 偏序，仍是一个开放问题。

备注 29. 开放问题。给定一个(p,q) 鲁棒稳定匹配实例(A, B)，其中 $2 \leq p$、$q \leq n$，$p + q < 2n$，且$p + q = \omega(1)$，确定是否存在鲁棒稳定匹配——若存在则计算出一个——是否属于 $\mathcal{P}$，仍是一个悬而未决的问题。

5 致谢

本研究部分由美国国家科学基金会（NSF）通过 CCF-2230414 项目资助。我们感谢 Simon Murray 分享他的见解，这些见解促成了算法 7，并感谢所有匿名审稿人提出的宝贵意见。

参考文献

[ABF+17] H. Aziz, P. Biro, T. Fleiner, S. Gaspers, R. de Haan, N. Mattei 和 B. Rastegari。具有不确定成对偏好的稳定匹配。载于自主智能体与多智能体系统国际联合会议，第 344–352 页，2017年。[ABG+16] H. Aziz, P. Biro, S. Gaspers, R. de Haan, N. Mattei 和 B. Rastegari。具有不确定线性偏好的稳定匹配。载于算法博弈论国际研讨会，第 195–206 页，2016年。[APR09] Atila Abdulkadiro˘glu、Parag A Pathak 和 Alvin E Roth。含无差异的匹配中策略无关性与效率的权衡：重新设计纽约市高中匹配。《美国经济评论》，99(5):1954–78，2009年。[APRS05] Atila Abdulkadiro ˘glu、Parag A Pathak、Alvin E Roth 和 Tayfun S¨onmez。波士顿公立学校匹配。《美国经济评论》，95(2):368–371，2005年。[B+37] Garrett Birkhoff 等。集合环。《杜克数学期刊》，3(3):443–454，1937年。[BBHN21] Niclas Boehmer、Robert Bredereck、Klaus Heeger 和 Rolf Niedermeier。稳定婚姻中的贿赂与操控。《人工智能研究期刊》，71:993–1048，2021年。[BCK+20] Robert Bredereck、Jiehua Chen、Dusan Knop、Junjie Luo 和 Rolf Niedermeier。使稳定匹配适应不断变化的偏好。载于第三十四届 *AAAI* 人工智能会议（*AAAI 2020*）、第三十二届人工智能创新应用会议（*IAAI 2020*）、第十届 *AAAI* 人工智能教育进展研讨会（*EAAI 2020*），美国纽约，*2020*年*2*月*7*日至*12*日，第 1830–1837 页。AAAI Press，2020年。[BCK24] Krist´of B´erczi、Gergely Cs´aji 和 Tam´as Kir´aly。操纵稳定婚姻与室友问题的结果。《博弈与经济行为》，147:407–428，2024年。[BFK+21] Robert Bredereck、Piotr Faliszewski、Andrzej Kaczmarczyk、Rolf Niedermeier、Piotr Skowron 和 Nimrod Talmon。多赢家投票规则的鲁棒性。《人工智能》，290:103403，2021年。

[BGS24] Martin Bullinger、Rohith Reddy Gangam 和 Parnian Shahkar。稳健热门匹配。载于第 23 届国际自主智能体与多智能体系统会议论文集（AAMAS '24），第 225–233 页，南卡罗来纳州里奇兰，2024年。国际自主智能体与多智能体系统基金会。[BHN22a] Niclas Boehmer、Klaus Heeger 和 Rolf Niedermeier。深入分析增量稳定匹配问题的（参数化）复杂性。载于 Stefan Szei-der、Robert Ganian 和 Alexandra Silva 编，第 47 届国际计算机科学数学基础研讨会（MFCS 2022），Leibniz 信息学国际论文集（LIPIcs）第 241 卷，第 21:1–21:15 页，德国达格施图尔，2022年。Schloss Dagstuhl – Leibniz-Zentrum f¨ur Informatik。[BHN22b] Niclas Boehmer、Klaus Heeger 和 Rolf Niedermeier。不断变化的双边匹配市场中最小侵入式稳定性保持的理论与实验。AAAI人工智能会议论文集，36(5):4851–4858，2022年6月。[CNS18] Jiehua Chen、Rolf Niedermeier 和 Piotr Skowron。具有多模态偏好的稳定婚姻。载于2018年 ACM经济学与计算会议论文集（EC '18），第 269–286 页，美国纽约，2018年。计算机协会。[Cs´a24] Gergely Cs´aji。具有不确定且多模态偏好的受欢迎匹配与支配匹配。载于 Kate Larson 编，第三十三届国际人工智能联合会议论文集（IJCAI-24），第 2740–2747 页。国际人工智能联合会议组织，2024年8月。主会场论文。[CSS21]Jiehua Chen、Piotr Skowron 和 Manuel Sorge。偏好下的匹配：稳定性强度与权衡。ACM经济学与计算汇刊，9(4)，2021年10月。[DF81] Lester E Dubins 和 David A Freedman。马基雅维利与盖尔-沙普利算法。《美国数学月刊》，88(7):485–494，1981年。[EFS09] Edith Elkind、Piotr Faliszewski 和 Arkadii Slinko。论若干投票规则的距离合理化。载于第 12 届理性与知识理论问题会议论文集（TARK '09），第 108–117 页，美国纽约，2009年。计算机协会。[EIV23] Federico Echenique、Nicole Immorlica 和 Vijay V. Vazirani 编。《在线与基于匹配的市场设计》。剑桥大学出版社，2023年。[FIM07] Tam´as Fleiner、Robert W. Irving 和 David F. Manlove。广义稳定婚姻与室友问题的高效算法。《理论计算机科学》，381(1):162–176，2007年。[FKJ16] Linda Farczadi、Georgiou Konstantinos 和 K¨onemann Jochen。具有一般偏好的稳定婚姻。《计算系统理论》，59(4):683–699，2016年。[FR16] Piotr Faliszewski 和 J¨org Rothe。投票中的操控与贿赂。载于 Felix Brandt、Vincent Conitzer、Ulle Endriss、J. Lang 和 Ariel D. Procaccia 编，《计算社会选择手册》第 7 章。剑桥大学出版社，2016年。

[G¨ar75] Peter G¨ardenfors. 配对：基于双边偏好的分配。《行为科学》，20(3):166–173，1975年。[GI89] Dan Gusfield 和 Robert W Irving。《稳定婚姻问题：结构与算法》。麻省理工学院出版社，1989年。[GMRV22] Rohith Reddy Gangam、Tung Mai、Nitya Raju 和 Vijay V. Vazirani。“相近”实例的稳定匹配格：结构与算法研究及其应用。载于第42届 IARCS 软件技术与理论计算机科学基础年会（FSTTCS 2022），2022年。[GMRV26] Rohith Reddy Gangam、Tung Mai、Nitya Raju 和 Vijay V. Vazirani。《稳定匹配：应对偏好变化》。载于第25届自主智能体与多智能体系统国际会议论文集（AAMAS 2026），2026年。即将出版。[GS62] David Gale 和 Lloyd S Shapley。大学招生与婚姻稳定性。《美国数学月刊》，69(1):9–15，1962年。[GSOS17] Begum Genc、Mohamed Siala、Barry O’Sullivan 和 Gilles Simonin。寻找稳定婚姻的鲁棒解。arXiv 预印本 arXiv:1705.09218，2017年。[GSSO17] Begum Genc、Mohamed Siala、Gilles Simonin 和 Barry O’Sullivan。关于鲁棒稳定婚姻的复杂性。载于组合优化与应用国际会议，第441–448页。施普林格，2017年。[IL86] Robert W Irving 和 Paul Leather。稳定婚姻计数的复杂性。《SIAM计算杂志》，15(3):655–667，1986年。[Irv85] Robert W Irving。一种求解“稳定室友”问题的高效算法。《算法杂志》，6(4):577–595，1985年。[Irv94] Robert W. Irving。稳定婚姻与无差异。《离散应用数学》，48(3):261–272，1994年。[Knu76] Donald Ervin Knuth。《稳定婚姻》。技术报告，1976年。[Knu97a] Donald Ervin Knuth。《稳定婚姻及其与其他组合问题的关系：算法数学分析导论》。美国数学学会，1997年。[Knu97b] Donald Ervin Knuth。《稳定婚姻及其与其他组合问题的关系：算法数学分析导论》，第10卷。美国数学学会，1997年。[KPG16] Adam Kunysz、Katarzyna E. Paluch 和 Pratik Ghosal。强稳定匹配的刻画。载于 Robert Krauthgamer 编，《第二十七届 ACM-SIAM 离散算法研讨会论文集》，SODA 2016，美国弗吉尼亚州阿灵顿，2016年1月10日至12日，第107–119页。SIAM，2016年。[Man02] David F. Manlove。带无差异偏好的稳定婚姻的结构。《离散应用数学》，122(1):167–181，2002年。[Man13] David Manlove。《偏好下匹配的算法学》。世界科学出版社，2013年。

[ML18] Vijay Menon 和 Kate Larson。部分信息下鲁棒且近似稳定的婚姻。载于 George Christodoulou 和 Tobias Harks 编，《Web and Internet Economics》，第341–355页，沙姆，2018年。Springer International Publishing。[MO19] Shuichi Miyazaki 和 Kazuya Okamoto。联合稳定匹配。《组合优化期刊》，38(2):646–665，2019年。[MV18] Tung Mai 和 Vijay V. Vazirani。寻找对输入错误具有鲁棒性的稳定匹配。载于欧洲算法研讨会，2018年。[Rot82] Alvin E Roth。匹配的经济学：稳定性与激励。《运筹学数学》，7(4):617–628，1982年。[Rot85] A. E. Roth。大学招生问题并不等同于婚姻问题。《经济理论杂志》，36(2):277–288，1985年。[Rot16] Alvin E. Roth。Al Roth 的博弈论、实验经济学与市场设计网页，2016年。http://stanford.edu/~alroth/alroth.html#MarketDesign。[RS12] Alvin E. Roth 和 Lloyd S. Shapley。《2012年诺贝尔经济学奖》，2012年。https: // www.nobelprize.org/nobel_prizes/economic-sciences/laureates/2012/。[Spi95] Boris Spieker。超稳定婚姻匹配构成一个分配格。《离散应用数学》，58(1):79–84，1995年。[Sta96] Richard Stanley。《计数组合学》，第1卷，Wadsworth and Brooks/Cole，太平洋丛林市，加利福尼亚州，1986年；第二次印刷，1996年。[SYE13] Dmitry Shiryaev、Lan Yu 和 Edith Elkind。关于具有稳健获胜者的选举。载于2013年自主智能体与多智能体系统国际会议论文集（AAMAS '13），第415–422页，南卡罗来纳州里奇兰，2013年。国际自主智能体与多智能体系统基金会。[TS98] Chung-Piaw Teo 和 Jay Sethuraman。分数稳定匹配的几何及其应用。《运筹学数学》，23(4):874–891，1998年。[Van89] John H. Vande Vate。线性规划带来婚姻幸福。《运筹学快报》，8(3):147–153，1989年。

A Pr利用稳定匹配格证明伯克霍夫定理 s

我们将在稳定匹配格的背景下证明定理 6；这是不失一般性的，因为稳定匹配格与有限分配格一样一般。在此背景下，偏序 Π 的真元素将是旋转，而元元素称为元旋转。令 $\mathcal{L} = L(\Pi)$为相应的稳定匹配格。

显然，只需证明：

- 给定一个压缩 Π'，$L(\Pi')$ 同构于 $\mathcal{L}$的一个子格。
- 任意子格 $\mathcal{L}'$ 都是某个压缩 Π'对应的$L(\Pi')$ 。

这两个证明分别见第A.1 节和A.2节。

A.1 $L(\Pi')$ 与$L(\Pi)$同构于一个子格

设I 是 Π' 的闭子集；显然I 是一个元旋转集合。定义 $\mathrm{rot}(I)$ 为I中所有元旋转的并集，即:

$$\mathrm{rot}(I) = \{\rho \in A : A \text{ is a meta-rotation in } I\}.$$

我们将把消去元旋转 $A$of Π' 的过程定义为按照与偏序 Π一致的顺序消去A 中的旋转。此外，消去 I 中的元旋转是指从格 $\mathcal{L}$ 中的稳定匹配M_0 开始，并按照与 Π'一致的顺序消去I 中的所有元旋转。注意，这等价于从 $\mathcal{L}$ 中的稳定匹配M_0 开始，并按照与偏序 Π一致的顺序消去 $\mathrm{rot}(I)$ 中的所有旋转。根据定义2，这是因为如果 Π 中存在旋转x和y ，使得x 属于元旋转X，y 属于元旋转Y ，且x 先于y，那么X 也必定先于Y。因此，如果消去 $\mathrm{rot}(I)$ 中的所有旋转得到匹配M_I，那么消去I 中的所有元旋转也会得到相同的匹配。

最后，为了证明本节标题中的论断，只需注意，若I 和J 是偏序 Π' 的两个真闭集，那么

$$\mathrm{rot}(I \cup J) = \mathrm{rot}(I) \cup \mathrm{rot}(J) \quad \text{and} \quad \mathrm{rot}(I \cap J) = \mathrm{rot}(I) \cap \mathrm{rot}(J).$$

由此可知，通过消去 Π' 的一个真闭集中的元旋转所得到的匹配集合，在交与并运算下封闭，因此构成 $\mathcal{L}$的一个子格。

A.2 子格 $\mathcal{L}'$ 由 Π的压缩 Π' 生成

我们将分阶段得到 Π 的压缩 Π' 。首先，我们说明如何划分 Π 的旋转集合，从而得到 Π'的元旋转。接着，我们找出这些元旋转之间的优先关系，从而得到 Π'。最后，我们证明$L(\Pi') = \mathcal{L}'$。

注意， $\mathcal{L}$ 可以用其哈斯图$H(\mathcal{L})$表示。$H(\mathcal{L})$ 的每条边都包含 Π中恰好一个（不一定唯一的）旋转。于是，根据引理 1，对于任意两个稳定匹配M_1,$M_2 \in \mathcal{L}$ ，若$M_1 \prec M_2$, ，则在$H(\mathcal{L})$ 中从M_1 到 M_2 的所有路径都包含相同的旋转集合。

定义 4. 对于 M_1,$M_2 \in \mathcal{L}'$，若 M_2 是 M_1 的 $\mathcal{L}'$-直接后继，当且仅当 $M_1 \prec M_2$ ，且不存在 $M \in \mathcal{L}'$ 使得 $M_1 \prec M \prec M_2$。设 $M_1 \prec \ldots \prec M_k$ 是 $\mathcal{L}'$ 中的一个匹配序列，并且对于所有 $1 \leq i \leq k-1$，M_{i+1} 都是 M_i 的 $\mathcal{L}'$-直接后继。那么，对于所有 $1 \leq i \leq k-1$，$H(\mathcal{L})$ 中任何一条从 M_1 到 M_k且包含 M_i的路径，都称为一条 $\mathcal{L}'$- 路径。

令 $\mathrm{M}_{0'}$ 和 $\mathrm{M}_{z'}$ 分别表示 $\mathcal{L}'$中的工人最优匹配和企业最优匹配。对于满足 $\mathrm{M}_1 \prec \mathrm{M}_2$, 的 M_1,$\mathrm{M}_2 \in \mathcal{L}'$ ，令 S_{M_1,M_2} 表示从 M_1 到 M_2的任意 $\mathcal{L}'$-路径上所包含的旋转集合。此外，令 $\mathrm{S}_{M_0,M_{0'}}$ 和 $\mathrm{S}_{M_{z'},M_z}$ 分别表示 $\mathrm{H}(\mathcal{L})$中从 M_0 到 $\mathrm{M}_{0'}$ 以及从 $\mathrm{M}_{z'}$ 到 M_z的任意路径上所包含的旋转集合。定义以下集合，其元素均为旋转集合。

$$\mathcal{S} = \{S_{M_i,M_j} \mid M_j \text{ is an } \mathcal{L}'\text{-direct successor of } M_i,$$

$$\text{对于每一对匹配 } M_i\text{、}M^j\text{，其中 } \mathcal{L}'\}\bigcup$$

$$\{S_{M_0,M_{0'}},\ S_{M_{z'},M_z}\}.$$

引理 40. $\mathcal{S}$ 是 Π的一个划分。

证明。首先，我们证明任意旋转都必须属于 $\mathcal{S}$ 中的某个元素。考虑 H($\mathcal{L}$) 中一条从 M_0 到 M_z 的路径 p，且 p 经由一条 $\mathcal{L}'$-路径从 $M_{0'}$ 到 $M_{z'}$。由于 p 是从 M_0 到 M_z 的路径，根据引理 1， Π 的所有旋转都包含在 p 上。因此，它们都会出现在 $\mathcal{S}$ 中的集合里。

接下来，假设存在两对 $(M_1,M_2)\ \neq (M_3,M_4)$，它们是 $\mathcal{L}'$-直接后继，使得$S_{M_1,M_2} \neq S_{M_3,M_4}$ 以及 $X = S_{M_1,M_2} \cap S_{M_3,M_4} \neq \emptyset$。从$M_0$ 到M_2 所消去的旋转集合为

$$S_{M_0,M_2} = S_{M_0,M_1} \cup S_{M_1,M_2}.$$

同理,

$$S_{M_0,M_4} = S_{M_0,M_3} \cup S_{M_3,M_4}.$$

因此,

$$S_{M_0,M_2\vee M_3} = S_{M_0,M_3} \cup S_{M_1,M_2} \cup S_{M_0,M_1}.$$

$$S_{M_0,M_1\vee M_4} = S_{M_0,M_3} \cup S_{M_3,M_4} \cup S_{M_0,M_1}.$$

令 $M = (M_2 \vee M_3) \wedge (M_1 \vee M_4)$，我们有

$$S_{M_0,M} = S_{M_0,M_3} \cup S_{M_0,M_1} \cup X.$$

因此,

$$S_{M_0,M\wedge M_2} = S_{M_0,M_1} \cup X.$$

由于 $X \subset S_{M_1,M_2}$ 和 $S_{M_1,M_2} \cap S_{M_0,M_1} = \emptyset$，$X \cap S_{M_0,M_1} = \emptyset$。因此,

$$S_{M_0,M_1} \subset S_{M_0,M\wedge M_2} \subset S_{M_0,M_2},$$

因此 M_2 不是 M_1,的 $\mathcal{L}'$-直接后继，从而导致矛盾。 □

我们分别用 A_s 和 A_t表示 $S_{M_0,M_{0'}}$ 和 $S_{M_{z'},M_z}$ 。 $\mathcal{S}$ 的元素将是 Π′的元旋转。接下来，我们需要定义这些元旋转之间的优先关系，以完成 Π′的构造。对于元旋转 $A \in \mathcal{S}$，$A \neq A_t$，定义 $\mathcal{L}'$的如下子集:

$$\mathcal{M}^A = \{M \in \mathcal{L}' \text{ such that } A \subseteq S_{M_0,M}\}.$$

Le引理 41. 对于每个元旋转 $A \in \mathcal{S}$，$A \neq A_t$, $\mathcal{M}^A$ 构成 $\mathcal{L}'$的一个子格 $\mathcal{L}^A$ 。

证明。取两个匹配 M_1,M_2 ，使得 S_{M_0,M_1} 和 S_{M_0,M_2} 都是 A 的超集。则 $S_{M_0,M_1\wedge M_2} = S_{M_0,M_1} \cap S_{M_0,M_2}$ 和 $S_{M_0,M_1\vee M_2} = S_{M_0,M_1} \cup S_{M_0,M_2}$ 也都是 A 的超集。 □

令M^A 为格 $\mathcal{L}^A$中的工人最优匹配。令p 为从$M_{0'}$ 到M^A的任意一条 $\mathcal{L}'$-路径，并令 $\mathrm{pre}(A)$ 为在p上出现在A 之前的元旋转集合。

引理 42。 集合pre(A) 与 p 无关。此外，在任何包含 A 的从 $M_{0'}$ 出发的 $\mathcal{L}'$-路径上，prepre(A) 中的每个元旋转都出现在 A 之前。

证明。 由于从 $M_{0'}$ 到 M^A 的所有路径都给出相同的旋转集合，因此从 $M_{0'}$ 到 M^A的所有 $\mathcal{L}'$-路径都给出相同的元旋转集合。此外，在从 $M_{0'}$到 M^A的任意 $\mathcal{L}'$-路径中，A 必定最后出现；否则，在 M^A之前会存在一个匹配 $\mathcal{L}^A$，从而导致矛盾。因此，pre(A) 与 p无关。

设 q 是从 $M_{0'}$ 出发且包含匹配 M′、M $\in \mathcal{L}'$ 的一条 $\mathcal{L}'$-路径，其中 M 是 M′的 $\mathcal{L}'$-直接后继。设 A 表示包含在边 (M′, M) 上的元旋转。假设存在一个元旋转 A′ $\in$ pre(A)，使得 A′ 在 q 中不会出现在 A 之前。那么 S_{M_0},$M^A \wedge$M $=S_{M_0}$,$M^A \cap S_{M_0}$,M 包含 A，但不包含 A′。因此，$M^A \wedge$ M 是 $\mathcal{L}^A$ 中先于 M^A的一个匹配，从而导致矛盾。于是，pre(A) 中的所有匹配都必须在所有此类路径 q 中出现在 A 之前。 □

最后，对于每个元旋转，在 pre(A) 中的所有元旋转与 A之间添加优先关系。同时，在 $\mathcal{S} - \{A_t\}$ 中的所有元旋转与 A_t之间添加优先关系。这样便完成了 Π′的构造。下文将证明 Π′ 确实是 Π的压缩，但首先我们需要证明该构造确实得到了一个有效的偏序集。

引理 43. Π′ 满足传递性和反对称性。

证明。首先证明 Π′ 满足传递性。设 A_1,A_2,A_3 是元旋转，且满足 $A_1 \prec A_2$ 和 $A_2 \prec A_3$。不妨设 $A_3 \neq A_t$。那么 $A_1 \in$ pre(A_2)，且 $A_2 \in$ pre(A_3)。由于 $A_1 \in$ pre(A_2)，S_{M_0},M^{A_2} 是 A_1的超集。根据引理 41，$M^{A_1} \prec M^{A_2}$。同理，$M^{A_2} \prec M^{A_3}$。因此 $M^{A_1} \prec M^{A_3}$，从而 $A_1 \in$pre(A_3)。

接下来我们证明 Π′ 满足反对称性。假设存在元旋转 A_1,A_2，使得 $A_1 \prec A_2$ 且 $A_2 \prec A_1$。显然 A_1,$A_2 \neq A_t$。由于 $A_1 \prec A_2$, $A_1 \in$ pre(A_2)。因此，S_{M_0},M^{A_2} 是 A_1 的超集。由此可得 $M^{A_1} \prec M^{A_2}$。应用类似的论证，我们得到 $M^{A_2} \prec M^{A_1}$。现在便得到矛盾，因为 A_1 和 A_2 是不同的元旋转。 □

引理 44. Π′ 是 Π 的压缩。

证明。 设 x、y 是 Π 中满足 $x \prec y$ 的旋转。设 X 为包含 x 的元旋转，Y 为包含 y 的元旋转。只需证明 $X \in$ pre(Y)。设 p 为一条从M_0 到 M^Y 的 $\mathcal{L}'$-路径。由于 $x \prec y$，<style id='x 必须出现在p 中y 之前。因此，X 也出现在p 中Y 之前。根据引理 42，$X \in$pre(Y)，证毕。 □

最后，接下来的两个引理证明了$L(\Pi') = \mathcal{L}'$。

引理 45. $L(\Pi')$ 中的任何匹配都必须属于 $\mathcal{L}'$。

证明。 对于 Π′ 中的任意真闭子集I，令M_I 为通过消去I中的元旋转所生成的匹配。令J 为 Π′ 中另一个真闭子集，满足$J = I \setminus \{A\}$，其中A是I中的极大元旋转。根据归纳法，M_J 是 $\mathcal{L}'$ 中的一个匹配。由于I 包含A，S_{M_0}，$M_I \supset A$。因此，$M^A \prec M_I$。由此可得$M_I = M_J \vee M^A \in \mathcal{L}'$。 □

引理 46。 $\mathcal{L}'$ 中的任意匹配都必须属于 $L(\Pi')$。

证明。 假设存在一个匹配 M 属于 $\mathcal{L}'$，使得 $M \notin L(\Pi')$。那么必然有 $S_{M_0,\ M}$ 无法划分为构成 Π 的闭子集的元旋转。现在分两种情况讨论。

首先，假设 $S_{M_0,\ M}$ 可以划分为元旋转，但它们并不构成 Π' 的闭子集。令 A 为一个元旋转，使得 $S_{M_0,\ M} \supset A$，并且存在 $B \prec A$ 使得 $S_{M_0,\ M} \not\supset B$。根据引理 41，$M \succ M^A$，因此 $S_{M_0,\ M}$ 是 $\mathrm{pre}(A)$ 中所有元旋转的超集，这就产生了矛盾。

接下来，假设 $S_{M_0 M}$ 无法在 Π' 中划分为元旋转。由于元旋转集合划分了 Π，因此存在一个元旋转 X，使得 $Y = X \cap \mathcal{S}_{M_0 M}$ 是 X 的非空子集。令 J 为 Π 中位于 X 之前的元旋转集合。

$(\mathrm{M}_J \vee \mathrm{M}) \wedge \mathrm{M}^X$ 是由 J $\cup$Y 中的元旋转生成的匹配。显然，J 是 Π' 中的一个闭子集。因此，$\mathrm{M}_J \in \mathrm{L}(\Pi')$。根据引理 45，$\mathrm{M}_J \in \mathcal{L}'$。由于 M、$\mathrm{M}^X \in \mathcal{L}'$，$(\mathrm{M}_J \vee \mathrm{M}) \wedge \mathrm{M}^X \in \mathcal{L}'$ 也成立。从 $\mathrm{H}(\mathcal{L})$ 中 M_J 到 $(\mathrm{M}_J \vee \mathrm{M}) \wedge \mathrm{M}^X$ 的路径上所包含的旋转集合恰好是 Y。因此，Y 不可能是任何元旋转的子集，这与 $\mathrm{Y} = \mathrm{X} \cap \mathcal{S}_{M_0}$、M 是 X 的非空子集这一事实相矛盾。 □

B 一方偏好改变时的算法

在此情形下，我们考虑工人的偏好保持不变而企业的偏好不同的实例，即$(0, n)$实例。根据对称性，$(n,0)$ 的情形与之等价。

定理4 表明，在 $\mathcal{M}_A \cap \mathcal{M}_B$ 中，稳定匹配的并与交在 $\mathcal{L}_A$ 和 $\mathcal{L}_B$ 的序下相同。这意味着工人对于交集中的匹配有着一致的偏好。这一结果的关键在于，工人在两个实例中具有相同的偏好，因此，单实例稳定匹配格中的许多论证都可以沿用。此外，集合 $\mathcal{M}_A \cap \mathcal{M}_B$ 在 $\mathcal{L}_A$ 和 $\mathcal{L}_B$ 中构成相同的子格，即偏序彼此一致；也就是说，$(\mathcal{M}_A \cap \mathcal{M}_B, \preceq_A) = (\mathcal{M}_A \cap \mathcal{M}_B, \preceq_B)$。这意味着 $\mathcal{M}_A \cap \mathcal{M}_B$ 中存在工人最优和企业最优稳定匹配，并表明延迟接受算法的某种变体或许能在此情形下奏效。

这引出了*复合实例*这一概念，它编码了两个输入实例中的代理偏好。我们在这个新实例上提出一种延迟接受算法的变体（算法 10），以寻找此类匹配。构造方法见 算法 8。

定义 5。 设 A 和 B 为$(0,n)$。定义复合实例 X 如下：

1. 对于每个工人 $w \in \mathcal{W}$，X 中的偏好列表与 A 和 B 中的相同。*2.* 对于每个企业 $f \in \mathcal{F}$，当且仅当 f 在 A 和 B 中都更偏好 w_i 而非 w^j 时，f 在 X 中更偏好工人 w_i 而非 w^j：

$$\forall f \in \mathcal{F}, \forall w_i, w_j \in \mathcal{W}, \quad w_i >_f^X w_j \iff (w_i >_f^A w_j \ and \ w_i >_f^B w_j).$$

在复合实例 X 中，工人的偏好列表是全序的，而企业的偏好列表可能是偏序。若企业 f 在 X 中并不严格偏好 w_i 而非 wj ，则称f 对两者无差异，记作$w_i \sim_f^X wj$。

给定一个匹配 M，若工人—公司对 (w, f) $\notin$ M 满足以下条件，则称其为相对于 M 的强阻塞对：w 更偏好 f，而非 M 中与其匹配的对象；并且 f 严格偏好 w，或对 w 与 M 中和其匹配的对象无差异。不存在强阻塞对的匹配称为在 X 下强稳定。令 $\mathcal{M}_X$ 表示在 X 下强稳定的匹配集合。有关强稳定匹配的更多信息，请参见 [Irv94, FIM07] 。

以下引理表明，复合实例在我们的语境中具有意义：

引理 47. *[FIM07]* 若 X 是 A 和 B 的复合实例，则匹配 M 在 X 下强稳定，当且仅当它在 A 和 B 下均稳定。也就是说， $\mathcal{M}_X = \mathcal{M}_A \cap \mathcal{M}_B$。

引理 *47* 的证明。 设 (w, f) 是在 X 下匹配 M 的强阻塞对。由于 w 在 A 和 B中的偏好相同，因此在两个实例中，w 都严格偏好 f 而非 $M(w)$ 。企业 f必须严格偏好 w 而非其在 M 中的配偶，或对二者无差异。无论哪种情况，f 在 A 或 B中至少有一个实例下严格偏好 w ，因此(w, f) 在该实例下构成阻塞对。因此，M 并非在 A 和 B 下均稳定。

反过来，如果（w，f）是A 或B下的阻塞对，那么它满足在X下构成强阻塞对的条件。因此，$\mathcal{M}_X = \mathcal{M}_A \cap \mathcal{M}_B$。 □

CompoundInstance(A，B)：**输入：** 工人一方偏好相同的稳定匹配实例A 和B 。**输出：** 实例X。1. $\forall w \in W$，X 中w的偏好与A 和B中的相同。2. $\forall f \in F$， $\forall w_i$，$wj \in W$，$w_i >_f^X wj$ 当且仅当$w_i >_f^A wj$ 且$w_i >_f^B wj$。3. 返回X。

算法 8： 用于构造复合实例的子程序。

B.1 工人最优和公司最优匹配

原始的延迟接受算法（算法 4）通过迭代运行。每次迭代分为三个步骤：(a) 提议方（例如工人）向尚未拒绝自己的最优企业提出提议；(b) 每家企业暂时接受本轮最优提议，并拒绝其他所有提议；(c) 所有工人划掉拒绝过自己的企业。循环继续，直到找到并输出完美匹配。关键思想是，一旦发生拒绝，该工人—企业对就不可能属于任何稳定匹配。

我们对该算法进行修改，以找出（0，n）情形下交集中的工人最优和企业最优稳定匹配，从而确保不会出现强阻塞对。

定理30。算法 *9*和算法 *10*若存在，则分别求出工人最优匹配和企业最优匹配。

复合实例的工人最优算法(X)：**输入：**复合稳定匹配实例X**输出：**完美匹配M 或 ⊠每位工人都有一份企业名单，初始时均未划去，并按优先级从高到低排列。1. 在所有工人都获得接受，或某位工人被所有企业拒绝之前，重复执行：(a) $\forall w \in W$：w 向名单上优先级最高且尚未划去的企业提议。(b) $\forall f \in F$：f 暂时接受其有史以来最优的提议，即其严格优于曾收到的所有其他提议的提议，并拒绝其余提议。(c) $\forall w \in W$：如果w 被企业f拒绝，就从名单上划去f。2. 如果所有工人都获得接受，则输出完美匹配（M）；否则输出 ⊠。

算法 9： [Irv94]使用复合实例寻找工人最优稳定匹配。

FirmOptimalForCompoundInstance(X)：**输入：** 复合稳定匹配实例 X**输出：** 完美匹配 M 或 ⊠每家企业都有一个涵盖所有工人的有向无环图，初始时未交叉，用以表示其偏序偏好。1. 直到所有企业都收到接受，或某家企业被所有工人拒绝，执行：(a) $\forall f \in F$：f 向最优的未交叉工人提出提议，即向所有未交叉工人w提出提议，且不存在未交叉的w' 满足$w' >_f^X w$。(b) $\forall w \in W$：w 暂时接受最优提议，并拒绝其余提议。(c) $\forall f \in F$：如果f 被工人w拒绝，则将w 从其有向无环图中划掉。2. 如果所有企业都收到接

p受，则输出p出该 p完美匹配 g (M)；否则输 p出 ⊠。

算法 10: [Irv94]使用复合实例寻找公司最优稳定匹配 .

中的工人最优算法 算法 9 ([Irv94] 的简化算法，与 [MO19、FKJ16、GMRV22]中的算法等价）与经典延迟接受算法的区别如下：企业仅当某项提议严格优于它们曾收到过的所有提议（包括之前轮次中的提议）时，才会接受该提议。这避免了涉及企业无差异的强阻塞对。请注意，如果企业对提出提议的工人无差异，它可能会在某一轮中拒绝所有提议；这是维持强稳定性所必需的。

中的公司最优算法算法 10(与 [Irv94]中的 SUPER 算法等价），企业使用部分偏好的有向无环图提出提议。每家企业向所有未交叉的工人 w 提出提议，条件是不存在未交叉的工人 w' 满足 $w' >_f^X w$。工人（具有完全偏好）暂时接受自己最喜欢的提议，并拒绝其余提议。如果工人拒绝某家企业，那么根据严格偏好，他们之后不可能与该企业构成阻塞对。如果多名工人暂时接受同一家企业的提议，其中一些人之后可能会拒绝该企业，此时算法要么继续运行，要么在不存在完美匹配时终止。

注意：当$A = B$时，X 两侧都具有完全偏好，这些算法便简化为标准延迟接受算法。

注 31。 ([FIM07]) 在一侧具有偏序偏好的实例Y 中的强稳定匹配构成一个格。由于每个偏序都有有限多个线性扩张， $\mathcal{M}_Y$可以表示为有限多个格的交集。然而，线性扩张的数量可能随 n 呈指数增长，因此这并不能提供一种高效的方式来计算伯克霍夫偏序。

B.2 算法 9 的正确性证明9

我们有实例 A 和 B ，它们都是 $(0,n)$，这意味着在 X 中，工人具有完全序偏好，企业具有偏序偏好。在 算法 9 中，工人提出提议；与 算法 4 不同的是，企业仅暂时接受自己严格优于所有其他已收到提议的工人——包括当前轮次和此前所有轮次收到的提议者。

引理48。 如果企业 f 在算法 9的任意一轮中拒绝工人 w，那么在 X 下的任何强稳定匹配中，w 和 f 都不可能匹配。

引理*48*的证明。 假设发生了拒绝（否则结论显然成立）。设企业f 拒绝工人w。我们将证明，任何满足$(w,f) \in M$ 的匹配M 都有一个强阻塞对。我们按发生拒绝的轮次进行归纳证明。

假设拒绝发生在第一轮。那么f 只有在收到另一名工人w 的提议时才会拒绝w w' ，而且企业对后者的偏好至少不低于前者。由于w' 在第一轮向f 提出了提议，w' 必定更偏好f 而非$M(w')$。因此，(w',f) 是M的一个强阻塞对。

现在假设拒绝发生在第$k > 1$轮。如果f 拒绝w，那么存在一名工人w'，他在本轮或之前的某一轮向f 提出过申请，并且$w' >_f^X w$ 或$w' \sim_f^X w$。如果w' 在更早的一轮中被f 拒绝，那么根据归纳法假设，w' 在任何稳定匹配中都不能与它在f之前提出过申请的任何企业匹配，包括f。因此，$f >_{w'} M(w')$，所以$(w'、f)$ 是一个强阻塞对。

□

引理49。 如果 X 存在强稳定匹配，那么算法 *9* 会找到一个。

引理*49*的证明。 假设存在一个强稳定匹配，但算法终止并输出 ⊠。只有当某名工人被所有企业拒绝时，才会发生这种情况。然而，根据引理48，这意味着该工人在任何强稳定匹配中都没有可行伴侣，这就产生了矛盾。

□

引理50。 如果 算法 *9* 返回一个匹配，那么它在 X 下必定是强稳定的。

引理*50*的证明 。 假设结论不成立。令(w, f) 为算法返回的匹配中的一个阻塞对。那么，w 必定在执行过程中向f 求过婚，并被拒绝。因此，f在算法结束时必定与一个它严格更喜欢的人匹配，这意味着(w, f)不是强阻塞对，矛盾。

□

引理 51. 如果 算法 *9* 返回一个匹配，那么它必定是工人最优强稳定匹配。

引理5的证明引理 51。 否则，某个工人 w 在工人最优匹配中会有更好的配偶。但在这种情况下，w 必定曾向该配偶求婚，并在算法运行过程中遭到拒绝，从而导致矛盾。

□

B.3 算法 10 正确性的证明

我们再次考虑实例 A 和 B ，它们是 $(0,n)$，这意味着工人具有全序偏好，而企业在 X 中具有偏序偏好。在 算法 10中，企业提出申请；与 算法 4不同的是，它们可以在同一轮迭代中向多名工人提出申请。

引理 52. 如果工人 w 在算法 10的任一轮中拒绝了企业 f，那么在任何同时对实例 A 和 B 稳定的匹配中，w 和 f 都不可能匹配。

证明引理552。 反设存在矛盾，w 在某一轮中拒绝了企业 f ，但二者却在一个稳定匹配中配对 μ。

我们按轮数进行归纳。假设 w 在第一轮拒绝了 f 。那么存在另一家企业 f_1 ， w 严格偏好于 f。由于企业向其最偏好的尚未划去的工人提出申请，且这是第一轮，因此不存在工人 w' ，使得 f_1 比 w 更受偏好；也就是说，$w >_{f_1} w'$ 或 $w \sim_{f_1} w'$。在匹配 μ 中，w 与 f 匹配，这意味着 $(w、f_1)$ 是一个强阻塞对。

假设 w 在第 k 轮拒绝了 f 。那么存在一家企业 f_k ，使得 w 严格偏好它而非 f。以下情况必有一种成立：

1. $w >_{f_k} \mu(f_k)$ 或 $w \sim_{f_k} \mu(f_k)$：那么 (w, f_k) 是一个强阻塞对。 2. $\mu(f_k) >_{f_k} w$：那么 f_k 必定在之前某一轮向 $\mu(f_k)$ 提出过提议，并遭到拒绝。根据归纳假设，这会导致 μ 出现另一个强阻塞对 (w', f')，这意味着 μ 并不稳定。

□

引理 53。 算法 10 终止。

引理553的证明。 对于算法的任意一轮，以下情况必有一种成立：

1. 第 k 轮提出的提议少于 n 个：那么某家企业 f 的列表（有向无环图）中已没有未划掉的工人可供其提出提议。算法会在该轮之前终止，并输出⊠。2. 向 n 名不同工人提出了恰好 n 个提议：那么所有工人都各收到一个提议，并且要么发生拒绝，要么算法在该轮结束时以完美匹配终止。
3. 提出的提议多于 n 个：那么至少有一名工人 w 收到多个提议，拒绝其中至少一个，并且至少有一家企业将 w 从其列表中划掉。

因此，每一轮要么发生一次拒绝，要么算法终止。由于拒绝次数至多为 $O(n^2)$ ，算法会在多项式 (n) 轮内终止。

□

引理 54. 如果 X 存在强稳定匹配，则算法 *10* 会找到一个。

证明引理*54*。 假设存在一个强稳定匹配，但算法以输出 ⊠ 终止。这只会发生在某个企业被所有工人拒绝时。然而，根据引理 52，这意味着该企业在任何强稳定匹配中都没有可行伴侣，这就产生了矛盾。

□

引理 55。 如果算法 *10* 返回一个匹配，那么它在 X 下必定是强稳定的。

引理 *55* 的证明 引理 *55*。 假设并非如此。设 (w, f) 是返回匹配中的一个阻塞对。那么，f必定在算法运行期间向w 求过婚，并遭到拒绝。因此，算法结束时w 必定与一家更受其青睐的企业匹配，这意味着(w, f) 不是强阻塞对——这就产生了矛盾。

□

引理 56。 如果算法 *10* 返回一个匹配，那么它必定是企业最优强稳定匹配。

引理 *56* 的证明引理 *56*。 否则，某个企业f 在企业最优匹配中会有一个更好的配偶。但该企业必定在算法运行期间向该配偶求过婚，并遭到拒绝，这就产生了矛盾。

□

请注意，上述所有证明均适用于任意复合实例 X ：一侧具有偏序偏好，另一侧具有全序偏好。因此，正如 定理 26所述，该设定自然可以推广到两个以上的实例。此外，还需注意，当 X 两侧均具有全序偏好时，即当实例 A 和 B 相同时，这两种算法的行为与原始的延迟接受算法完全一致。