Gilmore, James; Porter, David P.

Article

Match stability with a costly and flexible number of positions

Games

Provided in Cooperation with:
MDPI – Multidisciplinary Digital Publishing Institute, Basel

Suggested Citation: Gilmore, James; Porter, David P. (2025) : Match stability with a costly and flexible number of positions, Games, ISSN 2073-4336, MDPI, Basel, Vol. 16, Iss. 3, pp. 1-12, https://doi.org/10.3390/g16030027

games

Article

Match Stability with a Costly and Flexible Number of Positions

James Gilmore and David Porter *

Economic Science Institute, Chapman University, Orange, CA 92866, USA; jgilmore@chapman.edu
* Correspondence: dporter@chapman.edu

Abstract: One of the main goals of two-sided matching mechanisms is to pair two groups of agents in a stable manner. Stability means that no pair of agents has an incentive to deviate from their assigned match. The outcome of such a match can have significant consequences for the participants involved. Most existing research in this field assumes that the quotas of organizations are fixed and externally determined, which may not always be realistic. We introduce the concept of slot stability, which considers the possibility that organizations may want to adjust their quotas after the match process. To address this issue, we propose an algorithm that generates both stable and slot-stable matches by using flexible, endogenous quotas.

Keywords: matching; stability; quotas

Academic Editor: Ulrich Berger

Received: 17 February 2025
Revised: 25 April 2025
Accepted: 9 May 2025
Published: 21 May 2025

Citation: Gilmore, J., & Porter, D. (2025). Match Stability with a Costly and Flexible Number of Positions. *Games*, *16*(3), 27. https://doi.org/10.3390/g16030027

1. Introduction

Economic research has significantly influenced the development of market institutions, especially in the realm of matching institutions. These institutions are designed to manage the allocation of a limited number of positions or quotas by organizations to applicants, ensuring that each applicant is matched to only one position. The primary objective is to create a process that results in stable matches.

The concept of match stability was originally introduced by Gale and Shapley (1962) in their groundbreaking paper (hereafter, GS). They define a match as stable when there is no compelling incentive for a pair of participants to switch their assignments. In addition, GS introduced the deferred acceptance algorithm, a widely used method to achieve stable matches between applicants and organizations. In this algorithm, applicants submit rank order lists (ROLs) of organizations based on their preferences, while organizations similarly rank applicants according to their preferences. The outcome of this algorithm results in a stable match when the ROLs reveal preferences truthfully.

The basic matching problem has evolved and extended to cases where preferences are more complex. For example, the student-project allocation problem deals with matching students to projects that can have overlapping lecturers while taking into account individual preferences and class capacity constraints. In this environment, Abraham et al. (2007) modified the matching algorithm to ensure stability. Another example is the National Resident Matching Program, which assigns interns to different hospitals and specialties. At first, the algorithm treated every individual's preference as independent of any other individual's preference and gave a stable matching in that environment. However, couples in the match may have joint preferences because they want to be near each other. The deferred acceptance algorithm does not consider this and can produce unstable outcomes. In this environment, Roth and Peranson (1999) proposed a new matching algorithm that incorporates couple preferences, although it does not guarantee a stable match. Nevertheless, computational experiments demonstrate that the algorithm's outcomes closely

approximate stability. Crawford (2008) notes that the Resident Matching program has fixed salary specifications for residents. He extended the GS matching algorithm to allow for flexible salaries, promoting more efficient matches. This extension enables hospitals to offer alternative salaries for each position in the match. Thus, rankings can now reflect salary and location preferences in the matching process.

The matching literature currently defines stability under the assumption of fixed quotas. Organizations specify the maximum number of applicants that they will accept before the match begins. Rios et al. (2021) examined the Chilean university admission system, where the maximum number of slots can exceed the preset quota. Matches are based entirely on academic scores, which can have ties. Therefore, quotas can be exceeded if there is a tie between the accepted worst candidate and any other candidate who wants to join, in which case, they must accept all such candidates. However, this starts by posting a quota and then adjusting it in light of scores. Gokhale et al. (2024) examined how stable matchings shift when baseline quotas are changed (slots are added or subtracted from the baseline). They investigate whether applicants or organizations benefit from these capacity adjustments under various proposing rules. Limaye and Nasre (2021) explored cases where all applicants must be accepted, but the slots are costly. They create an algorithm that produces a minimum-cost stable match that ensures all applicants are successfully matched. However, the algorithm does not address the incentive for the organization to accept these quotas. While the total cost for the stable match is minimized, an organization might find that the marginal cost of an assigned candidate is greater than the value they bring to the organization.

Universities often face the challenge of having more highly qualified applicants than available spots, leading them to consider admitting more students than their initial quotas. This decision involves weighing the benefits of enrolling exceptional students against the practical limitations of campus resources and managing undergraduate enrollment. To manage this, universities employ wait lists for students who are not immediately accepted.

A similar dilemma arises when universities are in the process of recruiting new faculty members. In this case, while the administration may provide a specific number of available positions, academic departments may argue for additional positions if confronted with a pool of high-quality candidates. The ability to assess both the quality of applicants and the associated costs of creating additional positions becomes crucial in making these matching decisions.

In this paper, we extend the GS algorithm to handle the case of variable quotas or positions. We show that if we expand stability to include organizations offering a different number of positions, the current algorithms are not necessarily stable. We have adapted the deferred acceptance algorithm to accommodate a variable number of positions within organizations, ensuring adherence to a broader definition of stability. In particular, we propose a matching mechanism that allows ROLs to accommodate these trade-offs and ensure a stable match that is also slot-stable. By 'slot-stable', we mean that every organization has no incentive to deviate from its number of openings. In the following section, we define our matching environment, including match preferences and the convex costs associated with providing slots.

2. The Environment

Applicants are denoted as a, with indices $i = 1, 2, \ldots, n$, and organizations are denoted as o, with indices $j = 1, 2, \ldots, m$. Each organization o_j has a number of positions or slots to fill. Each applicant can fill one slot with (at most) one organization; the set of applicants admitted to o_j is denoted as A_j. The number of slots filled, s_j, is the cardinality of A_j. Let $V_i(o_j)$ denote applicant a_i's value if they are matched with o_j. Let $Z_j(a_i)$ denote o_j's value if

they are matched with a_i. V_i and Z_j may or may not represent strict preferences. Every a_i and o_j is individually rational and defined by refusing all matches such that $V_i(\emptyset) > V_i(o_j)$ or $Z_j(\emptyset) > Z_j(a_i)$, i.e., applicants and organizations only rank those who improve their value over the remaining unmatched.

Organization o_j has a non-decreasing convex total cost $C_j(x)$ of filling x slots. Specifically, $C_j(x+1) \geq C_j(x)$ and $C_j(x+2) - C_j(x+1) \geq C_j(x+1) - C_j(x)$. Denote $MC_j(x)$ to be the marginal cost of filling slot number x defined by $C_j(x) - C_j(x-1)$. We also assume every a_i ranks the organizations based only on V_i, where a_i prefers o_j over o_k if and only if $V_i(o_j) > V_i(o_k)$. Likewise, o_j ranks the applicants based only on Z_j, where o_j prefers a_i over a_k if and only if $Z_j(a_i) > Z_j(a_k)$.

The Gale–Shapley algorithm does not ensure stability in this context because it assumes fixed quotas for each organization. When an organization ranks applicants, it does not know with whom it will eventually match. Therefore, it cannot determine the value minus the marginal cost of the last slot for its least valued applicant. Below is an example that illustrates this problem.

Suppose we have two organizations o_1, o_2, and three applicants a_1, a_2, a_3. Both organizations have the same values Z_j and costs C_j, with $Z_j(a_1) = 5$, $Z_j(a_2) = 4$, $Z_j(a_3) = 3$, and $MC_j(1) = 2$, $MC_j(2) = 3.5$. For the applicants, their preferences are defined by $V_1(o_2) > V_1(o_1)$, $V_2(o_2) > V_2(o_1)$, $V_3(o_1) > V_3(o_2)$. Tables 1–4 list participants in the columns, while the rows depict the cost, values, or ranking of the object listed in the row.

Tables 1–4: values, costs, and ROLs.

Table 1. Organization $Z_j(a_i)$.

	o_1	o_2
a_1	5	5
a_2	4	4
a_3	3	3

Table 2. Organization $MC_j(x)$.

	o_1	o_2
Slot 1	2	2
Slot 2	3.5	3.5

Table 3. Organization ROLs.

o_1	o_2
a_1	a_1
a_2	a_2
a_3	a_3

Table 4. Applicant ROLs.

a_1	a_2	a_3
o_2	o_2	o_1
o_1	o_1	o_2

The applicant-proposing GS algorithm, when each organization has a fixed quota of 2 slots, results in o_1 being matched with a_3 and o_2 being matched with a_1 and a_2. This yields

a stable match, and neither organization has any incentive to want to change its quota. However, if a_2's preference was $V_2(o_1) > V_2(o_2)$, their ROL would now be o_1, o_2, and the applicant-proposing GS match would have o_1 matched with a_2 and a_3 and o_2 matched with a_1. Notice that with o_1 having two slots filled, a_3 in slot 2 has a value of 3, but the marginal cost of providing that slot to a_3 is 3.5, resulting in a loss of 0.5 being matched with a_3. Because of this, o_1 would prefer to leave the second slot unfilled. Here, o_1 has set their quota too high.

Now, suppose organizations have the same costs and values as before, but the quotas are 1 for each organization. Applicant preferences are the same as in the first example: $V_1(o_2) > V_1(o_1)$, $V_2(o_2) > V_2(o_1)$, $V_3(o_1) > V_3(o_2)$. The GS match would have o_1 and a_2 matched and o_2 matched with a_1. This match is stable; however, o_2 can do better. Here, o_2 would be willing to open a slot for a_2, and a_2 prefers o_2 over their current match, which would cause both to be better off. Here, o_2 sets their quota too low.

These examples highlight the need for a new form of stability that considers the quotas of organizations. Firstly, if an organization, o_j, can benefit by adding a slot for an applicant, a_i, who is matched with another organization o_u but would prefer o_j, then it is considered slot unstable. Secondly, if o_j gains by removing a slot and thus not matching an a_i in A_j, it is also slot unstable. Therefore, we propose the following definition to address this issue.

Definition 1. *A match is said to be slot-stable*[1] *if and only if the following two conditions hold for all j with* $|A_j| = s_j$

$$MC_j(s_j + 1) \geq Z_j(a_i) \;\; \forall a_i \notin A_j,\; a_i \in A_u :\; V_i(o_j) > V_i(o_u) \tag{1}$$

and

$$Z_j(a_k) \geq MC_j(s_j) \;\; \forall a_k \in A_j \tag{2}$$

Condition (1) specifies that for an organization, the cost of adding an extra slot for a willing applicant not currently in the match must not be less than the applicant's value; otherwise, the organization would prefer to include them. Condition (2) requires that the value of an applicant already in the match exceeds or equals the marginal cost of the final slot filled by the organization they are matched with; otherwise, the organization would not want to include them.

3. Matching Mechanisms

This section assumes that applicants and organizations submit ROLs consistent with their payoffs.[2] This section introduces applicant-proposing and organization-proposing algorithms that ensure stability and slot stability. These processes require organizations to provide a more detailed Rank Order List (ROL) that includes the costs associated with offering positions. Specifically, the organization's ROLs feature cutoff points that are similar to a waiting list. The new algorithm uses these cutoffs to align the organization's preferences and costs. While both ENPAP and ENPOP algorithms ensure stable matches that are also slot-stable, the matches they produce may not be identical. In our analysis, we demonstrate that the proposing side in our algorithms achieves its optimal match. In practice, only one algorithm is implemented to finalize the match. The choice of which proposed side algorithm to use depends on the organization conducting the match. For instance, the National Resident Matching Program (NRMP), a nonprofit organization, employs the applicant-proposing algorithm rather than the hospital-proposing algorithm.

3.1. Endogenous-Number-of-Positions Applicant-Proposing (ENPAP) Algorithm

3.1.1. ENPAP Inputs

As in the GS process, applicants submit ROLs listing organizations from their most to least preferred. For organizations, we will need an adjustment where organizations provide a *cutoff list* of rankings, henceforth called rank order cutoff lists ROCL. These ROCLs will be built as if organizations revealed their preferences, given the cost of providing the slots. First, we look at o_j's ROL of length k of applicants $a_{j,i} \in A$: $\text{ROL}_{j,k} = \{a_{j,1}, a_{j,2}, \ldots a_{j,k} : a_{j,i} \in A, Z_j(a_{j,1}) \geq Z_j(a_{j,2}) \geq \ldots \geq Z_j(a_{j,k})\}$, that is, we have labeled the applicants in the order of their value from highest to lowest for the first k slots. To create the rank order cutoff list for organization j, (ROCL_j), we find the **first** cutoff point, $n^*_{j,1}$, which is the largest number of applicants whose values exceed the marginal cost, assuming that all positions were filled. Thus, starting with **1** slot, we define $B^1_j(\mathbf{1}) = \{a_{j,1}, a_{j,2}, \ldots, a_{j,k} : Z_j(a_{j,k+1}) \leq MC_j(\mathbf{1})\}$. Thus, if we have **2** slots, $B^1_j(\mathbf{2}) = \{a_{j,1}, a_{j,2}, \ldots, a_{j,k} : Z_j(a_{j,k+1}) \leq MC_j(\mathbf{2})\}$. We keep doing this until we find the largest number of slots $n^*_{j,1}$ such that $B^1_j(n^*_{j,1}) = \{a_{j,1}, a_{j,2}, \ldots, a_{j,k} : Z_j(a_{j,k+1}) \leq MC_j(n^*_{j,1})\}$. We define $n^*_{j,1}$ as the first cutoff point for organization j. Note that the cardinality of the set, $|B^1_j(n^*_{j,1})|$ can be greater than or equal to $n^*_{j,1}$, since there can be more than $n^*_{j,1}$ applicants who have a value over the marginal cost of the $n^*_{j,1}$th slot. The organization values and MC have been reproduced from our previous example in Tables 5 and 6 below to demonstrate the cutoff list construction.

Table 5. Organization $Z_j(a_i)$.

	o_1	o_2
a_1	5	5
a_2	4	4
a_3	3	3

Table 6. Organization $MC_j(x)$.

	o_1	o_2
Slot 1	2	2
Slot 2	3.5	3.5
Slot 3	7	7

For organization 1, the first cutoff point is 2, since the MC(2) = 3.5, which is greater than the value of a_3, which is 3. Thus, $B^1_1(2) = \{a_1, a_2\}$.

The next cutoff point, $n^*_{j,2}$, is found in the same manner as $n^*_{j,1}$, except it is done over the set of remaining applicants, $A \setminus B^1_j(n^*_{j,1})$. For the second cutoff point, we define $B^2_j(n^*_{j,2})$ over the set $A \setminus B^1_j(n^*_{j,1})$. Again, we look for the largest number of slots $n^*_{j,2}$ where the last applicant in the list $B^2_j(n^*_{j,2})$ covers the marginal cost $\text{MC}(n^*_{j,2})$. We continue this process until the cutoff point is 0.

The intuition behind this series of cutoff points is straightforward. For $(n^*_{j,1})$, all applicants in $B^1_j(n^*_{j,1})$ are acceptable for these slots since they cover their marginal costs. However, if we match with less than $(n^*_{j,1})$ applicants, the organization can refer to the next set of applicants $A \setminus B_j(n^*_{j,1})$, and the list $B^2_j(n^*_{j,2})$ (think of it as a waiting list) to fill the remaining slots.

From our previous example, with $B^1_1(n^*_{1,1}) = \{a_1, a_2\}$ and $n^*_{1,1} = 2$, if neither a_1 nor a_2 matched with the organization, a_3 would cover the marginal cost for the organization if only one slot could be filled. Thus, the ROCL would be written as [a_1, a_2, 2, a_3, 1].

3.1.2. ENPAP Algorithm

As with the GS algorithm, all applicants begin by proposing to the organization at the top of their ROL. Each organization, o_j, reviews the applicants who proposed to them and are in the first cutoff, $B_j^1(n_{j,1}^*)$. Let A_j^1 represent the applicants in $B_j^1(n_{j,1}^*)$ who proposed to o_j. The organization, o_j, tentatively accepts the top candidates up to $n_{j,1}^*$. If the number of tentatively accepted candidates is less than $n_{j,1}^*$, o_j considers applicants in the second cutoff list, $B_j^2(n_{j,2}^*)$.

Let T_j^1 denote the list of tentatively accepted applicants from the first cutoff list $B_j^1(n_{j,1}^*)$. The algorithm then calculates $r_{j,2} = n_{j,2}^* - |T_j^1|$. If $r_{j,2} > 0$, o_j tentatively accepts the rank-ordered candidates in the second cutoff list up to $r_{j,2}$. This process continues with subsequent cutoff lists and so on.

Applicants who propose to o_j but are not tentatively accepted are rejected, and then propose to the next highest organization on their list. Rejected candidates from other organizations are then considered alongside the tentatively accepted applicants ($T_j^1 \cup T_j^2 \cup \ldots T_j^k$), and the algorithm proceeds with this new list in the same manner as above.

To illustrate, in Table 7 we use the applicant valuations $V_1(o_1) > V_1(o_2)$, $V_2(o_1) > V_2(o_2)$, $V_3(o_1) > V_3(o_2)$, along with the ROCL [a_1, a_2, 2, a_3, 1] for both o_1 and o_2. First, each applicant proposes to their highest-valued, individually rational organization, as depicted below.

Table 7. Applicants' first proposal.

o_1	o_2
a_1	$\emptyset$
a_2	$\emptyset$
a_3	$\emptyset$

Looking at the ROCL of o_1, [a_1, a_2,2,a_3,1], a_3 is rejected by o_1, and a_1 and a_2 are tentatively accepted. After being rejected, a_3 proposes to o_2, who accepts them since a_3 was ranked in cutoff 2 and $r_{2,2} = 1$.

Theorem 1. *ENPAP results in a stable match.*[3]

Proof. Suppose the ENPAP match is unstable, then $\exists a_i, o_j$ matched with o_u, a_u, such that $V_i(o_j) > V_i(o_u)$ and $Z_j(a_i) > Z_j(a_u)$. For a_i and o_j to not be matched with each other, either a_i never proposed to o_j or o_j rejected a_i.

If a_i never proposed to o_j, one of two scenarios could have happened:

(ia) a_i never put o_j on their list. If o_j is not on a_i's list, then $V_i(\emptyset) > V_i(o_j)$. All o_k ranked by a_i must satisfy $V_i(\emptyset) < V_i(o_k)$; specifically, $V_i(o_u) > V_i(\emptyset)$. Therefore, regardless of whether a_i is matched with no one or with any o_k in their ROL, $V_i(o_j) > V_i(o_u)$ is false.

(ib) a_i never proposed o_j on their ROL. However, since a_i matched with o_u, a_i must have proposed to o_u. But with $V_i(o_j) > V_i(o_u)$, this implies that o_j was ranked before o_u in a_i's ROL. This means that a_i has to propose to o_j before o_u in the algorithm.

(ii) From (ib), we know that a_i proposed to o_j. By assumption, we know that a_u proposed to o_j and was accepted. For a_u to be accepted, we must have $Z_j(a_u) \geq MC_j(s)$, where s is the number of accepted applicants. Since $Z_j(a_i) > Z_j(a_u)$, a_u could not have replaced a_i if they were tentatively accepted. When a_i proposes to o_j with a_u tentatively accepted, then a_i must replace another candidate in the tentatively accepted list. No other applicant who proposes to o_j will cause a_i to be rejected since a_u is accepted and $Z_j(a_i) > Z_j(a_u)$; otherwise, a_u would be rejected, which is a contradiction. □

Theorem 2. *ENPAP results in a slot-stable match.*[4]

Proof. Assume ENPAP results in slot instability. Then, by definition, there exists an o_j and either an $a_k \in A_j$ or an $a_i \notin A_j$ with $|A_j| = s_j$, such that

$$(1)\ \ MC_j(s_j + 1) < Z_j(a_i)\ \ a_i \in A_u :\ V_i(o_j) > V_i(o_u)$$

or

$$(2)\ \ Z_j(a_k) < MC_j(s_j)$$

From (1), $V_i(o_j) > V_i(o_u)$ implies that a_i would have proposed to o_j before o_u. Given $MC_j(s_j + 1) < Z_j(a_i)$, o_j would not have rejected a_i. Thus, $a_i \in A_j$, which is a contradiction.

Turning to (2), the cutoff points would not allow any applicant to be accepted who did not cover the MC of s. Since we have $Z_j(a_k) < MC_j(s_j)$, this implies $a_k \notin A_j$, which is a contradiction.

Since the algorithm cannot produce a match that satisfies either (1) or (2), the ENPAP must give a slot-stable match. □

Among the set of stable and slot-stable matches, an *applicant-optimal match* is the one that assigns applicants to their highest-ranked feasible organization.

Theorem 3. *ENPAP results in an applicant-optimal match.*[5]

Proof. The theorem is demonstrated through inductive reasoning based on the steps of the ENPAP algorithm. We start with the following premise: An organization, o_j, is considered reachable for applicant a_i if there exists at least one stable and slot-stable pairing, where o_j is matched with a_i.

Initially, in step 1, no applicant is turned away by any reachable organization. Assume this remains true up to the beginning of step k. In step k, applicant a_i receives a rejection from the reachable organization o_j. If o_j rejects a_i, o_j must have received a more favorable proposal from, say, a_b, or o_j prefers to be matched with no one. If o_j prefers to be matched with no one, then o_j is unreachable for a_i.

If o_j rejects a_i in favor of a_b, a_b must have proposed to o_j on or before step k. Furthermore, considering that no applicants were rejected by a reachable organization before step k, a_b must not have faced rejection from such an organization before proposing to o_j. Since applicants make proposals in the order ranked by their preference function V, o_j is their most preferred reachable choice.

If o_j is indeed reachable for a_i, then there exists a stable and slot-stable match that pairs o_j with a_i. In this scenario, a_b must be unpaired, paired to another reachable organization, or with o_j. If both a_b and a_i are in a stable and slot-stable match with o_j, then a_i would not have been rejected because of a_b's better proposal.

If a_b is unpaired or paired with another reachable organization, a_b would favor o_j over this match. Moreover, it is clear that o_j values a_b over a_i; thus, a_b and o_j would block this match, causing instability, implying o_j was never reachable for a_i.

In conclusion, since no applicant was rejected by a reachable organization before k, and rejections cannot occur by these organizations in step k, no applicant will face rejection by a reachable organization. Given that applicants propose according to their preferences within ENPAP, each applicant ultimately matches with the organization they favor the most among all stable and slot-stable pairings. □

3.2. Endogenous-Number-of-Positions Organization-Proposing (ENPOP) Algorithm

3.2.1. Inputs

We will use the same ROL and ROCL inputs as those used in the ENPAP algorithm described in Section 3.1.1.

3.2.2. Algorithm

Step 1: Each organization proposes to its top $n^*_{j,1}$

Step 2: Each a_i chooses their most preferred o_j among the organizations that proposed to them. For each o_j not chosen by a_i, a_i is removed from its ROCL.

Step 3: Organizations then propose to applicants on their lists from step 2, including those on later cutoff lists, as long as $r_{j,k} > 0$.

Step 4: Repeat steps 2 and 3 until no applicant has multiple organizations proposing to them.

To illustrate this, in Table 8 we use the applicant valuations $V_1(o_2) > V_1(o_1)$, $V_2(o_2) > V_2(o_1)$, $V_3(o_1) > V_3(o_2)$, along with the organization valuations for both organizations, leading to their respective ROCLs being [a_1, a_2, 2, a_3, 1]. First, each o_j proposes to its top two applicants as shown below.

Table 8. Organization-proposing first list.

o_1	o_2
a_1	a_1
a_2	a_2

Since both a_1 and a_2 receive proposals from both o_1 and o_2, they choose between them. In this case, both a_1 and a_2 choose o_2. We then repeat the process where o_2 submits the same list; however, o_1 submits a new optimal list [a_3] since their preferred candidates, a_1 and a_2, are tentatively in o_2's list. This leads to the final match as shown in Table 9.

Table 9. Organization-proposing match.

o_1	o_2
a_3	a_1
$\emptyset$	a_2

Theorem 4. *ENPOP results in a stable match.*[6]

Proof. Suppose ENPOP results in an unstable match. This means that there is a blocking pair, a_i and o_j. For this to occur, o_j must have put an applicant, a_u, on their ROCL that is ranked worse than a_i, i.e., $Z_j(a_i) > Z_j(a_u)$. Given the construction of the ROCL, this can only occur if a_i is unavailable, i.e., either unmatched or matched with another organization, o_u. This only happens when $V_i(o_u) > V_i(o_j)$ or $V_i(\emptyset) > V_i(o_j)$. However, since a_i and o_j are a blocking pair, it must be that $V_i(o_j) > V_i(o_u)$, which is a contradiction. Therefore, ENPOP must result in a stable match. □

Theorem 5. *ENPOP results in a slot-stable match.*[7]

Proof. Assume ENPOP results in slot instability, then by definition, there is an o_j, and either an $a_k \in A_j$ or $a_i \notin A_j$ with $|A_j| = s_j$, such that we have the following:

$$(1)\ \ MC_j(s_j + 1) < Z_j(a_i)\ \ a_i \in A_u :\ V_i(o_j) > V_i(o_u)$$

or

$$(2)\ \ Z_j(a_k) < MC_j(s_j)$$

Suppose a_i is ranked lower than the current set of accepted applicants, A_j. According to the rules of the ROCL, all of the applicants in A_j would receive proposals prior to a_i. Consequently, the organization, o_j, is free to extend proposals to higher-ranked applicants than a_i that are not in A_j, since they will all have a value higher than $MC(s_j + 1)$. Should any of these higher-ranked applicants accept the proposal, it would imply that $|A_j| > s_j$, leading to a contradiction. Conversely, if all higher-ranked applicants decline, o_j will propose to a_i. a_i will be accepted because they will be unmatched or they will be currently paired with a lower valued organization since $a_i \in A_u : V_i(o_j) > V_i(o_u)$. Again, it must be that $|A_j| > s_j$, which is a contradiction.

On the other hand, if a_i is not ranked lower than every applicant in A_j, then there is an a_h in A_j such that $Z_j(a_i) > Z_j(a_h) > MC(s_j)$. This would mean o_j would rather be matched with a_i than with a_h. In addition, with $V_i(o_j) > V_i(o_u)$, a_i would like to be matched with o_j than with o_u. So the outcome is not stable, contradicting Theorem 4. Thus, (1) is violated.

Given the construction of ROCLs, for a_k to be selected with s_j slots filled, $Z_j(a_k) > MC_j(s_j)$, which violates (2).

Thus, neither (1) nor (2) can hold true, concluding that ENPOP yields a slot-stable match. □

Among the set of stable and slot-stable matches, an *organization-optimal match* is one that assigns applicants to organizations so that no organization would like to switch to another stable and slot-stable match. Using the same inductive reasoning in Theorem 3, we have the following:

Theorem 6. *ENPOP results in an organization-optimal match.*[8]

Proof. The theorem is demonstrated through inductive reasoning based on the steps of the ENPOP algorithm. We start with the following premise: An applicant, a_i, is considered reachable for organization o_j if there exists at least one stable and slot-stable pairing, where a_i is matched with o_j.

Initially, in step 1, no organization was turned away by any reachable applicant. Assume that is the case up until the beginning of step k. In step k, organization o_j receives a rejection from the reachable applicant, a_i. If a_i rejects o_j, a_i must have received a more favorable proposal from, say, o_k, or a_i prefers to be matched with no one. If a_i preferred to be matched with no one, then ai_j is unreachable for o_j.

If a_i rejected o_j in favor of o_k, o_k must have proposed to a_i on or prior to step k as part of their list. Furthermore, considering that no organizations were rejected by a reachable applicant before step k, o_j must not have faced rejection from such an applicant before proposing to a_i. Since organizations offer their proposals ranked by their preference function Z and MC, a_i is in the set of their most preferred, reachable set.

If a_i is indeed reachable for o_j, then there exists a stable and slot-stable match that pairs o_j with a_i. Within this scenario, o_k must not be matched with a_i. Since o_k put a_i into their most preferred set, whatever match o_k has without a_i would be worse than a corresponding match with a_i.

If o_k is not matched with a_i, o_k would favor a match that includes a_i over this match. Moreover, it is clear that a_i values o_k over o_j; thus, a_i and o_k would block this match, causing instability, implying a_i was never reachable for o_j.

In conclusion, since no organization was rejected by a reachable applicant before k, and it is impossible for rejections to occur by these applicants in step k, no organization will

face rejection from a reachable applicant. Given that organizations propose according to their preferences within ENPOP, each organization ultimately matches with the applicants they favor the most among all stable and slot-stable pairings. □

3.3. Unique Set of Slot-Stable-Filled Slots

In our environment with strict preferences, we demonstrate that each organization has a unique set of filled positions leading to a slot-stable match, similar to the rural hospital theorem (Roth, 1986). This uniqueness property underlines the unlikely situation of organizations setting exact quotas that will lead to a slot-stable match.

We begin by *converting* the many-to-one matching problem into a one-to-one matching scenario by leveraging the ROLs provided by applicants and the ROCLs from organizations. For any given organization, o_j, it offers between 1 and n_j slots. We use the notation $o_{j,x}$ to reference the xth slot of o_j. The ROL for every slot $o_{j,x}$ is defined by the cutoff list $B_j(x)$, effectively treating each slot as a separate organization in the one-to-one match.

Every applicant, a_i, arranges their one-to-one ROLs such that organization o_j's xth slot, $o_{j,x}$, is placed higher than organization o_k's yth slot, $o_{k,y}$, if and only if applicant a_i prefers o_j over o_k in their original many-to-one ROL. Moreover, within the same organization, o_j, the xth slot, $o_{j,x}$, is favored over the yth slot, $o_{j,y}$, if x is less than y. Note that if $V_i(o_j) > V_i(o_k)$, then for every x, $o_{j,x}$ is ranked above the o_k list of slots.

Lemma 1. *An ordered many-to-one match is stable and slot-stable if and only if the converted one-to-one match is stable.*

Proof. Assume that in the ordered many-to-one match, o_j assigns its most preferred applicant in A_j in slot one, its second most preferred in slot two, and so forth. For the lemma to hold, the following three properties must be true: (i) if the corresponding one-to-one match is stable, then the many-to-one match is stable; (ii) if the one-to-one match is stable, the many-to-one match is slot-stable; (iii) if the ordered many-to-one match is stable and slot-stable, the one-to-one match is stable.

For (i), if the one-to-one match is stable while the many-to-one match is unstable, there must exist an applicant, a_i, matched with organization o_k, and an applicant, a_k, matched with organization o_j, such that a_i prefers o_j to o_k and o_j prefers a_i to a_k in the many-to-one match. For the corresponding one-to-one stable match, a_i must be matched with one of o_k's slots, $o_{k,x}$, and a_k must be matched with one of o_j's slots, $o_{j,y}$.

Since a_i ranks $o_{j,y}$ above $o_{k,x}$, if a_i has o_j ranked above o_k, then a_i must prefer $o_{j,y}$ to $o_{k,x}$. Since $o_{j,y}$ shares the same ROL ordering as o_j, this would mean that the one-to-one match is unstable as a_i would prefer $o_{j,y}$ and $o_{j,y}$ would prefer a_i over a_k, leading to a contradiction of the stability in the one-to-one match. For (ii), suppose the one-to-one match is stable while the many-to-one match is not slot-stable, then by definition, there is an o_j and either an $a_k \in A_j$ or $a_i \notin A_j$ with $|A_j| = s_j$, such that

$$MC_j(s_j + 1) < Z_j(a_i) \;\; a_i \in A_u : \; V_i(o_j) > V_i(o_u) \tag{3}$$

or

$$Z_j(a_k) < MC_j(s_j) \tag{4}$$

If (3) is true, then there exists an applicant, a_i, such that $MC(s_j + 1) < Z_j(a_i)$ and ranks below all other tentatively accepted applicants, or there exist applicants a_i, a_u such that $Z_j(a_i) > Z_j(a_u)$ and $MC_j(s_j + 1) < Z_j(a_u)$. For the first case, if applicant a_i prefers organization o_j to o_u, then a_i must prefer $o_{j,y}$ over $o_{u,x}$. From the slot side, o_{j,s_j+1} must have a_i above not being filled if $MC_j(s_j + 1) < Z_j(a_u)$ as a_i would be in $B_j(s_j + 1)$, which defines

o_{j,s_j+1}'s ROL. Therefore, since o_{j,s_j+1} and a_i prefer being with each other versus the original match, the one-to-one match must be unstable. For the second case, if $V_i(o_u) < V_i(o_j)$, then a_i prefers $o_{j,y}$ over $o_{u,x}$ through the construction of a_i's one-to-one ROL. Since $o_{j,y}$ shares the same ROL as o_j, $o_{j,y}$ must prefer a_i over their current applicant, a_u. The one-to-one match is unstable since both $o_{j,y}$ and a_i prefer each other over their current match.

If (4) is true, then there must exist an applicant, a_k, that is, the lowest-value $a_i \in A_j$ matched together, such that $MC_j(s_j) > Z_i(a_k)$. For this to be true, a_k cannot be a part of o_j's cutoff list $B_j(s_j)$. This would mean that slot o_{j,s_j} does not have a_k in its ROL. This would make the one-to-one match unstable as it is not individually rational due to o_{j,s_j} preferring not to be matched at all.

Thus, neither (3) nor (4) can hold true, leading to a contradiction.

For (iii), suppose the ordered many-to-one match is stable and slot-stable, while the one-to-one match is unstable. Then there must exist an applicant, a_i, matched with slot $o_{k,x}$, and an applicant, a_u, matched with slot $o_{j,y}$, such that a_i and $o_{j,y}$ prefer each other over their current match. This can happen if $j \neq k$ or $j = k$.

If $j \neq k$, then a_i must prefer organization o_j over organization o_k as a_i ranks $o_{j,y}$ above $o_{k,x}$ if and only if a_i has o_j ranked above o_k. Since o_j ranks applicants in the same order as its slots, then o_j must prefer a_i over applicant a_u that is in slot $o_{j,y}$. Since o_j prefers a_i over a_u and a_i prefers o_j over o_k, the many-to-one match must be unstable.

For the case where $j = k$, this would imply that the many-to-one match is unordered, as a lower-ranked applicant must be in a slot higher than the higher-ranked applicant—a contradiction.

We have shown that (i)–(iii) are true, so that an ordered many-to-one match is stable and slot-stable if and only if the corresponding one-to-one match is stable. □

Theorem 7. *For any set of applicants and organizations, there is only one set of filled slots that is both stable and slot-stable.*

Proof. Lemma 1 shows that the constructed ROLs lead to both stable and slot-stable outcomes, similar to those produced by our algorithm. Roth and Sotomayor (1990) demonstrated that for any one-to-one match with strict preferences, the group of unassigned agents, including both applicants and organizations, remains consistent across all stable matches. Consequently, for a one-to-one match to maintain stability, it must involve the matching of identical slots. Therefore, given that the matched slots are always the same and the many-to-one match is both stable and slot-stable, each organization must retain the same count of applicants across all matches that are stable and slot-stable. □

4. Discussion

We developed an enhanced matching algorithm that builds upon the Gale–Shapley (GS) algorithm by incorporating the costs associated with providing applicant slots. This refined approach ensures that the outcomes remain stable by taking into account these costs when organizations create their rank order lists (ROLs). Our algorithm's innovation lies in achieving what we term 'slot stability' within this environment.

Under the current system, achieving slot stability is improbable since the slot-stable set of quotas is unique. Our new process effectively prevents organizations from independently altering the number of slots they provide, as it aligns their incentives with the stable matching principles.

Author Contributions: Methodology, D.P.; Formal analysis, J.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Notes

1 In the case where indifference is allowed, slot stability can be defined as "weak" if there exists no strict incentive for the applicant to change ($V_i(o_j) > V_i(o_u)$), or "strong", where there exists no applicant who is indifferent to a possible change ($V_i(o_j) \geq V_i(o_u)$).

2 Just like with GS matching, the non-proposing side may not be incentivized to reveal their true rankings. Our mechanisms ensure that the match with truthful rankings will be stable.

3 For the case where valuations reflect indifference (i.e., $V_i(o_j) \geq V_i(o_u)$ and $Z_j(a_i) \geq Z_j(a_u)$) and participants submit their ranking lists according to their valuations while randomizing among options of equal value, the proof remains valid. However, the matches are now weakly stable. This means that no "blocking pairs" prefer to match with each other over their current partners.

4 For the case where valuations reflect indifference, as in Note 3, we guarantee a weakly slot-stable match.

5 For the case where valuations reflect indifference, as in Note 3, we cannot ensure the match with ENPAP is applicant-optimal.

6 For the case where valuations reflect indifference, as in Note 3, the proof remains valid. However, the matches are now weakly stable.

7 For the case where valuations reflect indifference, as in Note 3, ENPOP results in a weakly slot-stable match.

8 For the case where valuations reflect indifference, as in Note 3, we cannot ensure the match with ENPOP is organization-optimal.

References

Abraham, D., Irving, R., & Manlove, D. (2007). Two algorithms for the student-project allocation problem. *Journal of Discrete Algorithms*, *5*(1), 73–90. [CrossRef]

Crawford, V. (2008). The Flexible-Salary Match: A proposal to increase the salary flexibility of the National Resident Matching Program. *Journal of Economic Behavior and Organization*, *66*, 149–160. [CrossRef]

Gale, D., & Shapley, L. (1962). College admissions and the stability of marriage. *The American Mathematical Monthly*, *69*(1), 9–15. [CrossRef]

Gokhale, S., Narang, S., Singla, S., & Vaish, R. (2024). Capacity modification in the stable matching problem. *arXiv*, arXiv:2402.04645.

Limaye, G., & Nasre, M. (2021). Stable matchings with flexible quotas. *arXiv*, arXiv:2101.04425.

Rios, I., Tomas, L., Parra, G., & Cominetti, R. (2021). Improving the chilean college admissions system. *Operations Research*, *69*(4), 1186–1205. [CrossRef]

Roth, A. (1986). On the allocation of residents to rural hospitals: A general property of two-sided matching markets. *Econometrica*, *54*(2), 425–427 [CrossRef]

Roth, A., & Peranson, E. (1999). The Redesign of the matching market for american physicians: Some engineering aspects of economic design. *American Economic Review*, *89*(4), 748–780. [CrossRef] [PubMed]

Roth, A., & Sotomayor, M. (1990). *Two-sided matching: A study in game-theoretic modeling and analysis*. Econometric Society Monographs. Cambridge University Press.