Spring 5-2024

Match Stability with a Costly and Flexible Number of Positions

James Gilmore
Chapman University, jgilmore@chapman.edu

Recommended Citation
Gilmore, J. (2024). *Match stability with a costly and flexible number of positions* [Master's thesis, Chapman University]. Chapman University Digital Commons. https://doi.org/10.36837/chapman.000597

Match Stability with a Costly and Flexible Number of Positions

A Thesis by

James Gilmore

Chapman University

Orange, CA

Argyros School of Business & Economics

Submitted in partial fulfillment of the requirements for the degree of

Master of Science in Behavioral and Computational Economics

May 2024

Committee in charge:

David Porter, Ph.D.,Committee Chair

The thesis of James Gilmore is approved

David Porter

David Porter, Ph.D., Chair

May 2024

Match Stability with a Costly and Flexible Number of Positions

by James Gilmore

ABSTRACT

Match Stability with a Costly and Flexible Number of Positions

by James Gilmore

One of the objectives of two-sided matching mechanisms is to pair two groups of agents such that there is no incentive for pair deviation. The outcome of a match can significantly impact participants. While much of the existing research in this field addresses the matching with fixed quotas, this is not always applicable. We introduce the concept of slot stability, recognizing the potential motivation for organizations to modify their quotas after the match. We propose an algorithm designed to create stable and slot stable matches by employing flexible, endogenous quotas to address this issue.

TABLE OF CONTENTS

LIST OF TABLES

1 Introduction

Economic research has played a pivotal role in shaping the development of market institutions, with a particular emphasis on matching institutions. The conventional framework for addressing the matching problem involves organizations offering a set number of positions or quotas that need to be filled by applicants. Within this framework, each applicant can be assigned to at most one position. Matching institutions have been meticulously crafted to tackle these intricate challenges, aiming to provide a stable solution.

The concept of match stability was originally introduced by Gale and Shapley [1] 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. Additionally, GS introduced the deferred acceptance algorithm [1], a widely employed method for achieving 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 ROLs truthfully reveal preferences.

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. [2] modify the matching algorithm to ensure stability. Another example is in 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 [3] proposed a new matching algorithm that incorporates couple preferences, although it does not guarantee a stable match. Nonetheless, computational experiments demonstrate that the algorithm's outcomes closely approximate stability.

The matching literature currently defines stability under the assumption of fixed quotas. Organizations state the maximum number of applicants they will accept before the match begins. Rios et al. [4] examined the Chilean college 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. Limaye and Nasre [5] explore cases where all applicants must be accepted with costly slots. They then minimize the total cost to get a stable match with minimal cost. However, this does not address the incentive for the organization to accept these quotas. Here, there is minimal cost, yet there may be some excellent candidates the organization would be willing to accept at a higher cost.

In the context of university admissions, educational institutions often grapple with a challenging dilemma. They frequently find themselves with a surplus of highly qualified applicants, compelling them to consider increasing the number of admitted students beyond their initial enrollment quotas. However, this decision is not taken lightly, as universities must balance the advantages of admitting exceptional students and the practical constraints of managing undergraduate enrollment while considering campus resource costs. To navigate this complex scenario, universities have implemented wait lists for students who have not yet received acceptance offers.

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 pivotal in making these matching decisions.

In this paper, we show that if we expand stability to include organizations offering a different number of positions, the current algorithms are not necessarily stable. We show how a small change to the deferred acceptance algorithm allows for endogenous numbers of slots by organizations while guaranteeing this expanded 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 in their number of openings. We also show that our matching mechanism considers organizations' concerns in a wait list system and provides a solution to the endogenous quota problem.

2 The Environment

Applicants are denoted as a, with indices $i = 1, 2, ..., n$, and organizations are denoted as o, with indices $j = 1, 2, ..., m$. Each organization o_j has a number of positions or *slots* to fill. Each applicant can fill one slot with at most one organization, and the set of these applicants admitted to o_j is denoted as A_j. The amount 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. Both V_i and Z_j are one-to-one functions. 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 that improve their value over 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 GS algorithm does not guarantee stability in this environment. Below is an example illustrating the issue with the fixed quota assumption.

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 $V_1(o_2) > V_1(o_1)$, $V_2(o_2) > V_2(o_1)$, $V_3(o_1) > V_3(o_2)$. The following tables list the participants in the columns, while the rows depict the cost, values, or ranking of the object listed in the row.

	o_1	o_2
a_1	5	5
a_2	4	4
a_3	3	3

Organization $Z_j(a_i)$

	o_1	o_2
Slot 1	2	2
Slot 2	3.5	3.5

Organization $MC_j(x)$

o_1	o_2
a_1	a_1
a_2	a_2
a_3	a_3

Organization ROLs

a_1	a_2	a_3
o_2	o_2	o_1
o_1	o_1	o_2

Applicant ROLs

Tables 1-4: Values, Costs and Lists

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, the value of a_3 in slot 2 has a value of 3 but a marginal cost of 3.5, resulting in a loss of .5. Because of this, o_1 would prefer to leave the second slot unfilled since $MC_1(s_2) > Z_1(a_3)$. Here o_1 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 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 set their quota too low.

These examples demonstrate that another form of stability concerning organization quotas should be addressed. First, if organization o_j stands to gain by adding a slot for an a_i matched with some o_u that would prefer to be matched with o_j, it is slot unstable. Second, if organization o_j profits by eliminating a slot and terminating an a_i in A_j, it is slot unstable. Hence, we offer the following definition.

Definition: A match is said to be *slot stable* if and only if

$$(1)\ Z_j(A_j)-C_j(s_j) \geq Z_j(A_j \cup a_i)-C_j(s_j+1)\ \ \forall a_i \notin A_j,\ a_i \in A_u,\ \ V_i(o_u) < V_i(o_j),\ \ \forall j \in 1,2,....,m$$

and

$$(2)\ \ Z_j(A_j) - C_j(s_j) \geq Z_j(A_j \setminus a_i) - C_j(s_j - 1)\ \ \forall a_i \in A_j,\ \ \forall j \in 1,2,....,m$$

This can also be written in terms of marginal costs.

$$(1a)\ \ MC_j(s_j+1) \geq Z_j(a_i)\ \ \forall a_i \notin A_j,\ a_i \in A_u\ V_i(o_j) > V_i(o_u)$$

and

$$(2a)\ \ Z_j(a_i) \geq MC_j(s_j)\ \ \forall a_i \in A_j$$

3 Matching Mechanisms

This section assumes that applicants and organizations submit ROLs consistent with their payoffs.[1]

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

3.1.1 Inputs

Applicants submit ROLs listing organizations from their most to least preferred that are better than not being matched at all. For the organizations, we will need an adjustment where organizations provide a *cutoff list* of rankings, henceforth called a ROCL. First, o_j lists their applicants in rank order best to least. The first cutoff, $n_{j,1}$, is defined by the ordered list of top applicants $B_j(n_{j,1})$ that would be acceptable within the $n_{j,1}$ slots such that $|B_j(n_{j,1})| \geq n_{j,1}$. This list is all of the applicants ranked above $n_{j,1}$ in the submitted ROCL.

[1] 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.

Next, all applicants below $n_{j,1}$ and above $n_{j,2}$ are the set of applicants an organization is willing to accept if less than $n_{j,2}$ of the $B_j(n_{j,1})$ were accepted. $B_j(n_{j,2})$ consists of all of $B_j(n_{j,1})$ and these new applicants. This is repeated until $n_{j,n} = 0$.

Assuming that organizations reveal their preferences, what they should submit is clear. For an applicant to be ranked above any cutoff, the organization must find the value for the applicant to be higher than the cost of any slots in that cutoff. Therefore, $B_j(n_{j,1})$ is also the set of applicants such that $Z_j(a_i) > MC_j(n_{j,1})$. This is the top candidates such that the lowest ranked applicant still covers the cost at the margin. Below $n_{j,1}$ and above $n_{j,2}$ would be the applicants that cover the margin at the second cutoff but not the first one. $B_j(n_{j,2})$ is the set such that $Z_j(a_i) > MC_j(n_{j,2})$ which includes $B_j(n_{j,1})$ and these new applicants. This logic is repeated for all cutoffs up to $n_{j,n} = 0$.

For example, using the valuations from our first example, each organization has the following costs and applicant values: $Z_j(a_1) = 5$, $Z_j(a_2) = 4$, $Z_j(a_3) = 3$, $MC_j(1) = 2$, $MC_j(2) = 3.5$, $MC_j(3) = 7$. Creating the best possible list for o_1 and o_2 results in $n_1 = n_2 = 2$. This is because the best outcome for both is to be matched with a_1 and a_2. Here both o_1 and o_2 would take both a_1 and a_2 if they had to pay the marginal cost in slot 2 to match with them. So far we have [a_1, a_2, 2, ..., 1]. Next, we check for slot n-1, which in this case is 1. Both organizations would accept all three candidates if they only had to pay the marginal cost of slot 1. Therefore, the ROCL for o_1 and o_2 would be written as [a_1, a_2, 2, a_3, 1].

3.1.2 Algorithm

Using the notation from the GS algorithm, all applicants propose to the organization at the top of their ROL. Then, every o_j looks at their lowest value applicant a_k that proposed to them and checks if a_k is acceptable in slot s_j by looking at o_j's ROCL. If a_k ranks lower than s_j, o_j rejects a_k and o_j is removed from a_k's ROL. All applicants are tentatively accepted if a_k ranks higher than s_j. If there is an a_k such that a_k is unmatched and has any o_k remaining in their ROL, they propose to their top remaining organization, and so forth. To illustrate this, we use the applicant valuations $V_1(o_1) > V_1(o_2)$, $V_2(o_1) > V_2(o_2)$, $V_3(o_1) > V_3(o_2)$ and 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 depicted below in table 5.

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

Applicants first proposal

Looking at the ROCL of o_1, [a_1, a_2,2,a_3,1], we eliminate the lowest ranking applicant a_3, and a_1 and a_2 are tentatively accepted. After being rejected by o_1, a_3 proposes to o_2, who accepts them since a_3 was ranked if there is only one slot to fill for o_2.

Theorem 1.1: ENPAP results in a stable match

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 $V_i(\emptyset) < V_i(o_k)$. Therefore, regardless of whether a_i is being matched with no one or 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. For this to happen, since a_i applies to their highest ranked organization to their lowest ranked organization, a_i must have stopped when matched with o_u ranked higher than o_j such that $V_i(o_u) > V_i(o_j)$.

(ii) o_j rejected a_i. If $Z_j(a_i) < Z_j(\emptyset)$, then the algorithm cannot make a match where $Z_j(a_i) > Z_j(a_u)$. Since the algorithm only rejects the lowest ranked applicants, all other applicants tentatively accepted in the organization at the time must have ranked higher than a_i and $MC_j(s) > Z_j(a_i)$ where s in the number of tentatively accepted applicants. For o_j to still want a_i compared to one of the applicants they were matched with, someone ranked even lower than a_i must have been accepted later. If a_u ranks first to s among A_j, it follows that a_u must be ranked above at least one other a_k that was tentatively accepted while a_i was rejected, meaning $Z_j(a_u) > Z_j(a_k) > Z_j(a_i)$. If a_u was tentatively accepted with s or higher slots, then $Z_j(a_u) > MC_j(s) > Z_j(a_i)$. This would mean that in either case, a blocking pair does not exist as $Z_j(a_i) > Z_j(a_u)$ is false. Q.E.D.

Theorem 1.2: ENPAP results in a slot stable match

Proof: Assume ENPAP results in slot instability, then by definition $\exists a_k, o_j$ such that

(1) $Z_j(A_j) - C_j(s_j) < Z_j(A_j \cup a_i) - C_j(s_j + 1)$ and $V_i(o_u) < V_i(o_j)$,

or

(2) $Z_j(A_j) - C_j(s_j) < Z_j(A_j \setminus a_i) - C_j(s_j - 1)$.

(1) If the first inequality is true, then $\exists a_i$ such that $MC(s_j + 1) < Z_j(a_i)$ that ranks worse than all the other tentatively accepted applicants or $\exists a_i, a_u$ 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 $V_i(o_u) < V_i(o_j)$, then a_i would have already been

matched with o_j as a_i would have proposed to o_j before o_u and not be rejected. For the second case, if $V_i(o_u) < V_i(o_j)$, the match would have been unstable, which is not possible from Theorem 1.1.

(2) If the second inequality is true, $\exists a_k$, that is the lowest value $a_i \in A_j$ matched together such that $MC_j(s_j) > Z_i(a_k)$. However, the ENPAP algorithm rejects all a_i that do not satisfy $MC_j(s_j) < Z_j(a_i)$. Since a_k was not rejected by the algorithm, then $MC_j(s_j) < Z_j(a_k)$ must be true.

Since the algorithm cannot produce a match that satisfies either condition, the ENPAP must give a slot stable match. Q.E.D.

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

Theorem 1.3: ENPAP results in an applicant optimal match

Proof: Using induction and Theorem 1.1, assume that the algorithm does not give an applicant optimal match. That would mean that there exists an applicant a_i that could match with a better organization that did not. Since this is applicant proposing, assume that no applicant has yet been rejected by an organization that is achievable for them. This means that no o_j has rejected any a_i where there exists a stable, slot stable match with a_i matched to o_j. If a_i was rejected for being unacceptable, it is unachievable. If a_i was rejected in favor of a_k, then it is known that the applicant a_k prefers the organization o_u except for those that already rejected them. By the inductive assumption, those organizations are unachievable to a_k. If we consider a hypothetical matching that matches a_i to the o_u and everyone else to an achievable organization, a_k would prefer the o_u and vice versa, making it an unstable match. Q.E.D.

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

3.2.1 Inputs

We will be using the same inputs of the ROLs and ROCLs as the ENPAP algorithm described in section 3.1.1.

3.2.2 Algorithm

Step 1: Each organization proposes to their top $n_{j,1}$
Step 2: Each a_i chooses their most preferred o_j among those that proposed to a_i. For all o_j not chosen by a_i, a_i is removed from their ROCL.
Step 3: Organizations then propose to the top applicants on their lists that satisfy the cutoff

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

To illustrate this we use the valuations from before with applicant values resulting $V_1(o_2) > V_1(o_1)$, $V_2(o_2) > V_2(o_1)$, $V_3(o_1) > V_3(o_2)$, and organization values for both organizations leading to their respective ROCLs being [a_1, a_2, 2, a_3, 1]. First, each o_j submits their optimal organization list, shown below in table 6.

o_1	o_2
a_1	a_1
a_2	a_2

Organization Proposing First List

Since both a_1 and a_2 have been proposed to by 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 below in table 7.

o_1	o_2
a_3	a_1
$\emptyset$	a_2

Organization Proposing Match

Theorem 2.1: ENPOP results in a stable match
Proof: Assume that there is a blocking pair a_i and o_j. For this to happen, o_j must have put an applicant on their optimal list that is worse than a_i, a_u, in order for $Z_j(a_i) > Z_j(a_u)$ to be satisfied. By optimal list construction, this can only occur if a_i is unavailable. This only happens when $V_i(o_u) > V_i(o_j)$ or $V_i(\emptyset) > V_i(o_j)$ is satisfied. This violates $V_i(o_j) > V_i(o_u)$ therefore, ENPOP must result in a stable match. Q.E.D.

Theorem 2.2: ENPOP results in a slot stable match
Proof: For there to be slot instability, there $\exists a_i, o_j$ such that either
(1) $Z_j(A_j) - C_j(s_j) < Z_j(A_j \cup a_i) - C_j(s_j + 1)$ and $V_i(o_u) < V_i(o_j)$, or
(2) $Z_j(A_j) - C_j(s_j) < Z_j(A_j \setminus a_i) - C_j(s_j - 1)$.

(1) If the first inequality is true, then $\exists a_i$ such that $MC(s_j + 1) < Z_j(a_i)$ that ranks worse than all the other tentatively accepted applicants or an $\exists a_i, a_u$ 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 a_i wanted to go to that o_j more than their current match o_u, they would have been already matched as a_i would be qualified to be put

on o_j's optimal list and accept the offer. For the second case, if $V_i(o_u) < V_i(o_j)$ the match would have been unstable, which violates Theorem 2.1.
(2) If the second inequality is true, $\exists a_i, o_j$ matched together such that $MC_j(s_j) > Z_j(a_k)$ However, for that applicant to have been put into o_j's optimal list with s_j slots, o_j must have ranked before s_j in their ROL meaning $MC_j(s_j) < Z_j(a_k)$.

Since neither inequality can be true, ENPOP must give a slot stable match. Q.E.D

Theorem 2.3 ENPOP results in an organization optimal match
Proof: Using induction and Theorem 2.1, let's assume that the algorithm does not give an organization optimal match. That would mean that there exists an applicant a_i that was matched with an organization higher than their worst achievable organization. Since this is organization proposing assume that no applicant has yet rejected an organization that is achievable for him. This means that no a_i has rejected any o_j where there exists a stable, slot stable match with a_i matched to o_j. If a_i rejected an organization for being unacceptable, it's unachievable. If a_i rejected o_u in favor of o_j, we know that the organization o_j has the applicant in their optimal list except for those that already rejected them, and by the inductive assumption, those applicants are unachievable to o_j. If we consider a hypothetical matching that matches a_i to o_u and everyone else to an achievable organization, a_i would prefer o_j and o_j would prefer a_i over at least one other a_k from o_j's more constrained optimal list making it an unstable match which violates Theorem 2.1. Q.E.D

3.2.3 Wait list Comparison

The ENPOP algorithm closely resembles the wait list systems we see in places like graduate school admissions. Initially, each applicant submits applications to all organizations based on their ROL. Subsequently, each organization selects their top candidates, taking into consideration the trade-off between marginal costs and the applicant's preferences.

Following this, each applicant chooses the organization that provides them with the highest value among those who have accepted them. The process then repeats itself, with each organization once again selecting their preferred candidates, who are likely to accept their offers. In this context, the wait list comprises individuals whom the organization would consider if more preferred applicants declined their offers to match with that organization.

Both the ENPOP algorithm and the current wait list system enable organizations to fill vacancies left by applicants who choose another organization. However, in the wait list, system stability can be compromised by both early acceptances and deadline related decisions.

Let's examine the scenario of early acceptances. When an applicant, denoted as a_i, accepts an early offer from organization o_j, there are two possible scenarios to consider in their ROL. First, if there is no organization o_u ranked above o_j in a_i's ROL, it reflects a_i's alignment with the ENPOP framework, as they have secured their best match and have no incentive to deviate.

However, if such an organization o_u exists in a_i's ROL, a potential exists for o_u to extend an offer to a_i. However, if a_i has already accepted o_j's offer, they may be unable to switch to their preferred organization. This situation could lead to an unstable outcome.

Furthermore, we must consider the impact of deadline acceptances. Let's consider two organizations, o_1 and o_2, both of which have sent acceptances to a_1, a_2, and placed a_3 on their respective wait lists. If both a_1 and a_2 delay their decisions until the last possible moment to choose o_1, there may not be enough time for o_2 to send an acceptance offer to a_3 from the wait list, leaving insufficient time for a_3 to decide. This dynamic introduces potential instability not observed in the ENPOP or ENPAP frameworks.

3.3 Unique Set of Slot Stable Filled Slots

Next, we show that with costly slots, there is only one set of filled slots that result in a slot stable match. This uniqueness property highlights the improbable nature of organizations setting fixed quotas where each organization accepts the exact number of applicants needed to have stability in this environment.

We transform the many-to-one match into a one-to-one match by using the applicant's a_i ROLs and the organization's o_j ROCLs. For organization o_j, they offer 1,2,..., n_j slots. Let $o_{j,x}$ denote o_j's xth slot. For all slots $o_{j,x}$, their ROL is defined as the cutoff list $B_j(x)$. Thus, each slot an organization offers now has its own ROL, which is considered a "different" organization for the 1-1 match. For every applicant a_i, we set their one-to-one ROLs such that organization o_j's xth slot $o_{j,x}$ is ranked above organization o_k's yth slot $o_{k,y}$ if and only if a_i ranks o_j above o_k in their many-to-one ROL. In addition, organization o_j's x^{th} slot, $o_{j,x}$, is ranked above organization o_j's y^{th} slot, $o_{j,y}$, by a_i if and only if $x > y$.

Lemma 1: An ordered many-to-one match is stable and slot stable if and only if the transformed one-to-one match is stable

Proof: Here, and ordered many-to-one match is one such that o_j has their most preferred applicant in A_j in slot one, their second most preferred in slot two, and so forth. For this to be true, three properties must be true: if the corresponding one-to-one match is stable, then the many-to-one match is stable; if the one-to-one match is stable, the many-to-one match is slot stable; if the many-to-one match is stable and slot stable, the one-to-one match is stable.

First, we will show that if the many-to-one match is stable, then the transformed one-to-one match will be stable. We prove this by contradiction. 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, that 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.

Next, we will show that if the transformed one-to-one match is stable, then the many-to-one match is slot stable. We will prove this by contradiction. Suppose the one-to-one match is stable while the many-to-one match is slot unstable, then by definition there exists applicant a_i and organization o_j such that

$$(1)\quad MC_j(s_j+1) < Z_j(a_i),\quad a_i \notin A_j,\ a_i \in A_u,\ V_i(o_j) > V_i(o_u)$$

or

$$(2)\quad Z_j(a_i) < MC_j(s_j),\quad a_i \in A_j$$

1) If the first inequality is true, then there exists an applicant a_i such that $MC(s_j+1) < Z_j(a_i)$ and ranks below all the other tentatively accepted applicants or there exists 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 $o_{u,x}$ over $o_{j,y}$. 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 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 there current applicant a_u. Since both $o_{j,y}$ and a_i prefer each other over their current match, the one-to-one match is unstable.

2) If the second inequality is true, there exists 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 their ROL. This would make the one-to-one match unstable as it is not individually rational due to o_{j,s_j} preferring not being matched at all.

Lastly, we will show that if the many-to-one match is stable and slot stable, then the transformed one-to-one match will be stable. We will prove this with contradiction. Suppose the many-to-one match is stable and slot stable while the many-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 prefers $o_{j,y}$ prefer each other over their current match. This can happen either 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 is ranked above o_k. Since o_j ranks applicants in the same order as its slots, then o_j must prefer a_i over the applicant a_u who is in slot $o_{j,y}$. Since o_j prefers a_i over a_u and a_i prefers o_j over o_k. Then the many-to-one match must be unstable.

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

Therefore an ordered many-to-one match is stable and slot stable if and only if the corresponding one-to-one match is stable. Q.E.D.

Theorem 3: 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 these constructed ROLs result in both stable and slot stable outcomes as with our algorithm. For this and any one-to-one match with strict preferences, Roth and Sotomayor [6] have shown that the set of unassigned agents (applicants and organizations) is the same for all stable matches. Therefore, the same slots must be matched for a corresponding one-to-one match to be stable. Hence, since the set of matched slots is always the same, and the many-to-one match is stable and slot stable, each organization must have the same number of applicants for all stable, slot stable matches. Q.E.D.

4 Conclusion

We have successfully developed a new matching algorithm that incorporates the cost of supplying slots to be assigned to applicants by building upon the principles of the original GS algorithm. This new algorithm ensures stable outcomes by incorporating cutoff points in ROLs to account for the cost of supplying slots. Additionally, given the nature of the environment with costly slots, we have defined the requirement for our algorithm to be slot stable and shown the improbability of this occurring endogenously in the current system. This new concept requires organizations not to be incentivized to change their number of available slots unilaterally. We have also shown that our algorithm is comparable to the current wait list system used in college and graduate school admissions when looking at school concerns. Yet, it removes the possibility of potentially preemptive behavior that can lead to unstable matches.

5 Bibliography

[1] D. Gale, L. S. Shapley, College Admissions and the Stability of Marriage. *The American Mathematical Monthly*, 69(1), 9–15 (1962).

[2] D. Abraham, R. Irving, D. Manlove, Two algorithms for the Student-Project Allocation problem. *Journal of Discrete Algorithms*, Volume 5, Issue 1, 73-90 (2007).

[3] A. Roth, E. Peranson, The Redesign of the Matching Market for American Physicians: Some Engineering Aspects of Economic Design. *American Economic Review*, 89 (4): 748-780 (1999).

[4] I. Rios, T. Larroucau, Tomas, G. Parra, R. Cominetti, College Admissions Problem with Ties and Flexible Quotas. *SSRN Electronic Journal* (2014).

[5] G. Limaye, M. Nasre, Stable Matchings with Flexible Quotas. *ArXiv.* /abs/2101.04425 (2021)

[6] Roth, A. E., and M. Sotomayor. Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis. Econometric Society Monographs. Cambridge University Press, 1990.