**T.C. KOCAELİ ÜNİVERSİTESİ**
**SOSYAL BİLİMLER ENSTİTÜSÜ**
**İKTİSAT ANABİLİM DALI**
**İKTİSAT TEORİSİ VE TARİHİ BİLİM DALI**

# ÖĞRENCİ-OKUL EŞLEŞME PİYASASI ANALİZİ VE EŞLEŞME PİYASASI MEKANİZMALARI ALGORİTMALARI

**YÜKSEK LİSANS TEZİ**

**Uğur ÇUHALILAR**

**KOCAELİ 2021**

**T.C. KOCAELİ ÜNİVERSİTESİ**
**SOSYAL BİLİMLER ENSTİTÜSÜ**
**İKTİSAT ANABİLİM DALI**
**İKTİSAT TEORİSİ VE TARİHİ BİLİM DALI**

# ÖĞRENCİ-OKUL EŞLEŞME PİYASASI ANALİZİ ve EŞLEŞME PİYASASI MEKANİZMALARI ALGORİTMALARI

**YÜKSEK LİSANS TEZİ**

**Uğur ÇUHALILAR**

**Prof. Dr. Şevket Alper KOÇ**

**KOCAELİ 2021**

**T.C. KOCAELİ ÜNİVERSİTESİ**
**SOSYAL BİLİMLER ENSTİTÜSÜ**
**İKTİSAT ANABİLİM DALI**
**İKTİSAT TEORİSİ VE TARİHİ BİLİM DALI**

# ÖĞRENCİ-OKUL EŞLEŞME PİYASASI ANALİZİ ve EŞLEŞME PİYASASI MEKANİZMALARI ALGORİTMALARI

**YÜKSEK LİSANS TEZİ**

**Tezi Hazırlayan: Uğur ÇUHALILAR**

**Tezin Kabul Edildiği Enstitü Yönetim Kurulu Karar ve No:16.06.2021/14**

**Jüri Başkanı: Prof. Dr. Şevket Alper KOÇ** [redacted]

**Jüri Üyesi: Prof. Dr. Yılmaz KILIÇASLAN** [redacted]

**Jüri Üyesi: Doç. Dr. Ferhat PEHLİVANOĞLU** [redacted]

**KOCAELİ 2021**

# ÖNSÖZ

"Öğrenci-Okul Eşleşme Piyasası Analizi ve Eşleşme Piyasası Mekanizmaları Algoritmaları" isimli yüksek lisans tez çalışmasında bana yol gösteren ve hem manevi hem akademik anlamda desteğini hiç esirgemeyen danışman hocam Prof. Dr. Şevket Alper KOÇ'a; çalışmamda yardımlarını esirgemeyen ve çokça emeği bulunan Arş. Gör. M. Rıdvan İNCE'ye ve her daim yanımda bulunan aile üyelerim Özkan ÇUHALILAR, Nesrin ÇUHALILAR ve Onur ÇUHALILAR'a çok teşekkür ederim. Ayrıca lisans ve lisansüstü eğitimimde bilgilerini ve tecrübelerini benimle paylaşan Kocaeli Üniversitesi İktisat Bölümü hocalarıma teşekkürü borç bilirim.

# İÇİNDEKİLER

## İKİNCİ BÖLÜM



## ÜÇÜNCÜ BÖLÜM

**ÖZET**

Piyasalarda paylaşım ve eşleşme sorununa bir çözüm yolu getiren Eşleşme Teorisi, tek taraflı ve çift taraflı piyasalarda en etkin eşleşmelerin nasıl yapılabileceğine dair teorileri ortaya koymuş ve hala koymaktadır. Bu teorilerin en önemli argümanı Gale ve Shapley (1962) çalışmasıdır ve bu çalışmada Gale ve Shapley'nin açıkladıkları gecikmeli kabul algoritması piyasalarda etkin mekanizmaların oluşturulmasında yol gösterici olmuştur.

Piyasalarda eşleşme sorununun en önemli örneklerinden olan ve çalışmamızda detaylı olarak incelemeye aldığımız problemler Üniversiteye Giriş Problemi ve Okul Seçimi Problemi'dir. Aynı gibi gözüken bu problemler aslında birbirinden ayrışmaktadırlar. Bu iki problem arasındaki farklar, problemlerin çözülebilmesi için yazılan mekanizmaların birbirinden farklılaşmasındaki temel noktadır. Çoğu oluşturulan mekanizma özellik olarak birbirinden farklıdır. Çalışmamızda her mekanizma ayrı olarak örnekler üzerinden incelenirken, her bir mekanizmanın özellikleri detaylıca açıklanmıştır. Üniversiteye girişte ve okul seçiminde kullanılan bu mekanizmaların birbirlerinden algoritma ve özellik olarak farklılaştıkları önemli noktalar gösterilmek adına bu çalışmada mekanizmaların aynı örnekler üzerinden uygulamaları yapılmış, detaylı olarak çözümleri gösterilmiş ve teori ile uyumlulukları değerlendirilmiştir.

Piyasalarda yapılacak eşleşmeler için geliştirilen mekanizmaların uygulanması teorik olarak kolay olsa da, oyuncu veya nesne sayısının fazla olduğu durumlarda mekanizmalar uygulanarak doğru eşleşmelerin yapılması sorun olabilmektedir. Bu sorunu gidermek adına bu çalışmada dört farklı mekanizmanın Python programlama dili ile yazılan algoritmaları geliştirilmiştir ve bu algoritmaların özellikleri ve algoritmalar ile çözümü yapılan örneklere yer verilmiştir. Örnek sonucunda oluşan farklı atama sonuçlarının değerlendirilmesinde "fayda endeksi" adıyla geliştirilen ve öğrenci faydasının hesaplanması temeline dayanan endeks kullanılmış olup, oluşan endeks değerlerinin değerlendirmelerine yer verilmiştir.

**Anahtar Kelimeler:** Eşleşme Teorisi, Üniversiteye Giriş Problemi, Okul Seçimi Problemi, Eşleşme Piyasası Mekanizmaları

**ABSTRACT**

Matching Theory, which brings a solution to the problem of sharing and matching in the markets, has put forward and still puts forward the theories on how to make the most effective matches in two-sided and one-sided markets. The most important argument of these theories is the study of Gale and Shapley (1962), and the deferred acceptance algorithm explained by Gale and Shapley in this study guided the creation of effective mechanisms in the markets.

One of the most important examples of the matching problem in the markets are the College Admissions Problem and the School Choice Problem, which we have examined in detail in our study. These problems, which appear to be the same, actually diverge from each other. The differences between the set of problems are the main point in the differentiation of the mechanisms written in order to solve the problems. Most of the mechanisms created are different from each other in terms of features In our study, while each mechanism is examined separately through examples, the features of each mechanism are explained in detail. In order to show the important points where these mechanisms used in college admissions and school choice differ from each other in terms of algorithm and features, in this study, the applications of the mechanisms were made on the same examples, their solutions were shown in detail and their compatibility with the theory was evaluated.

Although the implementation of the mechanisms developed for the matches to be made in the markets is theoretically easy, it may be a problem to make the correct matches by applying the mechanisms in cases where the number of players or objects is high. In order to solve this problem, algorithms written in Python software language of four different mechanisms have been developed in this study, and the features of these algorithms and solutions of the examples that solved by algorithms are given. In the evaluation of the different assignment results resulting from the example, the index developed under the name of "utility index" and based on the calculation of student utility was used, and the evaluations of the index values formed were included.

**Keywords:** Matching Theory, College Admissions Problem, School Choice Problem, Matching Market Mechanisms

## KISALTMALAR LİSTESİ

**KISALTMALAR**

KTEGKA : Kadınların Teklif Ettiği Gecikmeli Kabul Algoritması

ETEGKA : Erkeklerin Teklif Ettiği Gecikmeli Kabul Algoritması

TTC : En Yüksek Değiş Tokuş Döngüleri (Top Trading Cycles)

YRMH-IGYT : Sen benim Evimi İste - Ben Senin Sıranı Alırım

MCSD : Çok Kategorili Seri Diktatörlük (Multi-Category Serial Dictatorship)

ABD : Amerika Birleşik Devletleri

İTÜ : İstanbul Teknik Üniversitesi

ODTÜ : Orta Doğu Teknik Üniversitesi

ÜSYM : Üniversitelerarası Öğrenci Seçme ve Yerleştirme Merkezi

ÜSS : Üniversitelerarası Seçme ve Yerleştirme Sınavı

ÖSYM : Öğrenci Seçme ve Yerleştirme Merkezi

YÖK : Yüksek Öğretim Kurulu

ÖSS : Öğrenci Seçme Sınavı

ÖYS : Öğrenci Yerleştirme Sınavı

YDS : Yabancı Dil Sınavı

OBP : Ortaöğretim Başarı Puanı

YGS : Yükseköğretime Geçiş Sınavı

LYS : Lisans Yerleştirme Sınavı

AOBP : Ağırlıklı Ortaöğrenim Başarı Puanı

TÜBİTAK : Türkiye Bilimsel ve Teknolojik Araştırma Kurumu

EADAM : Etkinlik-Düzeltilmiş Gecikmeli Kabul Mekanizması

ÖSYS : Öğrenci Seçme ve Yerleştirme Sistemi

# TABLOLAR LİSTESİ

## ŞEKİLLER LİSTESİ

## GİRİŞ

Tarihsel süreç boyunca insanoğlunun en büyük problemlerinden biri verimlilik sorunu olmuştur. Kaynaklar nasıl dağıtılırsa en verimli dağılımın gerçekleşeceği, piyasalarda en verimli eşleşmelerin nasıl yapılacağı, örneğin işçi-işveren eşleşmesinin en verimli biçimde nasıl gerçekleştirileceği vb. sorunlar büyük bir problem olarak görülmekteydi. Bu problemlere bir çözüm aranması Eşleşme Teorisi'nin doğuşuna ve gelişimine sebebiyet vermiştir. Bu süreçte David Gale ve Lloyd S. Shapley, 1962 yılında yayınladıkları makale ile Eşleşme Teorisi'nin ilk modellemesini evlilik piyasası üzerinde kurmuşlardır. Bu gelişmeden sonra Eşleşme Teorisi'nin yükselişi başlamış olup günümüzde çok geniş uygulama alanlarına ulaşmayı başarmıştır.

Daha sonra gelişimine devam ettiği süreçte geniş bir piyasa yelpazesine sahip olan Eşleşme Teorisi gruplara ayrılmış olup, piyasalar özelliklerine göre iki taraflı eşleşme piyasaları ve tek taraflı eşleşme piyasaları olmak üzere iki ana gruba ayrılmıştır. İki farklı oyuncu grubunun olduğu piyasalar iki taraflı eşleşme piyasaları, biri nesne biri oyuncu (insan) olan iki farklı grubun olduğu piyasalar tek taraflı eşleşme piyasaları olarak kabul edilmiştir. Öğrenci-üniversite, işçi-firma eşleşmesi piyasaları iki taraflı eşleşme piyasalarına örnekken, öğrencilerin yurtlarındaki odalarına yerleştirilmesi gibi piyasalar tek taraflı eşleşme piyasalarına örnek olarak gösterilebilir. İki taraflı eşleşme piyasalarında, eğer bir gruptaki oyuncu karşı gruptaki oyunculardan en fazla biri ile eşleşebiliyorsa bu piyasalar birebir eşleşme piyasalarıdır. Gale ve Shapley'nin (1962), kurduğu evlilik modeli bu piyasalara en iyi örneklerdendir.

Gale ve Shapley (1962), çalışmalarında Eşleşme Teorisi'nin en önemli algoritmasını tanıtmış ve modelin çözümünü bu algoritma üzerinden gerçekleştirmişlerdir. Daha sonra Eşleşme Teorisi alanında geliştirilen birçok mekanizma temelinde *gecikmeli kabul algoritması* olarak adlandırılan bu algoritmayı kullanmıştır.

İki taraflı eşleşme piyasalarının bir diğer ayağı olan bire çok eşleşme piyasalarında bir oyuncu birden fazla oyuncu ile eşleşebilmektedir. Bu piyasalara en iyi örnek üniversiteye giriş piyasalarıdır. Bu piyasada üniversitelerin kontenjan kısıtı olması ve öğrencilerin birden fazla üniversite ile eşleşebilmeleri *üniversiteye giriş*

*problemi*ni doğurmuştur. Aynı şekilde tek taraflı eşleşme piyasalarına baktığımızda bu piyasalara bir örnek olan okul-öğrenci eşleşmesi piyasasında da bir sorun mevcuttur. *Okul seçimi problemi* olarak adlandırılan bu problem, öğrencilerin tercihlerinin olduğu ve okulların aslında bir nesne gibi görüldüğü fakat öğrenciler üzerinde bir öncelik sıralamasına sahip olduğu piyasada, öğrencilerin hangi okullara, hangi sırayla yerleştirileceği sorunudur. Üniversiteye giriş problemi ve okul seçimi probleminin varlığı yıllardır uzmanların bu konu üzerine çözüm aramasına ve farklı mekanizmalar geliştirmesine yol açmıştır.

Çalışmanın, Eşleşme Teorisi'ne dair daha fazla bilgi verilecek ve Gale ve Shapley'nin (1962) kurduğu evlilik modeli örneğinin çözümünün *gecikmeli kabul algoritması* tanıtıldıktan sonra yapılacağı birinci bölümünde, üniversiteye giriş problemi ve okul seçimi problemleri örnekler ile daha fazla tanıtılacak ve bu iki problemin karşılaştırılması yapılacaktır.

Çalışmanın ikinci bölümünde üniversiteye girişte ve okul seçiminde kullanılan mekanizmalar örnekler üzerinden tanıtılacak olup, mekanizmaların uygulanması sonucu oluşan atamaların, öğrenci faydası ve okul faydası üzerine etkilerine yer verilecektir. Ve yine bu bölümde Türkiye'de kullanılan mekanizmanın tarihsel gelişimi ve uygulama esasları anlatılacaktır. Mekanizmaların geniş çaplı incelemesinin ve teorik olarak özelliklerinin anlatılacağı üçüncü bölümde, aynı zamanda üç önemli mekanizma olan Gale-Shapley Öğrenci Optimal Durağan Mekanizması, Gale-Shapley Okul Optimal Durağan Mekanizması ve Çok Kategorili Seri Diktatörlük Mekanizmasının aynı örnek üzerinden çözümlü karşılaştırılması yapılacaktır. Bir başka örnek ile En Yüksek Değiş Tokuş Döngüleri Mekanizması ile Gale-Shapley Öğrenci Optimal Durağan Mekanizması karşılaştırılacaktır. Bu karşılaştırmalar sonrası her mekanizma için oluşan eşleşmelerin değerlendirmesine yer verilecektir.

Son olarak Dördüncü bölümde ise Python programlama dili geliştirilmiş olan dört farklı mekanizma algoritması üzerinden eşleşme mekanizmalarının atamalarının otomatik olarak yapıldığı çok katılımcılı okul-öğrenci ataması örnekleri sonuçlarına yer verilecektir. Aynı zamanda bu bölümde yüz öğrenci ve otuz okulun bulunduğu aynı okul-öğrenci piyasası örneği üzerinden Gale-Shapley Öğrenci Optimal Durağan Mekanizması, Gale-Shapley Okul Optimal Durağan Mekanizması ve Çok Kategorili

Seri Diktatörlük Mekanizması atama sonuçlarına yer verilecektir. Aynı örnek üzerinden, çalışmamızda geliştirilen ve atama sonuçlarının değerlendirilmesinde öğrenci faydası temeline dayanan "fayda endeksi" ile atama sonuçlarının fayda endeks değerlerine yer verilecek olup hesaplanan endeks değerleri değerlendirmesi yapılacaktır.

# BİRİNCİ BÖLÜM

## 1. EŞLEŞME TEORİSİ ve EŞLEŞME PİYASALARI

### 1.1 EŞLEŞME TEORİSİ

"Eşleştirme", özellikle tahsis edilecek kıt mallar heterojen ve bölünemez olduğunda: Örneğin kim hangi işte çalışıyor, hangi öğrenciler hangi okula gidiyor, kim hangi nakledilebilir organı alıyor vb., ekonominin kimin neyi alacağı sorusuna odaklanır (Sönmez ve Ünver, 2011: 783). Eşleşme teorisi ise oyun teorisi ve mekanizma tasarımı araçlarını kullanarak, bölünmez kaynakların dağıtımını, değişimini ve birbirleriyle eşleşmesini inceleyen mikro iktisadın son otuz yılda hızla gelişen bir alanıdır (Doğan, 2014: 380). Eşleştirme teorisi, oyun teorisi, sosyal seçim teorisi ve mekanizma tasarımının kesişme noktasında yer almaktadır (Sönmez ve Ünver, 2011: 783).

Tarihsel ve teorik olarak eşleştirme teorisinin öncü çalışması 1962 yılında David Gale ve Lloyd S. Shapey tarafından yayınlanan "College Admissions and Stability of Marriage''(Üniversite Kabulleri ve Evliliğin Durağanlığı) adlı makaledir. Bu makalede *durağanlığın* tanımını yapan David Gale ve Lloyd S. Shapley, kendi geliştirdikleri ve *gecikmeli kabul algoritması,* daha sonra Gale-Shapley algoritması olarak da anılmıştır, adını verdikleri algoritmayı tanıtmışlardır. Her iki taraflı eşleşme piyasasında bir durağan eşleşme olacağını *gecikmeli kabul algoritması*nı kullanarak biri üniversite-öğrenci eşleşmesi ve diğeri kadın-erkek (evlilik) eşleşmesi olan iki farklı iki taraflı eşleşme piyasası örneği üzerinden göstermişlerdir. Roth (1984), A.B.D.'de stajyer doktorları hastanelere yerleştiren ve 1950'lerde uygulanmaya başlayan algoritmanın, Gale ve Shapley (1962)'de önerilen gecikmeli kabul algoritmasının eşdeğer olduğunu göstermiştir, bu durum piyasa tasarımının, yani gerçek hayattaki problemleri çözmek için teoriye dayanarak mekanizmalar tasarlamanın önem kazanmasına neden olmuştur (Doğan, 2014: 380).

Bu gelişmelerden sonra, böbrek nakli, okul seçimi, üniversiteye öğrenci kabulü gibi eşleşme teorisi alanlarında birçok mekanizma geliştirilmiştir. Bunlardan bazıları, Roth v.d. (2004) tarafından böbrek değişimi üzerine tasarladığı mekanizma, Abdulkadiroglu ve Sönmez (2003) tarafından *okul seçimi problemi* üzerine önerdikleri mekanizmalardır.

## 1.2 EŞLEŞME PİYASALARI

### 1.2.1 İki Taraflı Eşleşme Piyasaları

Doğan'a (2014, s.380) göre "İki taraflı eşleşme piyasalarında birbirleriyle eşleşecek iki ayrık (kesişimleri boş küme olan) oyuncu kümesi (örneğin, üniversiteler ve öğrenciler, işçiler ve firmalar, hastaneler ve stajyer doktorlar) bulunmaktadır." Bu piyasalarda her iki tarafın birbirleri üzerine tercihleri bulunmaktadır. Örneğin bir üniversite-öğrenci eşleşme piyasasında her üniversitenin, her öğrenci için, her öğrencinin ise her üniversite için bir tercih sıralaması mevcuttur. İki taraflı eşleşme piyasası temelde iki gruba ayrılmaktadır. Bunlardan ilki birebir eşleşme piyasası ve ikincisi ise bire çok eşleşme piyasasıdır.

#### 1.2.1.1 Birebir Eşleşme Piyasaları

İki taraflı eşleşme piyasasında her oyuncu karşı oyuncu kümesinden sadece tek bir oyuncu ile eşleşebiliyorsa bu tarz piyasalar birebir eşleşme piyasası olarak adlandırılır. Bu tarz piyasalara en iyi örnek yukarıda bahsettiğimiz evlilik modelidir.

Bu modelde sadece erkeklerin bulunduğu bir küme ($M$=$\{m_1, m_2, m_3, \ldots. m_k\}$) ve sadece kadınların bulunduğu bir diğer küme ($W$=$\{w_1, w_2, w_3, \ldots., w_k\}$) bulunmaktadır. Erkekler kümesindeki her erkeğin, kadınlar kümesindeki bütün kadınlar için, kadınlar kümesindeki her kadının ise, erkekler kümesindeki bütün erkekler için bir tercih sıralaması ($P$) mevcuttur. Örneğin $P_{m_1} = w_1, w_3, w_2, w_0, ..., w_k$, $m_1$'in tercih sıralamasını gösterirken, burada $m_1$'in ilk tercihi $w_1$, ikinci tercihi $w_3$ ve üçüncü tercihi $w_2$'dir. Benzer şekilde $P_{w_1} = m_2, m_1, m_3, ..., m_0$ ise $w_1$'in ilk tercihi $m_2$, ikinci tercihi $m_1$ ve üçüncü tercihi $m_3$'dür. Herhangi bir $m \in M$, herhangi bir $w \in W$ ile, herhangi bir $w \in W$ ise herhangi bir $m \in M$ ile evlenmek yerine hiç evlenmemeyi tercih edebilir. $P_{m_1} = w_1, w_3, w_2, w_0, ..., w_k$ sıralamasına göre $m_1$'in üçüncü tercihi olan $w_2$, $m_1$'in evlenmeye razı olduğu son tercihidir. Aynı şekilde $P_{w_1} = m_2, m_1, m_3, m_0 ..., m_k$ sıralamasına göre $w_1$'in üçüncü tercihi olan $m_3$, $w_1$'in evlenmeye razı olduğu son tercihidir. $w_3 \psi_{m_1} w_2$ ifadesi ise $m_1$'in $w_3$'ü $w_2$'ye tercih ettiğini belirtmektedir. Bir eşleşme $\mu$'de eğer $w_1$, $m_1$ ile eşleşmiş ise $\mu(w_1) = m_1$'dir. $\mu(w_1) = m_1$ ise $R_{\mu(w_1)} = 2$'dir. Yani $R_{\mu(w_1)}$, $m_1$'in kaçıncı tercihi ile eşleştiğini göstermektedir.

Bu birebir eşleşme modelinde, eşleşmeler yapılırken dikkate alınması gereken kavramlar vardır. Bunlardan ilki *bireysel rasyonellik*tir. Evlilik modeline göre

söylemek gerekirse bir $\mu$ eşleşmesinde eğer herhangi bir oyuncu eşleştiği eşini hiç evlenmemeye tercih ediyorsa $\mu$ eşleşmesi bireysel rasyoneldir. Aynı zamanda herhangi bir kadın-erkek çifti ($m$, $w$) birbirlerini bir $\mu$ eşleşmesi sonucunda eşleştikleri eşlerine tercih etmemeleri gerekmektedir. Eğer böyle bir durum var ise bu çift $\mu$ eşleşmesini bloke eder. Dikkate alınması gereken en önemli kavramlardan biri *durağanlık*tır. Bir $\mu$ eşleşmesinin durağan olabilmesi için, bireysel rasyonel ve hiçbir çift tarafından bloke edilmiyor olması gerekir.(Doğan, 2014: 382). Bu eşleşme modelinin eşleşmesinde önemli olan bir diğer kavram ise *pareto etkinlik* kavramıdır. Hiç bir oyuncuyu $\mu$ eşleşmesinde eşleştiği eşinden daha az tercih ettiği bir eş ile eşleştirmeden, en azından bir oyuncuyu $\mu$ eşleşmesindeki eşinden daha fazla tercih ettiği bir eş ile eşleştirebilen bir eşleşme yok ise $\mu$ eşleşmesi Pareto etkindir.

Gale ve Shapley (1962), gecikmeli kabul algoritmasını kullanarak evlilik modelinde durağan bir eşleşme olduğunu göstermişlerdir.

#### 1.2.1.1.1 Gale-Shapley Algoritması

Gale-Shapley algoritmasının temelinde *gecikmeli kabul algoritması* vardır. Gale ve Shapley (1962), durağan bir eşleşme bulmak için gecikmeli kabul algoritmasını tasarlamıştır (Economic Sciences Prize Committee of the Royal Swedish Academy of Sciences, 2012: 9). Gecikmeli kabul algoritmasında, eşleşmenin bir tarafındaki oyuncular diğer taraftaki oyunculara önceliklerine göre teklifte bulunurlar, teklif alan her oyuncu teklifler arasındaki en öncelikli tercihini tutar (bekletir) ve diğer teklifleri reddeder, burada önemli olan nokta oyuncuların gelen teklifleri hemen kabul etmeyip bekletmesidir. (Economic Sciences Prize Committee of the Royal Swedish Academy of Sciences, 2012, s.9). Prosedür, teklifte bulunan hiçbir oyuncu başka bir teklif yapmak istemeyene kadar devam eder, bu sırada oyuncular ellerinde tuttukları önerileri nihayet kabul eder (Economic Sciences Prize Committee of the Royal Swedish Academy of Sciences, 2012, s. 9). Gale ve Shapley (1962), ertelenmiş kabul algoritmasının durağan olduğunu, yani her zaman durağan bir eşleşme ürettiğini kanıtlamıştır (Economic Sciences Prize Committee of the Royal Swedish Academy of Sciences, 2012: 10).

Gale ve Shapley (1962), evlilik modelinin çözümünde gecikmeli kabul algoritmasını iki farklı uygulama yöntemi ile kullanmışlardır. Bunlar; kadınların

teklif ettiği gecikmeli kabul algoritması ve erkeklerin teklif ettiği gecikmeli kabul algoritmasıdır.

*Kadınların teklif ettiği gecikmeli kabul algoritması:* Algoritmanın bu yönteminde teklifi yapan taraf kadınlardır. İlk turda her kadın ilk tercihine teklifte bulunur. Her erkek aldığı teklifler arasında en çok tercih ettiği kadından gelen teklifi geçici olarak kabul eder ve diğer teklifleri reddeder. İkinci turda teklifleri reddedilen kadınlar bir sonraki tercihine teklif götürürler. Teklif alan her erkek, eğer ilk turda teklif aldıysa birinci ve ikinci turda aldıkları teklifler arasından en çok tercih ettiği teklifi geçici olarak kabul eder ve diğer teklifleri reddeder. Eğer ilk turda teklif almamış ise ikinci turda aldığı teklifler arasından tercihini yapar. Tek teklif aldıysa bu teklifi geçici olarak elinde tutar. Genel olarak *k*. turda, *k-1*. turda reddedilen her kadın, daha önce teklif götürmedikleri erkekler arasından en çok tercih ettiği erkeğe teklif götürür. *k*. turda teklif alan her erkek, *k-1*. adımda geçici olarak elinde bulundurduğu teklif var ise bu teklif ile birlikte *k*. turda aldığı teklif ve ya teklifler arasından en çok tercih ettiği kadının teklifini kabul eder. Algoritma kadınlardan hiçbirinin teklifinin reddedilmediği turda sona erer. Son turda bir kadının teklifini geçici olarak elinde bulunduran bir erkek var ise o erkek o kadınla eşleşir. Eğer bir kadının teklifini geçici olarak elinde bulunduran bir erkek yok ise o kadın hiçbir erkek ile eşleşemez.

*Erkeklerin teklif ettiği gecikmeli kabul algoritması:* Bu türde ise teklif yapan taraf erkekler ve teklifleri değerlendiren taraf kadınlardır. Dolayısıyla kadınların teklif ettiği gecikmeli kabul algoritmasındaki işleniş kadınlar ve erkekler yer değiştirerek aynı şekilde uygulanır.

Kadınların teklif ettiği gecikmeli kabul algoritmasında, kadınlar kümesi kendileri için en iyi (optimal) eşleşmeleri sağlarken, erkekler kümesi için en kötü eşleşmeler yapılır. Erkeklerin teklif ettiği gecikmeli kabul algoritmasında ise erkek optimal bir eşleşme söz konusudur. Bu algoritma sonucunda oluşan eşleşme ise kadınlar için en kötü eşleşmeleri doğurur. Optimal olmanın tanımını yapmak gerekirse, her başvuru sahibi en azından başka herhangi bir durağan atamada olduğu kadar iyi durumda ise durağan bir atama *optimal* olarak adlandırılır (Gale ve Shapley, 1962: 10).

Dolayısıyla hiçbir kadın başka herhangi bir durağan eşleşmeyi kadınların teklif ettiği gecikmeli kabul algoritmasında oluşan eşleşmeye tercih etmezken, hiçbir erkek de başka herhangi bir durağan eşleşmeyi erkeklerin teklif ettiği gecikmeli kabul algoritmasında oluşan eşleşmeye tercih etmez.

Algoritmanın uygulanması ve modelin çözümünü örnek üzerinden gösterelim.

*Örnek 1*: Dört kadın $W=\{w_1, w_2, w_3, w_4\}$ ve 4 erkeğin $M=\{m_1, m_2, m_3, m_4\}$ bulunduğu bir evlilik modeli olsun.

*tercihler:*

*kadınlar:*

$P_{w_1} = m_4, m_3, m_1, m_0, m_2$ $\quad$ $P_{w_3} = m_3, m_2, m_1, m_4, m_0$

$P_{w_2} = m_3, m_1, m_2, m_4, m_0$ $\quad$ $P_{w_4} = m_1, m_4, m_3, m_2, m_0$

*erkekler:*

$P_{m_1} = w_3, w_2, w_4, w_1, w_0$ $\quad$ $P_{m_3} = w_1, w_4, w_2, w_0, w_3$

$P_{m_2} = w_3, w_1, w_4, w_2, w_0$ $\quad$ $P_{m_4} = w_2, w_1, w_3, w_4, w_0$

*Kadınların teklif ettiği gecikmeli kabul algoritmasına göre çözüm:*

Birinci turda her kadın ilk tercihine teklifte bulunacak ve durum Şekil 1'deki gibi olacaktır:

**Şekil 1:** KTEGKA'na Göre Örnek 1 – 1. Tur

| | $m_1$ | $m_2$ | $m_3$ | $m_4$ |
|---|---|---|---|---|
| *1. tur* | $w_4$ | | $w_2, w_3$ | $w_1$ |

Bu durumda $w_1$, $m_4$'e, $w_4$, $m_1$'e, $w_2$ ve $w_3$ ise $m_3$'e teklif yapacaktır. İki teklif alan $m_3$, kendi tercih sıralamasında $P_{m3}= w_1, w_4, w_2, w_0, w_3$ daha üst sırada bulunan $w_2$'nin teklifini geçici olarak kabul eder ve $w_3$'ün teklifini reddeder. Zaten $m_3$, hiç evlenmemeyi $w_3$ ile evlenmeye tercih etmektedir. Diğer erkekler aldıkları teklifleri geçici olarak kabul ederler. Bir sonraki turda teklifi reddedilen $w_3$ ikinci sıradaki tercihine yani $m_2$'ye teklif götürür. İkinci turda birden fazla teklif alan bir erkek ve

teklifi reddedilen bir kadın bulunmadığı için geçici olan olarak kabul edilen her teklif nihai olarak belirlenir.

**Şekil 2:** KTEGKA'na Göre Örnek 1 – Nihai Atamalar

| | $m_1$ | $m_2$ | $m_3$ | $m_4$ |
|---|---|---|---|---|
| *1. tur* | $w_4$ | | $w_2$, $w_3$ | $w_1$ |
| *2. tur* | $w_4$ | $w_3$ | $w_2$ | $w_1$ |

Algoritma sonunda aşağıda gösterilen kadın optimal durağan eşleşme oluşur.

$$\mu_{kadın\text{-}optimal} = \begin{pmatrix} w_1 & w_2 & w_3 & w_4 \\ m_4 & m_3 & m_2 & m_1 \end{pmatrix}$$

$\mu_{kadın\text{-}optimal}$ eşleşmesine göre:

$\mu(w_1)= m_4$, $\mu(w_2)= m_3$, $\mu(w_3)= m_2$, $\mu(w_4)= m_1$,

$R_{\mu(w_1)} = 1$, $R_{\mu(w_2)}= 1$, $R_{\mu(w_3)}= 2$, $R_{\mu(w_4)}= 1$ ve $R_{\mu(m_1)}= 3$, $R_{\mu(m_2)}= 1$, $R_{\mu(m_3)}= 3$, $R_{\mu(m_4)}= 2$'dir.

*Erkeklerin teklif ettiği gecikmeli kabul algoritmasına göre çözüm:*

Birinci turda her erkek ilk tercihine teklifte bulunur. $m_3$, $w_1$'e, $m_4$, $w_2$'ye, $m_1$ ve $m_2$ ise $w_3$'e teklif götürür.

**Şekil 3:** ETEGKA'na Göre Örnek 1 – 1. Tur

| | $w_1$ | $w_2$ | $w_3$ | $w_4$ |
|---|---|---|---|---|
| *1. tur* | $m_3$ | $m_4$ | $m_1$, $m_2$ | |

İki teklif alan $w_3$, $m_2$'nin teklifini geçici olarak kabul eder ve $m_1$'in teklifini reddeder. Tek teklif alan diğer kadınlar kendilerine yapılan tekliflerini geçici olarak kabul ederler. İkinci turda teklifi reddedilen $m_1$, ikinci tercihi olan $w_2$'ye teklifte bulunur.

**Şekil 4**: ETEGKA'na Göre Örnek 1 – 1. ve 2. Tur

| | $w_1$ | $w_2$ | $w_3$ | $w_4$ |
|---|---|---|---|---|
| *1. tur* | $m_3$ | $m_4$ | $m_1$, $m_2$ | |
| *2. tur* | $m_3$ | $m_4$, $m_1$ | $m_2$ | |

İki teklif alan $w_2$, $m_1$'in teklifini geçici olarak kabul eder ve $m_4$'ün teklifini reddeder. Üçüncü turda, bir önceki turda teklifi reddedilen $m_4$, ikinci tercihi olan $w_1$'e teklifte bulunur ve $w_1$ bu teklifi geçici olarak kabul ederken daha önceki turlarda elinde bulundurduğu $m_3$'ün teklifini reddeder.

**Şekil 5:** ETEGKA'na Göre Örnek 1 – 1.– 2. ve 3. Tur

| | $w_1$ | $w_2$ | $w_3$ | $w_4$ |
|---|---|---|---|---|
| *1. tur* | $m_3$ | $m_4$ | $m_1, m_2$ | |
| *2. tur* | $m_3$ | $m_4, m_1$ | $m_2$ | |
| *3. tur* | $m_3, m_4$ | $m_1$ | $m_2$ | |

Dördüncü turda bir önceki turda teklifi reddedilen $m_3$, $w_4$'e teklifte bulunur. Elinde tek teklif bulunan $w_4$ bu teklifi geçici olarak kabul eder. Bu turda birden fazla teklif alan bir kadın ve teklifi reddedilen bir erkek bulunmadığı için geçici olan olarak kabul edilen teklifler nihai olarak belirlenir.

**Şekil 6**: ETEGKA'na Göre Örnek 1 – Nihai Atamalar

| | $w_1$ | $w_2$ | $w_3$ | $w_4$ |
|---|---|---|---|---|
| *1. tur* | $m_3$ | $m_4$ | $m_1, m_2$ | |
| *2. tur* | $m_3$ | $m_4, m_1$ | $m_2$ | |
| *3. tur* | $m_3, m_4$ | $m_1$ | $m_2$ | |
| *4. tur* | $m_4$ | $m_1$ | $m_2$ | $m_3$ |

Algoritmanın sonunda aşağıda gösterilen erkek-optimal durağan eşleşme oluşur.

$$\mu_{erkek\text{-}optimal} = \begin{pmatrix} m_1 & m_2 & m_3 & m_4 \\ w_2 & w_3 & w_4 & w_1 \end{pmatrix}$$

$\mu_{erkek\text{-}optimal}$ eşleşmesine göre:

$\mu(m_1)= w_2$, $\mu(m_2)= w_3$, $\mu(m_3)= w_4$, $\mu(m_4)= w_1$,

$R_{\mu(m_1)}= 2$, $R_{\mu(m_2)}= 1$, $R_{\mu(m_3)}= 2$, $R_{\mu(m_4)}= 2$ ve $R_{\mu(w_1)}= 1$, $R_{\mu(w_2)}= 2$, $R_{\mu(w_3)}= 2$, $R_{\mu(w_4)}= 3$'dür.

Bu mekanizmanın her evlilik piyasası eşleşmesinde durağan bir eşleşme meydana getireceğini daha önce belirtmiştik. Eğer bir mekanizma her evlilik problemi için Pareto etkin eşleşme yaratıyorsa Pareto etkin bir mekanizma, eğer her

evlilik problemi için bireysel rasyonel bir eşleşme yaratıyorsa bireysel rasyonel bir mekanizmadır (Doğan, 2014: 384-385). Bir mekanizmada hiçbir oyuncu tercihlerini olduğundan farklı göstererek daha iyi bir eşleşme elde edemiyorsa bu mekanizma manipüle edilemez (strategy proof) bir mekanizmadır. Yani eğer tercih belirtme oyununda, doğru tercihleri belirtmek, her oyuncu için zayıf baskın bir stratejiyse, bir mekanizma manipüle edilemezdir ( Sönmez ve Ünver, 2010: 829).

Roth'un (1982)'nin evlilik modeli için kanıtladığı teorilerinden biri şuydu: Tercihlerin doğru şekilde belirtilmesinin tüm oyuncular için baskın bir strateji olduğu genel eşleştirme problemi için durağan bir eşleştirme prosedürü mevcut değildir (Roth, 1982: 622). Roth'un (1982), kanıtladığı teori açıkça evlilik modelinde hem durağan hem de manipüle edilemez bir mekanizmanın var olamayacağını belirtmektedir. Aynı zamanda hem Pareto etkin hem bireysel rasyonel ve hem de manipüle edilemez bir mekanizma var olamaz (Alcalde ve Barbera, 1994; Aktaran Doğan, 2014: 385)

#### 1.2.1.2 Bire Çok Eşleşme Piyasası

Eğer bir oyuncu birden fazla oyuncu ile eşleşebiliyorsa bu piyasa bire çok eşleşme piyasası olarak adlandırılır. ABD'de stajyer doktor-hastane eşleşmeleri, firma-işçi eşleşmeleri bire çok eşleşme piyasasına örnek olarak gösterilebilir. Bire çok eşleşme piyasasına en iyi örnek ise üzerinde çokça tartışılan üniversiteye giriş problemidir.

##### 1.2.1.2.1 Üniversiteye Giriş Problemi

Üniversiteler ve öğrenciler stratejik oyuncular olarak kabul edilirse, çift taraflı bir eşleştirme sorunu ortaya çıkar (Economic Sciences Prize Committee of the Royal Swedish Academy of Sciences, 2012: 26). Bunun sonucunda Gale ve Shapley'nin (1962) tanıttığı "üniversiteye giriş problemi" ortaya çıkmaktadır. Gale ve Shapley (1962), üniversiteye giriş probleminin sebeplerini temel olarak şu şekilde belirtmiştir; (a) Belirli bir başvuru sahibinin başka bir yere de başvurup başvurmadığı bilinmeyebilir; eğer bu biliniyorsa, (b) başvurduğu okulları nasıl sıraladığı bilinmeyebilir; bu bilinse bile, (c) diğer okullardan hangisinin onu kabul etmeyi teklif edeceği bilinmeyecektir (Gale ve Shapley, 1962: 9). Üniversiteye giriş problemi evlilik problemi ile benzerlik göstermekle birlikte en büyük farkı evlilik probleminde her kadın bir erkek ve her erkek de bir kadın ile eşleşebiliyorken,

üniversiteye giriş probleminde üniversitelerin birden fazla öğrenci ile eşleşebilmektedir.

Üniversiteye giriş probleminde, evlilik modeline benzer olarak iki oyuncu kümesi mevcuttur. Bunlardan biri öğrenci kümesi $S=\{s_1, s_2, s_3, \ldots, s_n\}$, bir diğeri ise üniversitelerin bulunduğu $C=\{c_1, c_2, c_3, \ldots, c_m\}$ kümesidir. Her oyuncunun karşı kümedeki oyuncular için bir tercih sıralaması mevcuttur. Örneğin $s_1$ öğrencisinin tercih sıralaması $P_{s_1} = c_1, c_3, c_4, c_0, \ldots, c_m$ şeklinde gösterilebilir. $P_{s_1}$'e göre $s_1$'in ilk tercihi $c_1$, ikinci tercihi $c_3$, üçüncü ve son tercihi $c_4$'tür. $c_4$ üniversitesinden sonra herhangi bir üniversiteye gitmek yerine açıkta kalmayı tercih etmektedir. Öğrenciler bazı üniversiteleri tercih dışında bırakabilirler. Öğrencilerin bu tercihi, onların, o üniversiteye gitmek yerine açıkta kalmaya razı olduğu anlamına gelmektedir. $P_{c_1} = s_2, s_3, s_4, s_0, \ldots, s_n$ ise $c_1$'in tercih sıralamasını göstermektedir. $P_{c_1}$'e göre $c_1$'in ilk tercihi $s_2$, ikinci tercihi $s_3$, üçüncü ve son tercihi ise $s_4$'tür. $c_1$, $s_4$ tercihinden sonra herhangi bir öğrenciyi kabul etmek yerine kontenjanının boş kalmasını tercih etmektedir.

Üniversiteye giriş probleminde, her üniversitenin bir kontenjanı ($q$) mevcuttur. Örneğin $q_{c_1}$ , $c_1$'in kontenjan sayısını göstermektedir. $q$ ise toplam kontenjan sayısını göstermektedir. Eğer bir eşleşmede $C=\{c_1, c_2, c_3, c_4\}$ ise $q = q_{c_1} + q_{c_2} + q_{c_3} + q_{c_4}$'dür. Problem $n$ adet öğrencinin $m$ adet üniversiteye yukarıdaki kısıtlamalar altında nasıl yerleşeceği üzerinedir.

Eğer bir $\alpha$ eşleştirmesinde, bir öğrenci, herhangi bir $c$ okulunu kendi atandığı okula tercih ediyor ve $c$ okulunun kontenjanı dolu ise ve bu durum bütün öğrenciler için geçerli ise, $\alpha$ eşleşmesi *savurgan olmayan* bir eşleştirmedir.

### 1.2.2 Tek Taraflı Eşleşme Piyasaları

Tek taraflı eşleşme piyasalarında, eşleşmenin bir tarafında nesneler, diğer tarafında ise nesneler için tercihleri bulunan oyuncular bulunmaktadır. Dolayısıyla nesnelerin tercihleri olamayacağı için bu piyasalarda bir tarafın yani oyuncuların tercihleri mevcutken diğer tarafın tercihleri yoktur. Öğrencilerin devlet yurtlarındaki odalara yerleştirilmesi ve ya öğrencilerin kamu okullarına yerleştirilmesi tek taraflı eşleşme piyasalarına örnektir.

Eşleşme piyasalarındaki temel modellerden biri Shapley ve Scarf (1974) tarafından tanıtılmıştır. Ev piyasası olarak adlandırılan bu model ilk tek taraflı bir eşleşme piyasası modelidir. Modelde *n* tane ev ve *n* tane oyuncu (ev sahibi) mevcuttur. Her oyuncu bir eve sahiptir ve bütün evler için bir tercih sıralaması mevcuttur. Dolayısıyla, bir ev piyasası, oyuncuların daha iyi bir ev elde etmek için evlerini takas etme seçeneğine sahip olduğu bir değişim (bölünmez nesnelerle) piyasasıdır (Sönmez ve Ünver, 2010: 788).

Doğan'ın (2014, s.388) ifade ettiği gibi:

> *Bir dağılım, eğer oyuncular kümesinin hiçbir alt kümesindeki oyuncular kendi aralarında değiş tokuş yaparak, bu alt kümedeki hiçbir oyuncuyu daha kötü hale getirmeden en azından bir oyuncuyu daha iyi hale getiremiyorsa, çekirdektedir (core).*

Shapley ve Scarf (1974), çekirdekteki dağılımı bulmak için *En Yüksek Değiş Tokuş Döngüleri* (Top Trading Cycles-TTC) algoritmasını tanıtmışlardır. Ve Shapley ve Scarf (1974), David Gale'e atfettikleri En Yüksek Değiş Tokuş Döngüsü Algoritması'nın buna benzer modellerde her zaman durağan bir eşleşme ürettiğini kanıtlamışlardır (Economic Sciences Prize Committee of the Royal Swedish Academy of Sciences, 2012: 16).

En Yüksek Değiş Tokuş Döngüleri Algoritması'nın işleyişi şu şekildedir:

*1.tur*: İlk turda her oyuncu ilk tercih ettiği evin sahibini gösterir. Eğer ilk tercihi kendi evi ise onu işaret eder. Oyuncu sayısı sonlu olduğu için en az bir döngü oluşur. Döngüdeki her oyuncu işaret ettiği ev sahibinin evine yerleşir. Her oyuncu bir döngünün içinde ise algoritma sona erer. Herhangi bir döngüye giremeyen oyuncu mevcutsa algoritma devam eder.

*k.tur*: Daha önceki turlarda eşleşmeyen oyuncular en çok tercih ettikleri evlerin sahiplerini gösterir. En az bir döngü oluşur. Döngüdeki her oyuncu işaret ettiği ev sahibinin evine yerleşir. Her oyuncu bir döngünün içinde ise algoritma sona erer. Herhangi bir döngüye giremeyen oyuncu mevcutsa algoritma devam eder.

En Yüksek Değiş Tokuş Döngüleri Algoritması'nın çözümünü bir örnek üzerinden gösterelim.

*Örnek 2*: Altı oyuncu ($o_1$, $o_2$, $o_3$, $o_4$, $o_5$, $o_6$) ve altı evin ($h_1$, $h_2$, $h_3$, $h_4$, $h_5$, $h_6$) bulunduğu bir piyasa olsun. Piyasada, başlangıçta $o_1$, $h_1$'e, $o_2$, $h_2$'ye, $o_3$, $h_3$'e, $o_4$, $h_4$'e, $o_5$, $h_5$'e ve $o_6$, $h_6$'ya sahiptir. Oyuncuların tercihleri ise şu şekildedir:

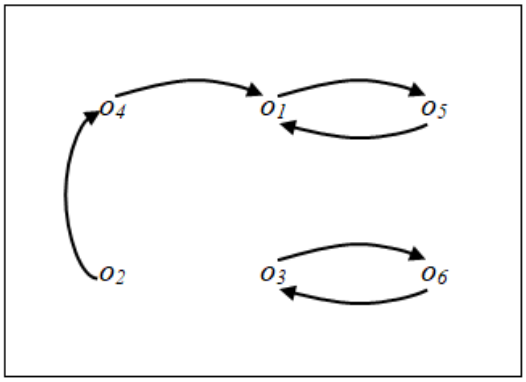
$P_{o_1} = h_5, h_1, h_6, h_4, h_2, h_3$ $\quad$ $P_{o_4} = h_1, h_3, h_5, h_6, h_4, h_2$

$P_{o_2} = h_4, h_2, h_3, h_1, h_6, h_5$ $\quad$ $P_{o_5} = h_1, h_6, h_2, h_3, h_5, h_4$

$P_{o_3} = h_6, h_5, h_3, h_1, h_2, h_4$ $\quad$ $P_{o_6} = h_3, h_5, h_4, h_6, h_1, h_2$

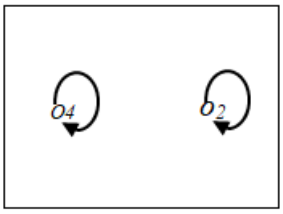
Birinci turda, her oyuncu ilk tercih ettiği evin sahibini gösterir. Dolayısıyla $o_1$, $o_5$'i, $o_2$, $o_4$'ü, $o_3$, $o_6$'yı, $o_4$, $o_1$'i, $o_5$, $o_1$'i ve $o_6$, $o_3$'ü gösterir.

**Şekil 7**: Örnek 2 – 1. Tur




Yukarıdaki şekilde de görüldüğü üzere iki farklı döngü söz konusudur. Bunlar $o_1 \leftrightarrow o_5$ ve $o_3 \leftrightarrow o_6$ döngüleridir. Birinci turda $h_5$, $o_1$'e, $h_1$, $o_5$'e, $h_6$, $o_3$'e ve $h_3$ ise $o_6$'ya tahsis edilir. Bu oyuncular algoritmadan çıkartılır. İkinci turda $o_2$ ve $o_4$ kalmıştır. Her iki oyuncu da kendi evini diğer oyuncunun evine tercih ettiği için kendi evleriyle eşleşirler.

**Şekil 8**: Örnek 2 – 2. Tur




En Yüksek Değiş Tokuş Döngüleri Algoritması'nın uygulanmasından sonra eşleşme şu şekilde oluşur:

$$\mu = \begin{pmatrix} o_1 & o_2 & o_3 & o_4 & o_5 & o_6 \\ h_5 & h_2 & h_6 & h_4 & h_1 & h_3 \end{pmatrix}$$

Tarihsel süreçte bu modelin farklı uygulamaları yapılmıştır. Hylland ve Zeckhauser (1979), ev piyasası modelini farklı bir şekilde kurgulamışlardır. Buna göre modelde, evlerin başlangıçta bir sahibi bulunmamaktadır. Ev piyasası modelinin en geniş ve günümüz ev piyasalarına daha uygun modelini Abdulkadiroglu ve Sönmez (1999) oluşturmuştur. Abdulkadiroglu ve Sönmez'in (1999) kurduğu modele göre evlerin bazılarının sahibi varken bazı evler ise boştur. Eşleşmede oyuncular, hem mevut olarak eve sahip olan ev sahipleri hem de piyasaya yeni gelenlerdir. Evler bu oyuncular arasında Abdulkadiroglu ve Sönmez'in (1999) geliştirdiği "sen benim evimi iste - ben senin sıranı alırım (YRMH-IGYT)" ismini verdikleri bir mekanizmaya göre yeniden dağıtılmaktadır. Bu mekanizma Pareto etkin, bireysel rasyonel ve manipüle edilemez bir mekanizmadır.

### 1.2.3 Okul Seçimi Problemi

Okul seçimi, eğitimde genişçe tartışılmaktadır. Öğrencilerin geleceğini oluşturan en etkin parçalardan biri okullardır. Dolayısıyla öğrenciler kendisi için, aileler ise çocukları için okul seçimi yaparken önemli bir karar almaktadırlar. Okul seçimi, bir nevi öğrenci ve ailesine eğitim göreceği ve geleceğini şekillendireceği okulu seçme fırsatı vermektedir. Geçmişten günümüze öğrenciler okullara atanırken, genellikle ikamet ettiği bölge göz önüne alınarak atanmaktadır. Çoğunlukla bu okullar devlet okullarıdır. Okul seçme hakkı daha çok mali durumu iyi aileler için daha fazla kullanılabilen bir hak olmuştur. Bu tarz sorunların var olması, okul seçimi üzerinde yeni seçim programlarının yaygınlaşmasına neden olmuştur. Bu sebeple okul seçimi problemi en önemli eşleşme piyasalarından biridir.

Her öğrenciyi en fazla tercih ettiği okula yerleştirmek mümkün değildir, dolayısıyla okul seçimiyle ilgili en önemli meselelerden birisi bir öğrenci yerleştirme mekanizmasının tasarlanmasıdır (Abdulkadiroğlu ve Sönmez, 2014: 303). Bu süreçte birçok farklı -kimi açık bir prosedüre sahip kimi ise açık olmayan- uygulamalar kullanılmaya başlanmıştır. Fakat açık prosedüre sahip olan uygulamaların dahi fazlasıyla eksiklikleri mevcuttur. Okulların uyguladığı farklı prosedürler sorunlara yol açmaktaydı. Bu sebeple meydana gelmiş bu sorunların mümkün olduğunca giderilmesi için yeni öğrenci yerleştirme mekanizmalar dizayn edilmiştir.

Okul seçimi probleminde Abdulkadiroğlu ve Sönmez'in (2014, s.307) belirttiği gibi:

*(...) her biri belirli bir sayıdaki okula yerleştirilecek olan bir miktar öğrenci vardır. Her okulun bir maksimum kapasitesi olmakla birlikte toplam kontenjan konusunda bir kıtlık yoktur. Her öğrencinin bütün okullar için kuvvetli tercihleri, her okulun da öğrenciler için kuvvetli öncelik sıralamaları vardır.*

Okul seçimi probleminde okullar, stratejik bir oyuncu değil tüketilecek bir nesne olarak görülmektedir. Okul seçimi probleminde okulların önceliklerinin dışsal olarak yani yasal düzenlemeler ile düzenlendiği varsayılmaktadır (Doğan, 2014: 391). Okulların tüketilecek birer nesne olarak kabul edilmesinin sebebi bu düzenlemelerin varlığıdır.

Eğer okullar öğrencilere, her öğrenci bir okula yerleşecek ve hiçbir okula kapasitesinden fazla öğrenci yerleştirilmeyecek şekilde atanırsa, okul seçimi problemi sonuca ulaşır (Abdulkadiroğlu ve Sönmez, 2014: 307). Bu atamalar her okul seçimi problemi için oluşan *öğrenci atama mekanizma*ları tarafından yapılır. Bir öğrenci atama mekanizması, eğer öğrencilerin okullar hakkındaki tercihlerini açığa çıkarmalarını gerektirip, bu tercihler ve öğrenci önceliklerine göre bir eşleştirme seçiyorsa bir "doğrudan mekanizmadır (direct mechanism)" (Abdulkadiroğlu ve Sönmez, 2014: 307). Bu şekildeki her sonuca "eşleştirme" adını veriyoruz. (Abdulkadiroğlu ve Sönmez, 2014: 307).

### 1.2.4 Üniversiteye Giriş Problemi Ve Okul Seçimi Probleminin Karşılaştırılması

Okul seçimi problemi ve üniversiteye giriş problemi benzer olmakla birlikte, bu iki problem arasındaki temel fark; üniversiteye girişlerde okullar, öğrenciler üzerinde kendi tercihleri olan aktörlerdir, ancak okul seçimi probleminde okullar, öğrenciler tarafından tüketilecek nesnelerdir (Abdulkadiroğlu ve Sönmez, 2014: 307). Yani okul seçimi probleminde tek oyuncular öğrencilerdir: Sınav puanları ve okullar için tercihleri atamaları belirler (Balinski ve Sönmez, 1999: 74). İki problem arasındaki bir diğer farklılık ise; Okul seçiminde haklı kıskançlığın (justified envy) ortadan kaldırılması mantıklıdır, ancak bir zorunluluk değildir, eğer empoze edilirse, o zaman okul seçimi sorunu, durağan üniversiteye giriş ataması ile izomorfik olur (Sönmez ve Ünver, 2011: 831). Dolayısıyla Abdulkadiroğlu ve Sönmez'in (2003) açıkladığı "okul seçimi problemi" ve Gale ve Shapley'nin (1962) tanıttığı üniversiteye giriş problemi benzer olmakla birlikte aslında önemli farkları mevcuttur.

Üniversite giriş probleminde önemli kavramlardan biri *durağanlık*'tır. Uygulama sonucunda oluşacak atamaların durağan olması çok önemlidir: Öğrenci *s*'nin okul *c*'yi, yerleştiği okula, okul *c*'ninse öğrenci *s*'yi, kabul ettiği bir veya birden daha fazla öğrenciye tercih ettiği öğrenci-okul (*s, c*) ikililerinden birbiriyle eşleşmeyen hiçbir ikili kalmamalıdır (Abdulkadiroğlu ve Sönmez, 2014: 305). Daha açık anlatmak gerekirse, herhangi bir öğrencinin yerleştirildiği okula ($c_1$) nazaran daha fazla tercih ettiği başka bir okula ($c_2$) yerleştirilmiş olan öğrenciden o okulda ($c_2$) daha öncelikli olmaması gerekmektedir. Bu matematiksel özellik, okulların tercihlerinin olmayıp önceliklerinin olduğu okul seçimi problemi bağlamında, şu cazip özelliğe karşılık gelmektedir: Öğrenci *s*'nin okul *c*'yi yerleştirildiği okula tercih ettiği ve okul *c*'ye yerleştirilen herhangi bir öğrenciden daha yüksek bir önceliği olduğu öğrenci-okul (*s, c*) ikililerinden, birbiriyle eşleşmeyen hiçbiri kalmamalıdır (Abdulkadiroğlu ve Sönmez, 2014: 305). Yani haklı kıskançlığı ortadan kaldırmalıdır. Bu durumda o eşleştirme adil bir eşleştirme olmaktadır. Bundan dolayı, üniversiteye giriş bağlamındaki durağan bir eşleşme, okul seçimi bağlamındaki haklı kıskançlığı ortadan kaldırmaktadır (Abdulkadiroğlu ve Sönmez, 2014: 305). Okul seçimi bağlamında önemli olan tek şeyin öğrencilerin refahı olduğundan bu eşleşme, haklı kıskançlığı ortadan kaldıran diğer herhangi bir eşleşmeyi Pareto domine etmektedir (Abdulkadiroğlu ve Sönmez, 2014: 305).

# İKİNCİ BÖLÜM

# 2. EŞLEŞME MEKANİZMALARI

## 2.1 ÜNİVERSİTEYE GİRİŞ PROBLEMİ VE OKUL SEÇİMİ PROBLEMİNDE KULLANILAN MEKANİZMALAR

Çalışmanın bir önceki bölümünde bahsedildiği gibi her öğrenciyi ilk tercihine yerleştirmek mümkün değildir. Bu yüzden atamaların yapılmasında mekanizmalara ihtiyaç duyulmaktadır. Geliştirilen mekanizmaların birçoğunun çıkış noktası Gale-Shapley algoritmasıdır.

### 2.1.1 Gale-Shapley Öğrenci Optimal Durağan Mekanizması

Gale-Shapley algoritması, hem üniversiteye giriş problemi hem de okul seçimi probleminin yaşandığı piyasalarda uygulanabilen bir algoritmadır. Algoritma teklif eden oyuncu tarafına göre oyuncuların getirisi yönünden farklılaşmaktadır. Haklı kıskançlığı ortadan kaldıran *Gale-Shapley Öğrenci Optimal Durağan Mekanizması*, öğrencinin teklifte (başvuru) bulunduğu mekanizmadır ve diğer eşleşmeleri Pareto domine etmektedir. Gale-Shapley Öğrenci Optimal Mekanizması, her yerleştirme problemi için ilgili üniversiteye giriş probleminin öğrenci için en ideal olan durağan eşleşmesini seçer (Balinski ve Sönmez, 1999: 80). Gale-Shapley Öğrenci Optimal Durağan Mekanizması'nın uygulaması şu şekilde yapılmaktadır:

*1.tur*: her öğrenci ilk tercihine başvuruda bulunur. Okullar mevcut başvuruları önceliklerine göre değerlendirir ve kotalarını öğrencilere geçici olarak tahsis eder. Diğer teklifleri reddeder. Henüz atamaları yapmaz.

*k.tur*: genel olarak, bir önceki turda reddedilen öğrenciler, tercih sıralamasında bir sonraki tercihe başvuruda bulunur. Öğrencinin başvurduğu okul, önceki turlarda beklettiği öğrenci başvurularını yeni başvurular ile karşılaştırır. Öncelik sırasına göre değerlendirir ve kotasını daha öncelikli öğrencilere geçici olarak tahsis eder ve diğerlerini reddeder.

Algoritma, hiçbir öğrencinin teklifi okul tarafından reddedilmediğinde ve her öğrenci son geçici kontenjanına yerleştiğinde son bulur (Abdulkadiroğlu ve Sönmez, 2014: 310).

Bu uyarılmış doğrudan mekanizmaya Gale-Shapley Öğrenci Optimal Durağan Mekanizması adı verilmektedir. (Abdulkadiroğlu ve Sönmez, 2014: 310).

Bir örnek üzerinden atamaları yapıp sonuçları bulalım.

*Örnek 3:* Dört okul $C=\{c_1, c_2, c_3, c_4\}$ ve bu okullara başvuruda bulunan dört öğrencinin $S=\{s_1, s_2, s_3, s_4\}$ olduğu bir örnek düşünelim:

*bütün okulların 1 kotası bulunmaktadır yani her okul en fazla bir öğrenciyi kabul edebilecektir:*

$q= q_{c_1} + q_{c_2} + q_{c_3} + q_{c_4}$

$q_{c_1} = q_{c_2} = q_{c_3} = q_{c_4} = 1,$

*tercihler*:

$P_{s_1} = c_4, c_2, c_3, c_1$ $\quad$ $P_{s_3} = c_1, c_2, c_4, c_3$

$P_{s_2} = c_4, c_3, c_1, c_2$ $\quad$ $P_{s_4} = c_2, c_3, c_4, c_1$

$P_{c_1} = s_1, s_4, s_2, s_3$ $\quad$ $P_{c_3} = s_4, s_3, s_2, s_1$

$P_{c_2} = s_1, s_3, s_4, s_2$ $\quad$ $P_{c_4} = s_3, s_1, s_2, s_4$

Eşleşmenin birinci turunda her öğrenci ilk tercihine başvuruda bulunur. $s_1$ ve $s_2$ öğrencileri $c_4$, $s_3$ öğrencisi $c_1$ ve $s_4$ öğrencisi $c_2$ okuluna başvurur. Okullar mevcut başvuruları önceliklerine göre değerlendirir. $c_1$ ve $c_2$ okulları mevcut başvuruları bekletir. $c_4$ okulu, kendi önceliklerine bağlı olarak kendisine başvuruda bulunan 2 öğrenciden daha öncelikli olanı ($s_1$) bekletir ve diğerini reddeder.

**Şekil 9**: Örnek 3 – Nihai Atamalar

| | $c_1$ | $c_2$ | $c_3$ | $c_4$ |
|---|---|---|---|---|
| *1.tur* | $s_3$ | $s_4$ | | $s_1, s_2$ |
| *2.tur* | $s_3$ | $s_4$ | $s_2$ | $s_1$ |

İkinci turda, $s_2$ öğrencisi hariç bütün öğrenciler beklemeye geçer, birinci turda başvurusu reddedilen $s_2$ bir sonraki tercihine yani $c_3$ okuluna başvuruda bulunur. Son olarak, bu turda bütün okulların kotaları dolmuş ve her öğrenci geçici olarak atanmıştır. Dolayısıyla başka bir başvuru olmaz. Başvurusu bekletilen bütün öğrenciler kabul edilir ve atama tamamlanır. $s_1$ öğrencisi $c_4$ okuluna, $s_2$ öğrencisi $c_3$ okuluna, $s_3$ öğrencisi $c_1$ ve $s_4$ öğrencisi $c_2$ okuluna atanır.

$$\mu = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 \\ c_4 & c_3 & c_1 & c_2 \end{pmatrix}$$

### 2.1.2 Gale-Shapley Okul Optimal Durağan Mekanizması

Gale-Shapley Okul Optimal Durağan Mekanizması uygulama olarak Gale-Shapley Öğrenci Optimal Durağan Mekanizması'ndan tek bir farklılık göstermektedir. Gale-Shapley Okul Optimal Durağan Mekanizması'nda teklif götüren taraf okullardır. Gale-Shapley Okul Optimal Mekanizması, her yerleştirme problemi için ilgili okul kabul probleminin okul için en ideal olan durağan eşleşmesini seçer (Balinski ve Sönmez, 1999: 80). Uygulaması Gale-Shapley Öğrenci Optimal Mekanizması ile aynı olan fakat teklif götüren tarafta farklılık olması ile ayrışan Gale-Shapley Okul Optimal Durağan Mekanizması'nın uygulamasını *Örnek 3*'ü bu mekanizmaya uygun olarak tekrar çözerek gösterelim.

*Örnek 4:* Dört okulun $C=\{c_1, c_2, c_3, c_4\}$ ve bu okullara başvuruda bulunan dört öğrencinin $S=\{s_1, s_2, s_3, s_4\}$ olduğu bir okul seçimi problemi düşünelim;

*bütün okulların 1 kotası bulunmaktadır, yani her okul en fazla bir öğrenciyi kabul edebilecektir:*

$q = q_{c_1} + q_{c_2} + q_{c_3} + q_{c_4}$

$q_{c_1} = q_{c_2} = q_{c_3} = q_{c_4} = 1$

*tercihler*:

$P_{s_1} = c_4, c_2, c_3, c_1$ $\qquad$ $P_{s_3} = c_1, c_2, c_4, c_3$

$P_{s_2} = c_4, c_3, c_1, c_2$ $\qquad$ $P_{s_4} = c_2, c_3, c_4, c_1$

$P_{c_1} = s_1, s_4, s_2, s_3$ $\qquad$ $P_{c_3} = s_4, s_3, s_2, s_1$

$P_{c_2} = s_1, s_3, s_4, s_2$ $\qquad$ $P_{c_4} = s_3, s_1, s_2, s_4$

**Şekil 10**: Örnek 4 – Nihai Atamalar

| | $s_1$ | $s_2$ | $s_3$ | $s_4$ |
|---|---|---|---|---|
| *1.tur* | $c_1, c_2$ | | $c_4$ | $c_3$ |
| *2.tur* | $c_2$ | | $c_4$ | $c_3, c_1$ |
| *3.tur* | $c_2$ | $c_1$ | $c_4$ | $c_3$ |

Eşleşmenin birinci turunda her okul ilk tercihine başvuruda bulunur. $c_1$ ve $c_2$ okulları $s_1$ öğrencisine, $c_3$ okulu $s_4$ ve $c_4$ okulu ise $s_3$ öğrencisine teklif götürür. Öğrenciler mevcut teklifleri önceliklerine göre değerlendirir. $s_3$ ve $s_4$ öğrencileri tek teklif aldıkları için bu teklifleri bekletirler. $s_1$ öğrencisi kendi önceliklerine bağlı olarak kendisine başvuruda bulunan iki okuldan daha öncelikli olanı ($c_2$) bekletir ve diğerini reddeder. İkinci turda, $c_1$ okulu hariç bütün okullar beklemeye geçer, birinci turda başvurusu reddedilen $c_1$ bir sonraki tercihine yani $s_4$ öğrencisine teklifte bulunur. İkinci turda iki teklif alan $s_4$ kendi tercih sıralamasına göre daha önce tercih ettiği okulu ($c_3$) bekletir ve diğer teklifi reddeder. Bir önceki turda teklifi reddedilen $c_1$, üçüncü turda bir sonraki tercihine ($c_2$) teklif götürür. Son olarak, bu turda bütün öğrenciler tek bir teklif almış ve dolayısıyla her öğrenci geçici olarak atanmıştır. Başka bir teklif olmaz. Aldıkları teklifleri bekleten bütün öğrenciler teklifleri kabul eder ve atama tamamlanır. $s_1$ öğrencisi $c_2$ okuluna, $s_2$ öğrencisi $c_1$ okuluna, $s_3$ öğrencisi $c_4$ okuluna ve $s_4$ öğrencisi $c_3$ okuluna atanır.

$$\mu_{\text{okul optimal}} = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 \\ c_2 & c_1 & c_4 & c_3 \end{pmatrix}$$

*Gale-Shapley Öğrenci Optimal Durağan Mekanizması'na göre yapılan atamalar;*

$$\mu_{\text{öğrenci optimal}} = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 \\ c_4 & c_3 & c_1 & c_2 \end{pmatrix}$$

Gale-Shapley Öğrenci Optimal Durağan Mekanizması'na göre yapılan atamalarda, $s_1$, $s_3$ ve $s_4$ öğrencileri ilk tercihine, $s_2$ öğrencisi ise ikinci tercihine yerleştirilmişti. Okullara bakacak olursak bu atamalar sonucu $c_1$ son tercihini, $c_2$ ve $c_3$ üçüncü tercihini ve son olarak $c_4$ ise ikinci tercihini kabul etmek durumunda kalmıştı. Fakat Gale-Shapley okul optimal durağan mekanizmasına göre yapılan atamalarda, $c_1$, $c_3$ ve $c_4$ okulları ilk tercihini, $c_2$ okulu üçüncü tercihini kabul ederken öğrencilerin durumunu karşılaştırdığımızda bütün öğrencilerin Gale-Shapley Öğrenci Optimal Durağan Mekanizması'na göre yapılan atamadan daha kötü durumda olduğunu görebilmekteyiz.

$R_{\mu_{\text{öğrenci optimal}(s_1)}} = 1$, $R_{\mu_{\text{öğrenci optimal}(s_2)}} = 2$, $R_{\mu_{\text{öğrenci optimal}(s_3)}} = 1$, $R_{\mu_{\text{öğrenci optimal}(s_4)}} = 1$

$R_{\mu_{\text{öğrenci optimal}(c_1)}}$= 4, $R_{\mu_{\text{öğrenci optimal}(c_2)}}$= 3, $R_{\mu_{\text{öğrenci optimal}(c_3)}}$= 3, $R_{\mu_{\text{öğrenci optimal}(c_4)}}$= 2

$R_{\mu_{okul\ optimal(s_1)}}$ = 2, $R_{\mu_{okul\ optimal(s_2)}}$ = 3, $R_{\mu_{okul\ optimal(s_3)}}$= 3, $R_{\mu_{okul\ optimal(s_4)}}$= 2

$R_{\mu_{okul\ optimal(c_1)}}$ = 1, $R_{\mu_{okul\ optimal(c_2)}}$ = 3, $R_{\mu_{okul\ optimal(c_3)}}$= 1, $R_{\mu_{okul\ optimal(c_4)}}$= 1

### 2.1.3 Gale-Shapley Öğrenci Tipine Özgü Kotalarla Optimal Durağan Mekanizması

Özellikle Amerika'da olmak üzere bazı bölgelerde okula kabul sürecinde bazı kısıtlamalar uygulanmaktadır. Öğrenciler için açılan kontenjanlara sınırlamalar koyulabilmektedir. Kontrollü seçim adı altında yapılan bu kısıtlamalar çoğunlukla okullardaki ırksal dengeyi sağlamak için yapılmaktadır. Farklı ırklardaki öğrencilere özgü açılan kotalar okulların bu sayıları kontrol edebilmelerini sağlamaktadır. Örneğin, İngiltere'deki Şehir Teknoloji Kolejleri'nin öğrenci kabul ederken yetenek aralığının tamamını kapsayacak şekilde bir öğrenci grubu tercih etmesi zorunludur ve aynı zamanda okulun öğrencileri, okulun hizmet ettiği bölgedeki zümrenin tamamını temsil etmesi gerekmektedir (Hirch, 1994: 120; Aktaran Abulkadiroğlu ve Sönmez, 2014: 314).

Kotalara özgü atama durumunun Gale-Shapley Öğrenci Optimal Durağan Mekanizmasın'a adapte edilmesi sonucu *Gale-Shapley Öğrenci Tipine Özgü Kotalarla Optimal Durağan Mekanizması* oluşturulmuştur.

İki farklı tipte öğrenci olduğunu varsayalım, kontrollü seçim kısıtlamalarının tamamen sabit olduğu durumda, Gale-Shapley Öğrenci Optimal Durağan Mekanizması her öğrenci tipi için ayrı ayrı uygulanarak mevcut mekanizmada bir farklılaştırma yapmadan atamalar yapılabilir ve her öğrenci kendi tipine özgü kotalarla yerleştirilir. Kontrollü seçim kısıtlamalarının esnek olduğu durumda, Gale-Shapley Öğrenci Optimal Durağan Mekanizması şu şekilde uygulanır:

*1.tur*: Her öğrenci ilk tercihine başvuruda bulunur. Okullar mevcut başvuruları önceliklerine göre değerlendirir ve kotalarını öğrencilere geçici olarak tahsis eder. Eğer bir tip için kota dolarsa, o tipten olan ve başvuruda bulunan kalan

öğrenciler reddedilir ve geçici atamalar farklı tipteki öğrencilerle devam eder (Abdulkadiroğlu ve Sönmez, 2014: 315).

*k.tur*: Genel olarak, bir önceki turda reddedilen öğrenciler, tercih sıralamasında bir sonraki tercihe başvuruda bulunur. Öğrencinin başvurduğu okul, önceki turlarda beklettiği öğrenci başvurularını yeni başvurular ile karşılaştırır. Öncelik sırasına göre değerlendirir ve kotasını daha öncelikli öğrencilere geçici olarak tahsis eder ve diğerlerini reddeder. Eğer bir tip için kota dolarsa, o tipten olan ve başvuruda bulunan kalan öğrenciler reddedilir ve geçici atamalar farklı tipteki öğrencilerle devam eder (Abdulkadiroğlu ve Sönmez, 2014: 315). Mekanizma, tüm öğrencilerin ataması yapılana kadar veya bütün okulların kontenjanları bitene kadar devam eder.

### 2.1.4 En Yüksek Değiş Tokuş Döngüleri Mekanizması (Top Trading Cycle Mechanism - TTC)

En Yüksek Değiş Tokuş Döngüleri Mekanizması Shapley and Scarf'ın (1974) temelini attığı doğrudan bir mekanizmadır. Shapley ve Scarf (1974) her biri bölünmez bir mal, bir "ev", ile donatılmış n tane oyuncudan oluşan bir ev piyasası modelini oluşturdu (Roth, Sönmez ve Ünver, 2004: 462). Algoritmanın kullanıldığı ilk alan evlerin, ev sahipleri arasında yeniden tahsis edildiği ev piyasasıydı. Zamanla algoritmanın uygulandığı piyasa sayısı arttı. Abdulkadiroğlu ve Sönmez (2003), bu algoritmayı öğrenci-okul eşleşmesi piyasasına uyguladılar. TTC, hem okul seçimi probleminin bulunduğu piyasalarda hem de üniversiteye giriş probleminin bulunduğu piyasalarda kullanılan mekanizmalardandır.

En Yüksek Değiş Tokuş Döngüleri Mekanizması uygulamada en yüksek önceliğe sahip öğrencilerle başlamaktadır. Bu öğrenciler kendi aralarında, eğer Pareto gelişim mevcutsa, en yüksek öncelikli okulları birbirleri ile değiş tokuş yapabilmektedir. Mekanizma, değiş tokuşu yapan öğrenciler çıktıktan sonra geri kalan öğrenciler arasından en yüksek önceliğe sahip olanlar ile devam eder. Bu Pareto etkin mekanizma *En Yüksek Öncelikler Arası Değiş Tokuş Döngüleri Mekanizması* olarak adlandırılmaktadır. Mekanizma tam olarak şu şekilde işlemektedir:

*1.tur*: Her okulun kendi kontenjanına bağlı olan sayaçları mevcuttur. Ve bu sayaçlar öğrenci atandıkça güncellenmektedir. Her okul kendi öncelik sırasına göre en yüksek önceliğe sahip öğrenciyi belirtir. Her öğrenci kendi tercih sırasına göre ilk tercih ettiği okulu belirtir. Okul ve öğrencilerin sayısı sonlu olduğu için en azından bir tane döngü vardır (Abdulkadiroğlu ve Sönmez, 2014: 312). Bu döngü: $c_1$'in $s_1$'i, $s_1$'in $c_2$'yi, … , $c_k$'nin $s_k$'yi, $s_k$'nin $c_1$'i belirttiği ($c_1$, $s_1$, $c_2$, …, $c_k$, $s_k$) şeklindeki listedir. Her öğrenci ve okul aynı anda sadece tek bir döngüde yer alabilir. Oluşan döngüdeki her öğrenci belirttiği okula yerleştirilir ve listeden çıkartılır, aynı zamanda o okulun sayacı bir azaltılır. Eğer okulun sayacı "0" olursa o okul listeden çıkartılır. Diğer okulların sayaçlarında herhangi bir değişiklik yapılmaz.

*k.tur*: Genel olarak, henüz atanmayan her öğrenci kalan okullar arasından ilk tercihini belirtir. Kalan her okul ise henüz atanmamış öğrenciler arasından en öncelikli öğrenciyi belirtir. En az bir döngü oluşur. Döngüdeki her öğrenci belirttiği okula yerleştirilir ve listeden çıkartılır, o okulun sayacı bir azaltılır. Eğer herhangi bir okulun sayacı "0" olursa o okul listeden çıkartılır. Bütün öğrenciler bir okula yerleşene kadar bu atamalar devam eder.

Bir örnek üzerinden En Yüksek Değiş Tokuş Döngüleri Mekanizması ile atamaları yapıp sonuçları bulalım.

*Örnek 5:* Dört okul $C=\{c_1, c_2, c_3, c_4\}$ ve bu okullara atanacak olan altı öğrencinin $S=\{s_1, s_2, s_3, s_4, s_5, s_6\}$ bulunduğu bir örnek düşünelim.

$q = q_{c_1} + q_{c_2} + q_{c_3} + q_{c_4}$

$q_{c_1} = q_{c_2} = 2,\ q_{c_3} = q_{c_4} = 1$

*tercihler:*

$P_{s_1} = c_1, c_3, c_2, c_4$ $\qquad$ $P_{s_4} = c_4, c_1, c_2, c_3$

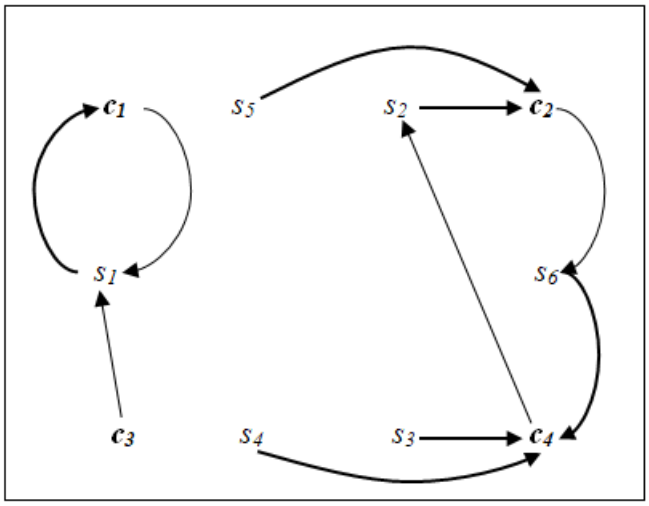
$P_{s_2} = c_2, c_3, c_1, c_4$ $\qquad$ $P_{s_5} = c_2, c_1, c_4, c_3$

$P_{s_3} = c_4, c_3, c_2, c_1$ $\qquad$ $P_{s_6} = c_4, c_1, c_3, c_2$

$P_{c_1} = s_1, s_3, s_2, s_5, s_4, s_6$ $\qquad$ $P_{c_3} = s_1, s_5, s_3, s_4, s_6, s_2$

$P_{c_2} = s_6, s_4, s_3, s_2, s_5, s_1$ $\qquad$ $P_{c_4} = s_2, s_3, s_1, s_4, s_5, s_6$

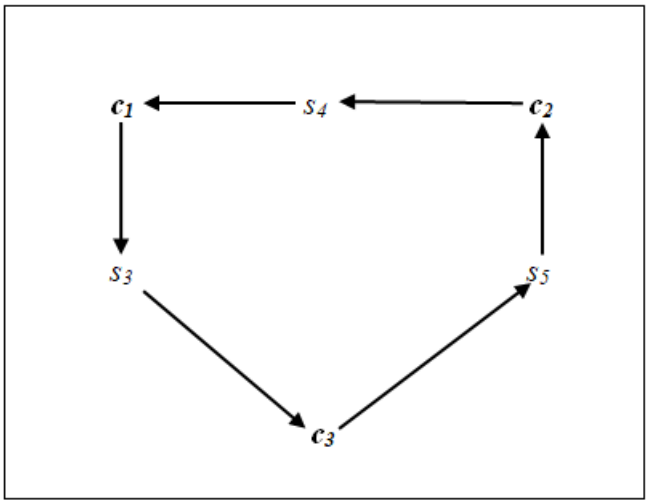
**Şekil 11**: Örnek 5 – 1. Tur




Şekil 1'de öğrencilerin ilk tercihlerini ve okulların en öncelikli öğrencisini belirttiği şekil tablosunu görmekteyiz. Aslında bu durum birinci tur'un şekil olarak gösterimidir. Şekil 1'de de görülebildiği üzere birinci tur'da iki döngü mevcuttur: $s_2 \rightarrow c_2 \rightarrow s_6 \rightarrow c_4 \rightarrow s_2$ döngüsü ve $s_1 \leftrightarrow c_1$ döngüsü. Bu döngüler sonucunda birinci tur'da $s_1$, $s_2$ ve $s_6$ öğrencileri sırasıyla $c_1$, $c_2$ ve $c_4$ okullarına yerleştirilir. Bu okulların kontenjanları azaltılır ve kontenjanı "0" olan okul listeden çıkartılır. Okullara yerleştirilen öğrenciler listeden çıkartılır. Birinci tur sonunda:

- $s_1$ öğrencisi $c_1$ okuluna, $s_2$ öğrencisi $c_2$ okuluna ve $s_6$ öğrencisi $c_4$ okuluna yerleştirilmiştir,
- Ataması yapılan $s_1$, $s_2$ ve $s_6$ öğrencileri ve kontenjanı biten $c_4$ okulu listeden çıkartılmıştır,
- $c_1$ *ve* $c_2$ okullarının kontenjanları 2'den 1'e düşürülmüştür: $q_{c_1} == 1$, $q_{c_2} = 1$.

**Şekil 12:** Örnek 5 – 2. Tur




İkinci turda bir önceki turda ataması yapılmayan öğrenciler ve kontenjanı henüz bitmemiş olan okullar atamaya katılır. Bu turda da diğer turlarda olduğu gibi öğrencilerin ilk tercihlerine ve okulların en öncelikli olarak belirttikleri öğrencilere bakılır. Fakat bir önceki turda atanan öğrenciler ve kontenjanı biten okullar dikkate alınmaz. Burada $s_3$ öğrencisi artık kontenjanı dolan $c_4$ okulunu ilk tercih olarak göstermeyeceği için en iyi ikinci tercihini yani $c_3$ okulunu belirtir. Aynı sebeple $s_4$ öğrencisi ise $c_4$ okulu yerine $c_2$ okulunu belirtir. Okullar ise ataması henüz yapılmamış olan öğrenciler arasından en öncelikli öğrencilerini belirtirler. Şekil 2'de de görülebildiği üzere ikinci turda tek bir döngü vardır: $s_4 \rightarrow c_1 \rightarrow s_3 \rightarrow c_3 \rightarrow s_5 \rightarrow c_2 \rightarrow s_4$ döngüsü. Ve bu döngü kalan bütün okulları ve öğrencileri içine almaktadır. Dolayısıyla bu döngü sonrası algoritmanın son bulacağı anlaşılmaktadır. İkinci tur sonunda:

- $s_4$ öğrencisi $c_1$ okuluna, $s_3$ öğrencisi $c_3$ okuluna ve $s_5$ öğrencisi $c_2$ okuluna yerleştirilmiştir,
- boş kontenjanı bulunan okul ve ataması yapılmayan öğrenci kalmadığı için algoritma son bulmuştur.

$$\mu_{TTC} = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 \\ c_1 & c_2 & c_3 & c_1 & c_2 & c_4 \end{pmatrix}$$

Atamalar sonucunda $s_1$, $s_2$, $s_5$ *ve* $s_6$ öğrencileri ilk tercihlerine, $s_3$ *ve* $s_4$ öğrencileri ise ikinci tercihlerine yerleşmişlerdir.

$$R_{\mu_{TTC(s_1)}} = 1,\ R_{\mu_{TTC(s_2)}} = 1,\ R_{\mu_{TTC(s_3)}} = 2,\ R_{\mu_{TTC(s_4)}} = 2,\ R_{\mu_{TTC(s_5)}} = 1,\ R_{\mu_{TTC(s_6)}} = 1$$

$R_{\mu_{TTC(c_1)}}$ = 1, $R_{\mu_{TTC(c_2)}}$ = 4, $R_{\mu_{TTC(c_3)}}$ = 3, $R_{\mu_{TTC(c_4)}}$ = 6

#### 2.1.4.1 Tipe Özel Kotalı En Yüksek Değiş Tokuş Döngüleri Mekanizması

Kontrollü seçim uygulamaları örneklerinden biri de Türe Özgü Kotalarla En Yüksek Değiş Tokuş döngüleri mekanizmasıdır. Kontrollü seçim uygulamalarının tamamen sabit olması durumunda En Yüksek Değiş Tokuş Döngüleri Mekanizması'nda herhangi bir değişiklik yapılmasına gerek yoktur. Kontrollü seçim kısıtlamaları esnek olduğundaysa, En Yüksek Değiş Tokuş Döngüleri Mekanizması'nda her okul için, boş kontenjanların hesabını tutan sayacın yanında bir de her tip öğrenci için tipe-özel bir sayaç dâhil edilir (Abdulkadiroğlu ve Sönmez, 2014: 316). Mekanizmanın uygulanması şu şekildedir:

*1.tur*: Her okula bir sayaç eklenir ve bu sayaç o okulun toplam kontenjana eşitlenir. Her okula tipe-özel sayaç eklenir ve bu sayaç o okuldaki ilgili tipin kontenjanına eşitlenir. Her öğrenci, mevcut okullar arasından kendi tipine ait kontenjanın "0" olmadığı bir okulu ilk tercihi olarak belirtir. Her okul, kendi için en yüksek önceliğe sahip öğrenciyi belirtir (Abdulkadiroğlu ve Sönmez, 2014: 316). En az bir döngü oluşur. Bu döngüdeki tüm öğrenciler belirttikleri okulların kontenjanlarına yerleştirilir. Öğrencilerin yerleştirildiği her okulun toplam sayacı ve tipe-özel sayacı (öğrenci hangi tipte ise o tipe ait sayaç) bir azaltılır. Ataması yapılan öğrenciler listeden çıkartılır. Eğer döngü sonrası toplam sayacı sıfıra düşen bir okul varsa o okul listeden çıkartılır. En az bir tane öğrenci kaldıysa, bir sonraki basamakla devam edilir (Abdulkadiroğlu ve Sönmez, 2014: 316).

*k.tur*: Genel olarak, her öğrenci kalan okullar arasından kendi tipine ait kontenjanı bitmemiş olan bir okulu ilk tercihi olarak belirtir. Kalan her okul henüz atanmamış olan öğrenciler arasından kendisi için en öncelikli olanını belirtir. En az bir döngü oluşur. Bu döngüdeki tüm öğrenciler belirttikleri okulların kontenjanlarına yerleştirilir. Öğrencilerin yerleştirildiği her okulun toplam sayacı ve tipe-özel sayacı (öğrenci hangi tipte ise o tipe ait sayaç) bir azaltılır. Ataması yapılan öğrenciler listeden çıkartılır. Eğer döngü sonrası toplam sayacı sıfıra düşen bir okul varsa o okul listeden çıkartılır. En az bir tane öğrenci kaldıysa, bir sonraki basamakla devam edilir (Abdulkadiroğlu ve Sönmez, 2014: 316).

### 2.1.5 Çok Kategorili Seri Diktatörlük

En Yüksek Değiş Tokuş Döngüleri Mekanizması'nın en yüksek önceliğe sahip öğrencilerden başlayarak bu öğrencilerin önceliklerini birbirleri arasında takas yapmasına olanak sağlayan bir mekanizma olduğunu biliyoruz. Tüm okulların aynı öncelik sırasına sahip olduğu çok özel durumda (örneğin tüm okullar tek bir kura çekiminden gelen sıralamayı kullanırsa) mekanizma, bu öncelik sıralamasından yönlenen *seri diktatörlük*'e indirgenmektedir (Abdulkadiroğlu ve Sönmez, 2014: 316). Bu durumda en yüksek puana sahip öğrenci ilk tercihine ikinci öğrenci kalan okullar arasındaki en yüksek tercihine yerleştirilir.

Çok Kategorili Seri Diktatörlük Mekanizması'nda (MCSD), seri diktatörlükten farklı olarak kategoriler mevcuttur. Öğrenciler farklı kategoriler için farklı öncelik sırasına sahip olabilirler. Örneğin öğrenciler okula yerleşmek için girdikleri bir sınavdan iki farklı kategoride de puan alabilirler. Ve bu iki farklı kategoriye uygun kontenjanlara başvurabilirler. Bu mekanizma üniversiteye giriş probleminin bulunduğu piyasalarda kullanılabilen bir mekanizmadır. Mekanizmanın uygulamasını kısa bir örnek ile açıklamak gerekirse: öğrencilerin tek kategorisi ($t_1$) olan ortak sınavdan aldıkları puana göre sıralandıkları ve bu sıralama önceliğine göre okullara yerleştirildikleri bir mekanizma düşünelim. İki okul ($c_1$ ve $c_2$) ve bu okullara atanacak iki öğrencinin ($s_1$ ve $s_2$) olduğu bir atama yapalım. Her iki okulunda kontenjanının 1 olduğunu varsayalım. Öğrencilerin sınavdan aldıkları puanlar: $f^{s_1} = (f_{t_1}=90)$, $f^{s_2} = (f_{t_1} = 80)$ ve $P_{s_1}=c_1, c_2$, $P_{s_2}=c_1, c_2$ şeklindedir. Öğrencilerin ikisi de $c_1$ okulunu ilk tercih olarak belirtmiştir. Fakat öğrenciler puan sıralamasına göre önceliğine sahip oldukları için $s_1$ öğrencisi ilk tercihi olan $c_1$ okuluna yerleşirken $s_2$ öğrencisi ikinci tercihi olan $c_2$ okuluna yerleşir.

Bu mekanizma, Gale-Shapley Öğrenci Tipine Özgü Kotalarla Optimal Durağan Mekanizması ve Tipe-Özel Kotalı En Yüksek Değiş Tokuş Döngüleri Mekanizması ile benzerlik göstermektedir. Fakat Çok Kategorili Seri Diktatörlük Mekanizması'nda öğrenciler iki kategoriden de tercih yapıp o kategoriye uygun okula yerleşebilirler. Üstteki örneği biraz genişleterek Çok Kategorili Seri Diktatörlük Mekanizması'nın uygulamasını yapalım.

*Örnek 6*: Dört okul $C=\{c_1, c_2, c_3, c_4\}$ ve bu okullara ortak bir sınavdan aldıkları puanlara göre atanacak olan altı öğrenci $S=\{s_1, s_2, s_3, s_4, s_5, s_6\}$ olsun. Bu sefer

sınavda iki farklı tip olduğunu (iki farklı puan) ve okulların iki farklı tipte öğrenci kabul edebileceklerini varsayalım:

*Okullardan $c_1$ ve $c_3$, tip 1($t_1$) tipindeki öğrencileri kabul ederken, $c_2$ ve $c_4$, tip 2 ($t_2$) tipindeki öğrencileri kabul etmektedir.*

*tipler:* $T=\{t_1, t_2\}$, $t(c_1)= t(c_3)= t_1$, $t(c_2)= t(c_4)= t_2$

$q = q_{c_1} + q_{c_2} + q_{c_3} + q_{c_4}$

$q_{c_1} = q_{c_2} = 2$, $q_{c_3} = q_{c_4} = 1$

*Öğrencilerin aldıkları puanlar:*

$f^{s_1} = (f_{t_1}, f_{t_2}) = (80,80)$

$f^{s_2} = (f_{t_1}, f_{t_2}) = (70,60)$

$f^{s_3} = (f_{t_1}, f_{t_2}) = (90,50)$

$f^{s_4} = (f_{t_1}, f_{t_2}) = (95,95)$

$f^{s_5} = (f_{t_1}, f_{t_2}) = (60,70)$

$f^{s_6} = (f_{t_1}, f_{t_2}) = (50,85)$

*Her tip için öğrencilerin aldıkları puanlara göre sıralaması:*

| $t_1$ | $t_2$ |
|---|---|
| $s_4$ | $s_4$ |
| $s_3$ | $s_6$ |
| $s_1$ | $s_1$ |
| $s_2$ | $s_5$ |
| $s_5$ | $s_2$ |
| $s_6$ | $s_3$ |

*tercihler:*

$P_{s_1} = c_3, c_2, c_4, c_1$ $\quad$ $P_{s_4} = c_4, c_1, c_3, c_2$

$P_{s_2} = c_2, c_4, c_3, c_1$ $\quad$ $P_{s_5} = c_3, c_1, c_2, c_4$

$P_{s_3} = c_1, c_2, c_3, c_4$ $\quad$ $P_{s_6} = c_1, c_2, c_4, c_3$

Mekanizma her tip için öğrencilerin aldıkları puanlara göre yapılan sıralamada ilk sırada bulunan öğrenci ile başlar ve okulların kontenjanları dolana kadar sırayla devam eder. Her iki tip için de bu uygulama yapılır. İlk olarak $t_1$ tipi atamaları ile başlayabiliriz. Burada $s_4$ öğrencisi $t_1$ tipinde ilk sırada yer aldığı için algoritma $s_4$

öğrencisi ile başlamaktadır. $t_1$ tipinde atama yapılırken öğrencinin tercihlerinde $t_1$ tipinde öğrenci kabul eden okullar dikkate alınır. Aynı şekilde $t_2$ tipinde atamalar yapılırken öğrencilerin tercihlerinde $t_2$ tipinde öğrenci kabul eden okullar dikkate alınır. $s_4$ öğrencisinin ilk tercihi $c_4$, $t_2$ tipinde öğrenci kabul etmektedir. Dolayısıyla bu tercihi atlıyoruz. Şu anda $t_1$ tipinde atama yapıldığı için öğrencinin tercihlerinde $t_1$ tipinde öğrenci kabul eden ilk okulu bulana kadar sırasıyla ilerliyoruz. $s_4$ öğrencisinin ikinci tercihinin $c_1$ olduğunu ve bu okulun $t_1$ tipinde öğrenci kabul ettiğini görüyoruz. Bu yüzden $s_4$ öğrencisini $c_1$ okuluna geçici olarak atayarak $c_1$ okulunun kontenjanını 1 düşürüyoruz. $t_1$ tipindeki puan sıralamasında bir sonraki sıradaki öğrenciye geçiyoruz: $s_3$. $s_3$ öğrencisinin ilk tercihi $c_1$ ve okulun kontenjanı henüz dolmadığı için $s_3$ öğrencisi $c_1$ okuluna geçici olarak atanır. $c_1$ okulunun kontenjanı şimdilik dolduğu için bundan sonra herhangi bir öğrenci bu okula atanamaz. Sıralamada diğer öğrenciye geçiyoruz: $s_1$. $s_1$ öğrencisinin ilk tercihi $c_3$ olduğunu görüyoruz. Okulun tipi uygun olduğu için ve kontenjanı dolmadığı için $s_1$ öğrencisi $c_3$ okuluna geçici olarak atanır. Okulun kontenjanı 1 azaltılır. $t_1$ tipinde öğrenci kabul eden bütün okulların kontenjanı dolduğu için mekanizm $t_2$ tipinde öğrenci kabul eden okullara öğrenci atamalarıyla devam eder. $t_2$ tipinde en yüksek puanı alan öğrenci $s_4$ olduğu için bu tipte de atamalara $s_4$ öğrencisi ile başlanır. $s_4$ öğrencisinin ilk tercihi $c_4$'tür. Bu okul $t_2$ tipinde öğrenci kabul ettiği için $s_4$ öğrencisini $c_4$ okuluna geçici olarak atayabiliriz. Okulun kontenjanı 1 azaltılır. $t_2$ tipindeki puan sıralamasına bir sonraki öğrenci $s_6$'dır. $s_6$'nın ilk tercihi olan $c_1$ okulu tip uyuşmazlığı nedeniyle atlanır ve ikinci tercihi olan $c_2$ okuluna öğrenci yine geçici olarak atanır. Üçüncü sıradaki öğrenci $s_1$'in ilk tercihi tip uyuşmazlığı sebebiyle atlanır ve öğrenci ikinci tercihi olan $c_2$ okuluna geçici olarak atanır. Bu atamalardan sonra okulların tamamının kontenjanları dolduğu için ilk tur burada biter. Mekanizmada her tur sonunda atanan öğrencilerin tercihleri güncellenir. Öğrenciler o turda hangi okula atandıysa tercih sıralamasında o okuldan sonra $c_0$ yazılır. Eğer bir öğrenci o turda birden fazla okula yerleştirildiyse daha önce tercih ettiği okuldan sonra $c_0$ yazılır. Birinci tur sonunda güncellenen tercihler listesi:

$P_{s_1} = c_3, c_0$ $\quad$ $P_{s_4} = c_4, c_1, c_0$

$P_{s_2} = c_2, c_4, c_3, c_1$ $\quad$ $P_{s_5} = c_3, c_1, c_2, c_4$

$P_{s_3} = c_1, c_0$ $\quad$ $P_{s_6} = c_1, c_2, c_0$

**Şekil 13**: Örnek 6 – 1. Tur

| | $c_1$ | $c_2$ | $c_3$ | $c_4$ |
|---|---|---|---|---|
| *1.tur* | $s_4, s_3$ | $s_6, s_1$ | $s_1$ | $s_4$ |

İkinci turda ilk olarak birden fazla okula atanan öğrencinin olup olmadığı kontrol edilir. Bu atamada $s_1$ ve $s_4$ öğrencilerinin iki farklı okula atandığı görülmektedir. $s_1$ öğrencisi hem $c_2$ hem de $c_3$ okuluna, $s_4$ öğrencisi ise hem $c_1$ hem de $c_4$ okuluna geçici olarak atanmıştır. Bu durumda öğrencilerin tercih sıralamasına bakılır. Öğrenci atandığı iki okuldan hangisini daha önce tercih etmişse o okula yerleştirilir. Dolayısıyla $s_1$ öğrencisi $c_3$ okuluna, $s_4$ öğrencisi $c_4$ okuluna kalıcı olarak atanır. Bu sebeple öğrenciler ilk başta okullara geçici olarak atanmışlardır. $s_1$ ve $s_4$ öğrencileri listeden çıkartılır. $c_1$ ve $c_2$ okullarının kontenjanları 1 arttırılır. $s_1$ ve $s_4$ öğrencisi kalıcı olarak atandıktan sonra mekanizma aynı şekilde işlemeye devam eder. İkinci turda bütün öğrenciler okullara atanır. Ataması yapılmayan bir öğrenci kalmadığı için ve okulların kontenjanları dolduğu için mekanizma ikinci turda sona ermiştir.

**Şekil 14**: Örnek 6 – Nihai Atamalar

| | $c_1$ | $c_2$ | $c_3$ | $c_4$ |
|---|---|---|---|---|
| *1.tur* | $s_4, s_3$ | $s_6, s_1$ | $s_1$ | $s_4$ |
| *2.tur* | $s_3, s_2$ | $s_6, s_5$ | $s_1$ | $s_4$ |

Nihai atamalar şu şekildedir:

$$\mu_{MCSD} = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 \\ c_3 & c_1 & c_1 & c_4 & c_2 & c_2 \end{pmatrix}$$

$R_{\mu_{MCDS(s_1)}} = 1$, $R_{\mu_{MCDS(s_2)}} = 4$, $R_{\mu_{MCDS(s_3)}} = 1$, $R_{\mu_{MCDS(s_4)}} = 1$, $R_{\mu_{MCDS(s_5)}} = 3$, $R_{\mu_{MCDS(s_6)}} = 2$

### 2.1.6 Boston Öğrenci Atama Mekanizması

Öğrenci atamalarında kullanılan diğer bir mekanizma *Boston öğrenci atama mekanizması*dır. Boston'da 1974 yılında Mahkeme sorumluluğu aldıktan sonra Boston 3 farklı öğrenci atama planını uygulamaya aldı ve 1989 yılında kontrollü seçim öğrenci atamaları planını benimsedi (Willie and Alves, 1996: 47). Son olarak

1999 yılında uygulanmaya başlayan Boston öğrenci atama mekanizması doğrudan bir atama mekanizmasıdır ve okul seçimi probleminin bulunduğu piyasalarda kullanılan bir mekanizmadır. Mekanizma şu şekilde işlemektedir:

1. Her öğrenci okullar için tercih sıralamasını belirtir.
2. Her okul aşağıdaki aşama sıralamasına göre öğrenciler üzerinde önceliğini belirler:
    a. Birinci öncelik: öğrencinin okulda kardeşi olup olmaması ve yürüme mesafesi,
    b. İkinci öncelik: öğrencinin okulda kardeşi olup olmaması,
    c. Üçüncü öncelik: yürüme mesafesi,
    d. Dördüncü öncelik: diğer öğrenciler.

Eğer öğrenciler aynı öncelik grubunda ise, sıralama kuralları önceden belirlenen bir kura sistemine göre belirlenir.

3. Son aşamadır ve atamalar yapılır.

*1.tur*: Birinci turda sadece öğrencilerin birinci tercihleri göz önünde bulundurulur (Abdulkadiroğlu ve Sönmez, 2014: 308). Her okul için o okulu ilk tercih eden öğrenciler dikkate alınır, bu öğrenciler öncelik sıralamasına göre okulların kontenjanlarına, okulların kontenjanları dolana kadar veya herhangi bir okulu ilk tercih olarak belirten öğrenci kalmayıncaya kadar birer birer atanır.

*2.tur*: İkinci turda henüz atanmamış olan öğrencilerin sadece ikinci tercihleri dikkate alınır. Birinci turda kontenjanı dolmamış okullara, okulları ikinci tercihlerinde belirten öğrenciler, okulların kontenjanları dolana kadar veya herhangi bir okulu ikinci tercih olarak belirten öğrenci kalmayıncaya kadar öncelik sıralamasına göre birer birer atanır.

*k.tur*: Geneler olarak, henüz atanmamış olan öğrenciler ve *k.* tercihleri dikkate alınır. Bir önceki turda kontenjanı dolmamış okullara, okulları *k.* tercihlerinde belirten öğrenciler, okulların kontenjanları dolana kadar veya herhangi bir okulu *k.* tercih olarak belirten öğrenci kalmayıncaya kadar öncelik sıralamasına göre birer birer atanır. Mekanizma, tüm öğrencilerin ataması yapılana kadar veya bütün okulların kontenjanları bitene kadar devam eder.

### 2.1.7 Columbus Öğrenci Atama Mekanizması

Columbus öğrenci atama mekanizması da Boston öğrenci atama mekanizması gibi okul seçimi probleminin bulunduğu piyasalarda kullanılan bir mekanizmadır fakat doğrudan bir mekanizma değildir. Columbus okul bölgesinde uygulanmakta olup şu şekilde işlemektedir:

1. Her öğrenci maksimum 3 farklı okula başvurabilir.
2. Bazı okullar için, eğer bir öğrenci kontenjanlarına sürekli yerleştirme yaptığı bir bölgede yaşıyor ise o öğrencinin kontenjanı garantilidir (Abdulkadiroğlu ve Sönmez, 2014: 309). Bu bölge içerisinde ikamet etmeyen öğrenciler için öncelik rastgele bir kura ile belirlenmektedir. Kontenjanı boş kalan okullar için, geriye kalan tüm başvuru sahipleri arasındaki öncelik rastgele bir kurayla belirlenmektedir (Abdulkadiroğlu ve Sönmez, 2014: 309).
3. Her okul için o okulun kontenjanı, o okulda en yüksek önceliğe sahip öğrencilere bir kura ofisi aracılıyla teklif edilir ve teklif alan öğrencinin bu teklifi kabul veya reddetmek için üç günü vardır (Abdulkadiroğlu ve Sönmez, 2014: 309). Teklif almayan başvurular ise bir bekleme listesine alınmaktadır. Eğer teklif alan başvuru sahibi teklifi kabul ederse okulun o kontenjanına atanır ve teklifi kabul ettikten sonra diğer okulların tekliflerini reddetmelidir. Böylece diğer okulların bekleme listesinden çıkartılır. Reddedilen teklifler sonucu kontenjanlarda boşluk oluşur ve kura ofisleri bu kontenjanları bekleme listesindeki diğer öğrencilere teklif ederler (Abdulkadiroğlu ve Sönmez, 2014: 309).

Columbus Öğrenci Atama Mekanizması'nda öğrencilerin optimal başvuru stratejileri belirsizdir (Abdulkadiroğlu ve Sönmez, 2014: 309). Bu mekanizmada aileler karar alma noktasında çok zorlanmaktadır. Çünkü bir öğrenci ikinci veya üçüncü tercihinden teklif aldığında bu teklifi kabul etmenin optimal bir karar olup olmadığından emin değildir. Aynı zamanda öğrenci için başvuracağı okulların optimal bir listesi yoktur. Dolayısıyla aileler bu süreçte zor ve stratejik bir oyun oynamaya itilmektedir.

Columbus Öğrenci Atama Mekanizması'nda verimlilik sorunu mevcuttur. İki öğrenci (a, b) düşünelim ve a öğrencine b öğrencisinin ilk tercihi b öğrencisine ise a

öğrencisinin ilk tercihi olan okul başvuruda bulunmuş olsun. Bu öğrenciler mevcut tekliflerinden daha iyi bir teklif alıp almayacaklarını bilemeyecekleri için teklifi kabul etmeleri onlar için daha iyi olur. Bu nedenle böyle eşleşmeler verimsiz bir atama meydana getirir.

Yukarıda bahsettiğimiz son iki mekanizma okul seçimi problemi için geliştirilen önemli mekanizmalardı. Şimdi son olarak ABD'de liselere geçişte kullanılan fakat okul seçimi probleminin en önemli gerekçelerinden biri olan, okulların sadece tüketilecek bir nesne olması durumuna istisna olan bir atama sisteminden bahsedeceğiz; New York City'de kullanılan liselere yerleştirme mekanizması.

Aslında bu mekanizma New York City'de öğrencilerin liselere girişinde kullanılan bir atama mekanizmasıdır. Fakat mekanizmanın uygulanmasındaki bir takım farklılıklar bu mekanizmanın çift taraflı eşleşme piyasası olarak kabul edilmesine neden olmuştur. Bu mekanizmanın çift taraflı eşleşme piyasası olarak kabul edilmesinin iki temel nedeni vardır. Bunlardan ilki, daha önce okul seçimi bağlamında, öğrencilerin stratejik bir oyuncu ve okulların ise tüketilecek bir nesne olduğunu belirtmiştik. New York City'de kullanılan sistem bu kriter ile uyuşmamaktadır. Bu sistemde okullar önceliklerini kendileri belirleyebilmektedirler. Okullar öğrencilere farklı kriterlere göre öncelik tanıyabilir. Bazı okullar daha yüksek puanlı öğrencileri tercih ederken, diğer okullar derse katılım oranı daha iyi olan öğrencileri tercih edebilirler (Abdulkadiroğlu, Roth ve Pathak, 2005: 365). Dolayısıyla bu sistemde okullar da stratejik birer oyuncu olarak kabul edilmektedir. New York City Atama Mekanizması'nın çift taraflı eşleşme piyasası olarak kabul edilmesinin bir diğer nedeni ise okulların, tercih ettikleri öğrencilerle eşleşme kapasitesine sahip olmamasıdır (Abdulkadiroğlu, Roth ve Pathak, 2005: 365).

### 2.1.8 Türkiye'de Üniversiteye Girişte Kullanılan Mekanizmalar

#### 2.1.8.1 Türkiye'de Uygulanan Mekanizmaların Tarihsel Süreci

Türkiye'de liseye geçiş ve üniversiteye geçişte uygulanan uygulamalar tarihsel gelişimlerini paralel bir şekilde devam ettirmiş ve başvurulan uygulamalar çoğunlukla aynı olmuştur. Bu başlık altında üniversiteye giriş uygulamaları incelenirken aslında her iki öğrenim dereceleri arası geçiş süreci incelenmiş olacaktır.

Türkiye'de yükseköğretime geçişteki uygulamalara bir bütün olarak bakıldığında, 1960 yılı öncesinde üniversite ve yüksekokulların farklı yöntemler uyguladıkları görülür (Eşme, 2014: 149). O dönemde her okul kendine has yöntemlerle öğrenci alımlarını gerçekleştirmiştir. Kontenjanları uygun olan fakülte ve yüksekokullar, sınavsız öğrenci alırken, kontenjanı aşan taleple karşılaşan fakülte ve yüksekokullar kendi şartlarına uygun ölçütleri dikkate almışlardır (Eşme, 2014: 149). Mevcut kontenjanları aşan bir talep ile karşılaşan fakülteler seçme işlemini şu üç esasa göre yapmışlardır:

a) Başvuru sırasına göre kayıt işlemi yapılmış, kontenjanlar dolduktan sonra gelen adaylara "kayıtların kapandığı" ilan edilmiştir.
b) Bazı fakülteler verdikleri eğitime göre, liselerin fen veya edebiyat bölümlerinden mezun olan öğrencileri kaydetmişlerdir.
c) Bazı fakülteler de öğrencilerin lise mezuniyet derecelerini esas alarak, öncelikle pekiyi derecesine sahip öğrencileri, daha sonra da iyi ve orta dereceye sahip adayları kabul etmişlerdir (Kılıç, 1999: 299-300)

Dolayısıyla aslında bu dönem için tam olarak işleyen herhangi tek bir mekanizmadan bahsedebilmek söz konusu değildir. Çoğunlukla mekanizma, kendi kontenjanını aşmayan talep alan okullar için bütün öğrencileri kabul etmek, kontenjanını aşan talep alan okullar için ise kendi belirledikleri kriterlere göre öğrencileri kabul etmek şeklindeydi. Yani işleyiş bir anlamda New York City'de liselere geçiş için kullanılan sisteme benzerdi.

1960'lı yıllarda artan öğrenci sayısı, dolayısıyla artan talep, sebebiyle üniversitelerin yukarıda bahsedilen yöntemleri yetersiz kalmıştır. Bu dönemde üniversiteler öğrencileri kendi yaptıkları sınavlara göre kabul etmeye başlamışlardı. Bu uygulama, öğrencilerin farklı üniversitelerin sınavları için farklı şehirlere seyahat etmek zorunda kalması ve bazı üniversite sınavlarının saatlerinin çakışması sebebi ile tam olarak verimli olamamıştır. Fakat 1960 yılından önce uygulanan uygulamalarda yaşanan bazı sorunlar merkezi sistemin ortaya çıkmasında öncü olan Ankara Üniversitesi Siyasal Bilgiler Fakültesi'nin kullandığı sınav yöntemi ile çözüme kavuşmuştur. Tek kayıt sistemi olan bu uygulamada test yöntemi ile yapılan tek bir giriş sınavı mevcuttu ve öğrenciler bu sınavdan aldıkları puanla, önceden belirttikleri tercihlerine, kontenjan uygunluğuna göre atanmaktaydılar.

Ankara Üniversitesi Siyasal Bilgiler Fakültesi'nin önayak olduğu uygulama zamanla diğer üniversitelere de sıçramıştı. Bu sınav sisteminin diğer üniversitelerde de başarılı bir şekilde uygulanmasının ardından merkezi sınav uygulaması konuşulmaya başlandı. Ve nihayet 1964–65 eğitim öğretim yılında, İTÜ ve ODTÜ dışındaki üniversiteler, ilk kez merkezi sınavla öğrenci aldı (Eşme, 2014: 149).

Uygulama henüz bir tüzel kişiliğe sahip değildi. Bu sebeple farklı üniversiteler sırayla bu uygulamanın sorumluluğunu üstlenmekteydi. Bu mekanizmanın tüm üniversiteleri içeren daha kuvvetli bir merkezi sınav sistemi haline gelmesi adına 19 Kasım 1974 tarihinde Üniversitelerarası Öğrenci Seçme ve Yerleştirme Merkezini (ÜSYM) kurulmuştur.

ÜSYM tarafından uygulanan sınavın adı Üniversitelerarası Seçme ve Yerleştirme Sınavı (ÜSS) olarak belirlendi. Daha sonra Üniversitelerarası Öğrenci Seçme ve Yerleştirme Merkezini (ÜSYM), Öğrenci Seçme ve Yerleştirme Merkezi (ÖSYM) adıyla Yüksek Öğretim Kurulu'nun (YÖK) bünyesine geçti.

#### 2.1.8.2 Üniversiteye Giriş Sınavında Yapılan Değişiklikler

Tarihsel süreçte sınavın uygulanmasında bir takım değişiklikler yapıldı. Daha önce tek aşamalı olarak uygulanan bu sınav sistemi, 1981 sonrası iki aşamalı (Öğrenci Seçme Sınavı (ÖSS) ve Öğrenci Yerleştirme Sınavı (ÖYS)) hale gelmişti ve daha sonra bu iki sınav ayrı ayrı olarak uygulanmaya başladı.

1998 yılında ÖYS'nin kaldırılma kararı alındı, 1999 'dan itibaren üniversiteye girişte ÖSS ele alındı ve yabancı dil esaslı öğrenci alan programlar için ÖSS'den ayrı olarak Yabancı Dil Sınavı (YDS) düzenlenmeye başladı. Yine aynı sene yapılan değişiklik ile öğrenciler sınav sonrası aldıkları puanlara göre farklı puan türlerinde tercihlerini yapabileceklerdi. Bu süreçte Ortaöğretim Başarı Puanı'nın (OBP) yüzdelerinde değişikler yapıldı.

2010 yılından sonra üniversiteye giriş sınavı tekrar iki aşamalı olarak uygulanmaya başlandı. Yükseköğretime Geçiş Sınavı (YGS) adı verilen sınavın ilk aşaması, 2006 öncesindeki ÖSS kapsamında, Lisans Yerleştirme Sınavı (LYS) adı verilen ikinci aşaması beş ayrı oturumda gerçekleşen bir sınav olarak düzenlendi (Eşme, 2014: 150).

#### 2.1.8.3 ÖSYS Mekanizmasının Tanıtılması

Yukarıda anlattığımız bilgiler ışığında Türkiye'de şu anda üniversiteye girişte kullanılan mevcut mekanizmanın özelliklerine bakalım:

a. Öğrenciler üniversiteye girebilmek için merkezi uygulanan bir sınava girmelidir.
b. Sınavda öğrencilerin farklı puanlar alabileceği birden fazla kategori mevcuttur.
c. Öğrenciler sınav sonuçlarına göre, farklı kategori gruplarına göre öğrenci kabul eden üniversiteleri tercih olarak yazabilmektedirler.
d. Üniversitelerin kontenjan sınırı mevcuttur.
e. Üniversiteler, kabul edecekleri öğrenciler için, kendilerini tercih eden öğrencilerin ilgili kategorideki sıralamasına göre öncelik tanımaktadır. En yüksek puanı alan öğrenci en yüksek önceliğe sahiptir.
f. Atamalar sonucunda her öğrenci bir üniversite ile eşleşmelidir.

Uygulanan mekanizmanın özellikleri incelendiğinde, bu mekanizmanın daha önce tanıttığımız Çok Kategorili Seri Diktatörlük Mekanizması olduğunu söyleyebilmekteyiz.

Bunun dışında Türkiye'de kullanılan sistemde öğrencilerin üniversiteye yerleşmesinde kullandıkları puanlarını etkileyen ve farklı başarı durumlarına göre farklı öncelikler tanıyan bir takım özellikleri de mevcuttur. Bunlardan yerleşme puanına etki eden Ortaöğrenim Başarı Puanı (OBP), ortaöğretim başarı durumuna göre hesaplanır, *Ağırlıklı Ortaöğrenim Başarı Puanı* (AOBP) her kategori için hesaplanır ve (AOBP)'nin hesaplanmasında öğrencinin OBP'si ve okulundan sınava katılan öğrencilerin ilgili kategorideki ortalaması kullanılır. Mesleğe yönelik eğitim veren liselerden mezun olan adaylara aynı mesleğe yönelik eğitim veren programlara yerleştirilirken, bu programların öğrenci almakta kullandığı puan türündeki puanlarına eklenmek üzere ek puan verilmektedir (Doğan ve Yuret, 2010: 63). Okullarını birinci olarak bitiren öğrencilere üniversite programları ek kontenjan yaratır. TÜBİTAK yarışmalarında veya spor müsabakalarında başarılı olan öğrencilere farklı öncelikler tanınmaktadır.

# ÜÇÜNCÜ BÖLÜM

# 3. GELİŞTİRİLEN MEKANİZMALARIN DEĞERLENDİRİLMESİ

## 3.1 GALE-SHAPLEY ALGORİTMASI

Gale-Shapley Algoritması'nın uygulama farklılıkları olması ve bu farklı uygulamaların farklı sonuçlar vermesi sebebiyle iki farklı başlık altında değerlendirmesini yapacağız. Bunlar;

*Gale-Shapley Öğrenci Optimal Durağan Mekanizması,*

*Gale-Shapley Okul Optimal Durağan Mekanizması*'dır.

Daha önce Gale-Shapley algoritmasının üniversiteye giriş bağlamındaki durağan bir eşleşmenin okul seçimi bağlamındaki haklı kıskançlığı ortadan kaldırdığını söyledik. Gale-Shapley Öğrenci Optimal Durağan Mekanizması'nın en önemli özelliği haklı kıskançlığı ortadan kaldıran bütün eşleşmeleri Pareto domine etmesidir. Gale-Shapley Öğrenci Optimal Durağan Mekanizması'nın bir cazip özelliği daha vardır: manipüle edilemezdir (Abdulkadiroğlu ve Sönmez, 2014: 305). Gale-Shapley Öğrenci Optimal Durağan Mekanizması'nda, tercihlerin dürüstçe açıklanması baskın stratejidir ve bu nedenle öğrenci ve ailelerin tercih yaparken karmaşık stratejilere başvurması gerekmemektedir (Abdulkadiroğlu ve Sönmez, 2014).

Gale-Shapley Öğrenci Optimal Durağan Mekanizması, gelişmeleri ödüllendiren tek mekanizmadır (Balinski ve Sönmez, 1999). Notlarını geliştiren öğrencilerin daha öncelikli tercihine yerleşme ihtimalini arttırmaktadır. Yani öğrencilerin notlarını yükseltmesini teşvik etmektedir.

Hem adil hem de Pareto optimum olan bir yerleştirme mekanizması yoktur. Bu nedenle Gale Shapley Öğrenci Optimal Mekanizması'na en iyi ikinci mekanizma diyebiliriz çünkü bireysel olarak rasyonel, savurgan olmayan, adil ve manipüle edilemez (strategy-proof) olan tek öğrenci yerleştirme mekanizmasıdır (Balinski ve Sönmez, 1999: 89).

Fakat okul seçimi durumunda Gale-Shapley Öğrenci Optimal Durağan Mekanizması problemsiz değildir (Abdulkadiroğlu ve Sönmez, 2014: 305). Pareto etkinlik haklı kıskançlığın ortadan kaldırılması arasında çelişki bulunduğu için bu

mekanizmanın, haklı kıskançlığı ortadan kaldıran bütün eşleşmeleri Pareto domine etmesine rağmen kendi eşleşmesi Pareto domine edilmiş olabilir. Bu durumu bir örnek üzerinden açıklayalım:

*Örnek 7:* Üç okul $C=\{c_1, c_2, c_3\}$ ve bu okullara atanacak olan üç öğrencinin $S=\{s_1, s_2, s_3\}$ bulunduğu bir örnek düşünelim.

*tercihler*:

$P_{s_1} = c_2, c_1, c_3$

$P_{s_2} = c_1, c_2, c_3$

$P_{s_3} = c_1, c_2, c_3$

$P_{c_1} = s_1, s_3, s_2$

$P_{c_2} = s_2, s_1, s_3$

$P_{c_3} = s_2, s_1, s_3$

**Şekil 15:** Örnek 7 – Nihai Atamalar

| | $c_1$ | $c_2$ | $c_3$ |
|---|---|---|---|
| *1. tur* | $s_2, s_3$ | $s_1$ | |
| *2. tur* | $s_3$ | $s_1, s_2$ | |
| *3. tur* | $s_3, s_1$ | $s_2$ | |
| *4. tur* | $s_1$ | $s_2, s_3$ | |
| *5. tur* | $s_1$ | $s_2$ | $s_3$ |

Bu atamada durağan olan tek eşleşme şu şekildedir:

$$\mu = \begin{pmatrix} s_1 & s_2 & s_3 \\ c_1 & c_2 & c_3 \end{pmatrix}$$

Fakat üstteki eşleşme, aşağıdaki eşleşme tarafından Pareto domine edilmektedir:

$$\gamma = \begin{pmatrix} s_1 & s_2 & s_3 \\ c_2 & c_1 & c_3 \end{pmatrix}$$

Bu örnekte öğrenciler $s_1$ ve $s_2$ sırasıyla $c_1$ ve $c_2$ okullarında en yüksek önceliğe sahiptirler ve öğrenci $s_1$ ve $s_2$'nin tercihlerince okullar $c_1$ ve $c_2$ ilk iki sırada bulunmaktadır. Dolayısıyla öğrenci $s_1$ okul $c_1$'den daha kötü bir tercihe, öğrenci $s_2$

ise okul $c_2$'den daha kötü bir tercihe yerleştirilemez ve bundan dolayı bu iki öğrenci $s_1$ ve $s_2$ okullarına yerleştirilecektir. Böylece, öğrenciler $s_1$ ve $s_2$, okullar $c_1$ ve $c_2$'yi kendi aralarında paylaşmalıdırlar (Abdulkadiroğlu ve Sönmez, 2014: 311). Burada durağan eşleşme, okulların öğrenciler arasında Pareto etkin bir şekilde paylaşılmasına neden olmaktadır. Eğer öğrenciler $s_1$ ve $s_2$, sırasıyla okullar $c_2$ ve $c_1$'e yerleştirilirse, öğrenci $s_3$'ün okul $c_1$'i, yerleştirildiği okul olan $c_3$'e tercih ettiği bir durumla karşılaşırız ve $s_3$, okul $c_1$ için $s_2$'den daha yüksek önceliğe sahiptir. Dolayısıyla burada haklı kıskançlık meydana gelmektedir. Bu durum eşleşmenin durağan olmasına engel olmaktadır.

*Örnek 7*'de görüldüğü üzere haklı kıskançlığın ortadan kaldırılması ile Pareto etkinlik arasında çatışma mevcuttur. Eğer karar alıcılar haklı kıskançlığın tamamen ortadan kaldırılmasına Pareto etkinlikten daha fazla önem verirlerse, Gale-Shapley Öğrenci Optimal Durağan Mekanizması çok uygun bir mekanizmadır (Abdulkadiroğlu ve Sönmez, 2014: 311).

*Tipe-Özel Kotalı Gale-Shapley Öğrenci Optimal Mekanizması*, Gale-Shapley Öğrenci Optimal Durağan Mekanizması'nın aksine manipüle edilebilir bir mekanizmadır. Dolayısıyla atamaların bu mekanizma ile yapılması durumunda öğrenciler ve aileleri stratejik davranarak tercihlerini olduğundan farklı gösterip manipülasyona gidebilirler.

*Gale-Shapley Okul Optimal Durağan Mekanizması* da Tipe-Özel Kotalı Gale-Shapley Öğrenci Optimal Mekanizması gibi manipüle edilebilirdir. Yani öğrenciler tercihlerinde manipülasyona giderek kendilerini daha iyi duruma getirme şansına sahiptirler. Bu durumu bir örnek üzerinden açıklayalım:

*Örnek 8:* Dört okul $C=\{c_1, c_2, c_3, c_4\}$ ve bu okullara atanacak dört öğrencinin $S=\{s_1, s_2, s_3, s_4\}$ bulunduğu bir örnek düşünelim.

*tercihler*:

$P_{s_1} = c_1, c_2, c_3, c_4$ $\qquad$ $P_{s_3} = c_1, c_2, c_4, c_3$

$P_{s_2} = c_1, c_3, c_2, c_4$ $\qquad$ $P_{s_4} = c_3, c_4, c_2, c_1$

$P_{c_1} = s_4, s_3, s_2, s_1$ $\quad\quad$ $P_{c_3} = s_1, s_2, s_4, s_3$

$P_{c_2} = s_4, s_1, s_3, s_2$ $\quad\quad$ $P_{c_4} = s_2, s_1, s_4, s_3$

**Şekil 16**: Örnek 8 – 1. ve 2. Tur

| | $s_1$ | $s_2$ | $s_3$ | $s_4$ |
|---|---|---|---|---|
| *1.tur* | $c_3$ | $c_4$ | | $c_1, c_2$ |
| *2.tur* | $c_3$ | $c_4$ | $c_1$ | $c_2$ |

Bütün öğrencilerin tercihlerini dürüst bir şekilde belirtmesi durumunda oluşan atama şu şekilde;

$$\mu = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 \\ c_3 & c_4 & c_1 & c_2 \end{pmatrix}$$

$R_{\mu(s_1)} = 3$, $R_{\mu(s_2)} = 4$, $R_{\mu(s_3)} = 1$, $R_{\mu(s_4)} = 3$

Şimdi öğrenci $s_4$ manipülasyona giderek tercihlerini değiştirmiş olsun. Diğer öğrencilerin tercihlerinin aynı kaldığını varsayalım. Öğrenci $s_4$'ün yeni tercih sıralaması şu şekilde olsun; $P_{s_4} = c_3, c_4, c_1, c_2$. $s_4$'ün yeni tercih sıralaması ile atamaları tekrardan yapalım.

**Şekil 17**: Örnek 8 – Nihai Atamalar

| | $s_1$ | $s_2$ | $s_3$ | $s_4$ |
|---|---|---|---|---|
| *1.tur* | $c_3$ | $c_4$ | | $c_1, c_2$ |
| *2.tur* | $c_3, c_2$ | $c_4$ | | $c_1$ |
| *3.tur* | $c_2$ | $c_4, c_3$ | | $c_1$ |
| *4.tur* | $c_2, c_4$ | $c_3$ | | $c_1$ |
| *5.tur* | $c_2$ | $c_3$ | | $c_1, c_4$ |
| *6.tur* | $c_2$ | $c_3$ | $c_1$ | $c_4$ |

$$\gamma = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 \\ c_2 & c_3 & c_1 & c_4 \end{pmatrix}$$

$R_{\mu(s_1)} = 2$, $R_{\mu(s_2)} = 2$, $R_{\mu(s_3)} = 1$, $R_{\mu(s_4)} = 2$

Bu durumda yeni eşleşme yukarıdaki gibi olacaktır. Ve $\gamma$ eşleşmesi sonucunda $s_4$ öğrencisi, daha önce tercih sıralamasında üçüncü sırada bulunan $c_2$ okuluna atanmışken, tercihlerinde manipülasyon yaptıktan sonra tercih sıralamasında ikinci

sırada bulunan $c_4$ okuluna atanmıştır. Dolayısıyla tercihlerinde manipülasyona giderek kendini daha iyi bir duruma getirmiştir. Bu değişim de mekanizmanın manipüle edilebilir olduğunu açıkça göstermektedir.

## 3.2 EN YÜKSEK DEĞİŞ TOKUŞ DÖNGÜLERİ MEKANİZMASI

Bir mekanizmanın durağan olmasının haklı kıskançlığı ortadan kaldıracağını daha önce belirtmiştik. Şimdi, önceliklerin Pareto etkinlikle çatışmadığı bir durumu ele alalım: Varsayalım ki okul *c* için öğrenci $s_1$'in önceliğinin öğrenci $s_2$'den daha yüksek olması, öğrenci $s_1$'in okul *c*'de öğrenci $s_2$'den muhakkak öncelikli olarak bir kontenjan hakkı kazanacağı anlamına gelmiyor olsun (Abdulkadiroğlu ve Sönmez, 2014: 311). Öğrenci $s_1$'in önceliğinin öğrenci $s_2$'den daha yüksek olması, daha ziyade öğrenci $s_1$'in okul *c*'ye girebilme fırsatını göstermektedir (Abdulkadiroğlu ve Sönmez, 2014: 306). Diğer şeyler eşitken, eğer $s_1$'in önceliği $s_2$'den daha yüksek ise, okul *c*'ye girmek için daha iyi bir fırsatı vardır (Abdulkadiroğlu ve Sönmez, 2014: 311).

En Yüksek Değiş Tokuş Döngüleri Mekanizması Gale-Shapley Öğrenci Optimal Durağan Mekanizması'ndan farklı olarak Pareto etkindir. Fakat haklı kıskançlığı tamamen ortadan kaldırmayan bir mekanizmadır.

En Yüksek Değiş Tokuş Döngüleri Mekanizması manipüle edilemezdir. Bu mekanizmada öğrencilerin gerçek tercihlerini belirtmeleri baskın stratejidir. Yani örneğin Boston Öğrenci Atama Mekanizması'nda olduğu gibi öğrencilerin önceliklerini kaybetme korkuları yoktur. Dolayısıyla tercihlerini dürüstçe açığa vurabilmektedirler. Ve bu sayede öğrenci ve velileri tercihlerin belirlenmesinde zorlu bir strateji belirleme sürecine katlanmak zorunda değillerdir.

En Yüksek Değiş Tokuş Döngüleri Mekanizması'nda öğrencilerin tercihlerini dürüstçe açığa vurmasının baskın strateji olmasını basitçe şu şekilde açıklayabiliriz; bir öğrencinin tercihlerini dürüstçe belirttiğini ve dürüst olan bu tercihlerine göre yapılan atamada mekanizmanın *n*. basamağında ayrılarak bir okula atandığını varsayalım. Öğrenci *n*. basamakta algoritmadan ayrıldığı için tercihlerinde manipülasyona gitse dahi *n*. basamaktan önceki hiçbir döngüyü değiştiremeyecektir. Yani, öğrenci manipülasyon yapmayı denediğinde sadece kendisine zarar verecektir (Abdulkadiroğlu ve Sönmez, 2014: 313).

### 3.2.1 Tipe Özel Kotalarla En Yüksek Değiş Tokuş Döngüleri Mekanizması

Diğer mekanizmalar gibi Tipe Özel Kotalarla En Yüksek Değiş Tokuş Döngüleri Mekanizması'nda da kontrollü seçim kısıtlamaları uygulanması sonucu verimlilik kayıpları oluşabilir.

Abdulkadiroğlu ve Sönmez'e (2014, s.316) göre "bir eşleştirme, kontrollü seçim kısıtlamalarını karşılayan ve her öğrenciye göreceli daha iyi bir okul ve en az bir öğrenciye kuvvetli bir şekilde daha iyi bir okul atayabilen başka hiçbir eşleştirme yoksa *kısıtlamalı etkin*dir."

Tipe-Özel Kotalı En Yüksek Değiş Tokuş Döngüleri Mekanizması'nda tercihlerin doğru bir şekilde belirtilmesi baskın stratejidir. Aynı zamanda mekanizma kısıtlamalı etkindir (Abdulkadiroğlu ve Sönmez, 2014: 316).

Tipe-Özel Kotalı En Yüksek Değiş Tokuş Döngüleri Mekanizması manipüle edilemezdir (Abdulkadiroğlu ve Sönmez, 2014: 316).

## 3.3 ÇOK KATEGORİLİ SERİ DİKTATÖRLÜK MEKANİZMASI

Seri diktatörlüğün, En Yüksek Değiş Tokuş Döngüleri Mekanizması'nın özel bir durumu olduğunu belirttik. Ve Çok Kategorili Seri Diktatörlük Mekanizması'nın da seri diktatörlüğün özel bir durumu (öğrencilerin birden fazla kategoride olabildiği) olduğunu biliyoruz. Şimdi Çok Kategorili Seri Diktatörlük Mekanizması'nın özelliklerini açıklayacağız.

Çok Kategorili Seri Diktatörlük Mekanizması adil bir mekanizmadır. Ancak bir takım ciddi eksiklikleri vardır, Pareto etkin değildir (Balinski ve Sönmez, 1999: 74). Bu durumu bir örnek ile gösterelim;

*Örnek 9:* İki okul $C = \{c_1, c_2\}$ ve bu okullara ortak bir sınavdan aldıkları puanlara göre atanacak olan iki öğrenci $S = \{s_1, s_2\}$ olsun. Sınavda iki farklı kategori vardır. Öğrencilerin bu iki kategoriden de puanları mevcuttur.

$T=\{t_1, t_2\}$, $t(c_1)= t_1$, $t(c_2)= t_2$

$q = q_{c_1} + q_{c_2}$

$q_{c_1} = 1$, $q_{c_2} = 1$

*Öğrencilerin aldıkları puanlar:*

$f^{s_1} = (f_{t_1}, f_{t_2}) = (70,90)$

$f^{s_2} = (f_{t_1}, f_{t_2}) = (90,70)$

*Her tip için öğrencilerin aldıkları puanlara göre sıralaması:*

| $t_1$ | $t_2$ |
|---|---|
| $s_2$ | $s_1$ |
| $s_1$ | $s_2$ |

*tercihler:*

$P_{s_1} = c_1, c_2$ $\qquad$ $P_{s_2} = c_2, c_1$

**Şekil 18**: Örnek 9 – Nihai Atamalar

| | $c_1$ | $c_2$ |
|---|---|---|
| *1.tur* | $s_2$ | $s_1$ |

$t_1$ kategorisi ile atamalara başlayalım. $t_1$ için puan sıralamasında ilk sırada bulunan öğrenci $s_2$'nin ilk tercihi $c_2$'dir. Fakat $c_2$, $t_2$ olduğu için $s_2$'nin bu tercihi es geçilir ve ikinci tercihi olan $c_1$'e yerleştirilir. $t_2$ kategorisi için atama yaptığımızda ilk sırada bulunan $s_1$'in ilk tercihi $c_1$'dir fakat $c_1$, $t_1$ kategorisinde öğrenci kabul eden okul olduğu için öğrencinin bu tercihi atlanır ve $s_1$ ikinci tercihi olan $c_2$ okuluna yerleştirilir. Atama sonucunda $s_1$ öğrencisi $c_2$ okuluna, $s_2$ öğrencisi $c_1$ okuluna yerleştirilmiştir. Bu atama adildir, fakat Pareto etkin değildir. Çünkü bu atama $s_1$ öğrencisinin $c_1$ okuluna, $s_2$ öğrencisinin de $c_2$ okuluna yerleştirildiği atama tarafından Pareto domine edilmektedir. Çünkü iki öğrenci de önceki atamadan daha iyi duruma gelmiştir.

$$\delta = \begin{pmatrix} s_1 & s_2 \\ c_2 & c_1 \end{pmatrix}$$

Çok Kategorili Seri Diktatörlük Mekanizması adil mekanizmalar arasında en iyi ikinci mekanizma bile değildir.

Çok Kategorili Seri Diktatörlük Mekanizması manipüle edilebilir bir mekanizmadır.

Bu savı *örnek 10* üzerinden açıklayabiliriz.

*Örnek 10:* İki okul $C = \{c_1, c_2\}$ ve bu okullara ortak bir sınavdan aldıkları puanlara göre atanacak olan iki öğrenci $S = \{s_1, s_2\}$ olsun. Sınavda iki farklı kategori vardır. Öğrencilerin bu iki kategoriden de puanları mevcuttur.

$q = q_{c_1} + q_{c_2}$

$q_{c_1} = 1, q_{c_2} = 1$

$T = \{t_1, t_2\}, t(c_1) = t_1, t(c_2) = t_2$

$f^{s_1} = (f_{t_1}, f_{t_2}) = (90,70)$

$f^{s_2} = (f_{t_1}, f_{t_2}) = (70,90)$

*Her tip için öğrencilerin aldıkları puanlara göre sıralaması:*

| $t_1$ | $t_2$ |
|---|---|
| $s_1$ | $s_2$ |
| $s_2$ | $s_1$ |

*tercihler:*

$P_{s_1} = c_2, c_1, c_0$ $\qquad$ $P_{s_2} = c_1, c_2, c_0$

Bu durumda atamalar şu şekilde olmaktadır.

$$\alpha = \begin{pmatrix} s_1 & s_2 \\ c_1 & c_2 \end{pmatrix}$$

Şimdi $s_1$ öğrencisinin tercihlerini yanlış (olduğundan farklı) olarak açıkladığını varsayalım:

$P_{s_1} = c_2, c_0, c_1$

Bu durumda oluşacak atamalar şu şekilde olacaktır:

$$\varepsilon = \begin{pmatrix} s_1 & s_2 \\ c_2 & c_1 \end{pmatrix}$$

Böylece $s_1$ tercihlerini farklı göstererek kendini daha iyi duruma getirmiştir. Bu durum Çok Kategorili Seri Diktatörlük Mekanizması'nın manipüle edilebildiğini açık bir şekilde göstermektedir.

Çok Kategorili Seri Diktatörlük Mekanizması öğrencilerin test puanlarındaki artışlarına saygı göstermez. Yani bir öğrenci, puanını bir veya daha fazla kategoride artırabilir ve yine de diğer her şey sabitken daha kötü bir atama ile cezalandırılabilir. Ya da daha kötü bir performans göstermesine rağmen daha iyi bir tercihine atanabilir. Aşağıdaki örnekte $s_1$ öğrencisinin *Örnek 9*'dan daha kötü puanlar almasına rağmen daha yüksek bir tercihine yerleştiği durum gösterilmiştir.

*Örnek 11:* Örnek 9'daki bilgiler geçerlidir. Şimdi $s_1$ öğrencisinin her iki tipten de daha düşük bir puan aldığını düşünelim. Yeni puanlar;

$f^{s_1} = (f_{t_1}, f_{t_2}) = (50,50)$

$f^{s_2} = (f_{t_1}, f_{t_2}) = (90,70)$

Atamayı gerçekleştirdiğimizde, oluşan eşleşmeye göre $s_1$, bir önceki örneğe göre daha kötü puanlara sahip olmasına rağmen ilk tercihine yerleştirilmiştir.

**Şekil 19**: Örnek 11 – Nihai Atamalar

| | $c_1$ | $c_2$ |
|---|---|---|
| *1.tur* | $s_1$ | $s_2$ |

$\gamma = \begin{pmatrix} s_1 & s_2 \\ c_1 & c_2 \end{pmatrix}$

## 3.4 BOSTON ÖĞRENCİ ATAMA MEKANİZMASI

Boston Öğrenci Atama Mekanizması manipüle edilebilir bir mekanizmadır. Bir öğrenci okul *c* için daha yüksek önceliğe sahip olabilir fakat o okulu ilk tercihi olarak belirtmediği takdirde önceliğini *c*'yi ilk tercihi olarak belirten öğrencilere kaptırmaktadır (Abdulkadiroğlu ve Sönmez, 2014: 308). Dolayısıyla bu mekanizma öğrenci ve velileri, tercihlerini aslında olduğundan farklı belirtmeye itmektedir. Örneğin bir aile bir okulu kendi ilk tercihleri olarak düşünmektedir. Fakat aile bu ilk tercih ettikleri okula fazla talep olabileceğini düşünüp çocuklarının bu okula yerleşemeyebileceğine kanaat getirirse, ilk tercihlerine en yakın olan ikinci tercihlerini ilk tercih olarak belirterek çocuklarının en azından bu okula yerleşmesini sağlamak isteyebilirler. Aksi takdirde ikinci tercih ettikleri okulu ilk tercih olarak belirten öğrencilere kaptırma ihtimalleri vardır. Eğer öğrenciler gerçek tercihlerini

belirtirlerse Boston Öğrenci Atama Mekanizması Pareto etkin olmaktadır (Abdulkadiroğlu ve Sönmez, 2014: 309). Fakat açıkladığımız nedenlerden dolayı aileler tercihlerinde dürüst olmayı tercih etmeyebilirler, dolayısıyla bu durumda mekanizmanın Pareto etkin olması pek muhtemel değildir.

Boston öğrenci atama mekanizmasında önceliklerin belirlenmesi konusunda bir takım sorunlar mevcuttur. Aynı önceliğe sahip iki öğrenci var olabileceği için bu iki öğrencinin önceliklerinin nasıl belirleneceği bir sorun haline gelmiştir. Aynı önceliğe sahip öğrenciler arasında sıralamanın belirlenmesi için en yaygın yöntem olarak başlangıçta sıralamanın rastgele yapılması önerilmiştir. Fakat bu yöntemin etkinlik kaybına yol açacağı Erdil ve Erkan (2008) tarafından gösterilmiştir ve bu etkinlik kaybına engel olmak için *durağan iyileştirme döngüsü* adını verdikleri bir algoritma önermişler. Bu algoritma Pareto etkin eşleşmeyi bulmasına karşın manipüle edilebilmektedir.

Daha sonra bu etkinlik kaybını önlemek için Kesten (2010) *etkinlik-düzeltilmiş gecikmeli kabul mekanizması* (EADAM) ismini verdiği mekanizmayı geliştirmiştir. Fakat Abdulkadiroğlu vd. (2009) mutlak öncelikleri belirlemek için herhangi bir yöntem kullanılarak hem Gale-Shapley gecikmeli kabul algoritmasına hem Pareto baskın olan hem de manipüle edilemez bir mekanizmanın elde edilemeyeceğini göstermişlerdir (Doğan, 2014: 395).

Bunun dışında bütün mekanizmalar için genel bir değerlendirme yapmak gerekirse, geliştirilen mekanizmalarda öğrenci tercihlerinin sınırsız olduğu varsayımı mevcuttur. Fakat bu varsayım doğru değildir. Doğan (2009), ÖSYS'de öğrencilerin tercih bildirimlerinde sayı kısıtlaması olması halinde oluşan eksik bilgili oyunun, dengede durağan olmayan yerleştirmeler doğurabileceğini göstermiştir (Doğan, 2014: 400). Calsamiglia vd. (2010), tercih kısıtlamasının etkinlik bakımından etkilerini deneysel olarak incelemiş ve En Yüksek Değiş Tokuş Döngüleri Mekanizması'nın Gale-Shapley gecikmeli kabul algoritmasına, Gale-Shapley gecikmeli kabul algoritmasının ise Boston Öğrenci Atama Mekanizması'na üstün olduğunu göstermişlerdir.

ÖSYS'nin tercih kısıtlamasından ne kadar etkilendiği Doğan ve Yuret (2010) tarafından incelenmiş ve 2005 yılı yerleştirme sonuçlarına göre dört binden fazla

öğrencide haklı kıskançlığa yol açtığı saptanmıştır. Bu çalışma sonrası o dönem 24 olan tercih sayısının en azından 30 yapılması önerisinde bulunulmuştur. Bu öneriyi dikkate alan ÖSYM 2010 yılında tercih sayısını 24'den 30'a çıkarmıştır

## 3.5 MEKANİZMALARIN AYNI ÖRNEK ÜZERİNDEN ÇÖZÜMLÜ KARŞILAŞTIRILMASI

### 3.5.1 Gale-Shapley Algoritması ve Çok Kategorili Seri Diktatörlük Mekanizması

*Örnek 12:* Dört okul $C=\{c_1, c_2, c_3, c_4\}$ ve bu okullara atanacak olan altı öğrenci $S=\{s_1, s_2, s_3, s_4, s_5, s_6\}$ olsun. Bu örnekte üniversiteye girişlerde tercih sıralaması olarak kullanılabilecek bir sınav uygulaması mevcuttur. Sınavda iki farklı kategori mevcuttur. Bunlar $t_1$ ve $t_2$'dir. Öğrencilerin atamasını Gale-Shapley algoritması ve Çok Kategorili Seri Diktatörlük Mekanizması'na göre yapalım ve sonuçları karşılaştıralım:

$T=\{t_1, t_2\}$, $t(c_1)$ *ve* $t(c_4)= t_1$, $t(c_2)$ *ve* $t(c_3)= t_2$

$q = q_{c_1} + q_{c_2} + q_{c_3} + q_{c_4}$

$q_{c_1} = 2$, $q_{c_2} = 2$, $q_{c_3} = 1$, $q_{c_4} = 1$

*Öğrencilerin aldıkları puanlar:*

$f^{s_1} = (f_{t_1}, f_{t_2}) = (90,90)$

$f^{s_2} = (f_{t_1}, f_{t_2}) = (80,60)$

$f^{s_3} = (f_{t_1}, f_{t_2}) = (70,70)$

$f^{s_4} = (f_{t_1}, f_{t_2}) = (50,80)$

$f^{s_5} = (f_{t_1}, f_{t_2}) = (60,50)$

$f^{s_6} = (f_{t_1}, f_{t_2}) = (65,95)$

*Her tip için öğrencilerin aldıkları puanlara göre sıralaması:*

| $t_1$ | $t_2$ |
|---|---|
| $s_1$ | $s_6$ |
| $s_2$ | $s_1$ |
| $s_3$ | $s_4$ |
| $s_6$ | $s_3$ |
| $s_5$ | $s_2$ |
| $s_4$ | $s_5$ |

*tercihler:*

$P_{s_1} = c_3, c_2, c_1, c_4, c_0$ $\quad$ $P_{s_4} = c_3, c_1, c_2, c_0, c_4$

$P_{s_2} = c_4, c_1, c_3, c_0, c_2$ $\quad$ $P_{s_5} = c_1, c_2, c_3, c_4, c_0$

$P_{s_3} = c_2, c_3, c_4, c_1, c_0$ $\quad$ $P_{s_6} = c_3, c_4, c_2, c_0, c_1$

*Gale-Shapley Öğrenci Optimal Durağan Mekanizması'na göre çözüm:*

Gale-Shapley Öğrenci Optimal Durağan Mekanizması'nda her öğrenci okullara kendi tercih sıralamasına göre herhangi bir okula atanana kadar başvuruda bulunurlar. Öğrenciler için tercih yapma sırası yoktur. Yani her öğrenci aynı anda tercihini yapmaktadır. Bu örnekte okulların öncelik sıralaması her tip için öğrencilerin ayrı olarak oluşturduğu puan sıralaması olmaktadır.

*$t_1$ kategorisindeki tüm okulların tercih sıralaması:*

$P_{c(t_1)} = s_1, s_2, s_3, s_6, s_5, s_4$

*$t_2$ kategorisindeki tüm okulların tercih sıralaması:*

$P_{c(t_2)} = s_6, s_1, s_4, s_3, s_2, s_5$

Örneğin çözümü:

Birinci turda, bütün öğrenciler ilk tercihlerine başvuruda bulunurlar. Bu başvurular sonucunda $s_5$, $c_1$ okuluna, $s_3$, $c_2$ okuluna ve $s_2$, $c_4$ okuluna geçici olarak yerleştirilir. Bu okulların kontenjanları bir azaltılır. Tek kontenjanı olan $c_3$ okuluna ise üç başvuru mevcuttur. $c_3$ okulu, başvuruda bulunan üç öğrenciden kendi tercih sıralamasına göre en yüksek önceliğe sahip olan yani $t_2$ kategorisinde en yüksek puana sahip olan $s_6$ öğrencisinin başvurusunu kabul edip diğer öğrencilerin tekliflerini reddeder. $s_6$ öğrencisi, $c_3$ okuluna geçici olarak atanır ve okulun kontenjanı bir azaltılır.

**Şekil 20**: Örnek 12 – Gale-Shapley Öğrenci Optimal Durağan Mekanizması 1. Tur

| | $c_1$ | $c_2$ | $c_3$ | $c_4$ |
|---|---|---|---|---|
| *1.tur* | $s_5$ | $s_3$ | $s_1, s_4, s_6$ | $s_2$ |

İkinci turda, başvuruları reddedilen $s_1$ ve $s_4$ öğrencileri kendi tercih sıralamalarındaki ikinci tercihlerine başvuruda bulunurlar; $s_1$ öğrencisi $c_2$ okuluna, $s_4$

öğrencisi ise $c_1$ okuluna başvuruda bulunur. Bu turda, her okulun kontenjanı dolmuş olup her öğrenci bir okula atanmıştır. Boşta herhangi bir öğrenci kalmamıştır. Dolayısıyla bu turda atamalar nihai sonuca ulaşmıştır. Öğrenciler geçici olarak atandıkları okullara artık kalıcı olarak atanırlar.

**Şekil 21**: Örnek 12 – Gale-Shapley Öğrenci Optimal Durağan Mekanizması Nihai Atamaları

| | $c_1$ | $c_2$ | $c_3$ | $c_4$ |
|---|---|---|---|---|
| *1.tur* | $s_5$ | $s_3$ | $s_1, s_4, s_6$ | $s_2$ |
| *2.tur* | $s_5, s_4$ | $s_3, s_1$ | $s_6$ | $s_2$ |

$$\mu_{\text{öğrenci optimal}} = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 \\ c_2 & c_4 & c_2 & c_1 & c_1 & c_3 \end{pmatrix}$$

*Gale-Shapley Okul Optimal Durağan Mekanizması'na göre çözüm:*

Gale-Shapley Okul Optimal Durağan Mekanizması'na göre bu örnekte her okul kendi kategorisine ve kontenjan sayısına göre öğrencilere başvuruda bulunurlar. Okulların tercih sıralamalarında öğrencilerin aldıkları puanlar rol oynamaktadır. Her okul kendi kategorisinde en yüksek puana sahip olan öğrenciye ilk teklifini yapacak şekilde bir tercih sıralamasına sahiptir. Eğer puan sıralamasında ilk sıradaki öğrenci o okulu tercih etmezse, okul bir sonraki sırada olan öğrenciye teklif götürecektir. ikinci sıradaki öğrenci de başka okula gitmeyi tercih ederse okul üçüncü sıradaki öğrenciye başvuracaktır. Bu durum okul kontenjanını doldurana kadar ve ya boşta herhangi bir öğrenci kalmayana kadar bu şekilde devam edecektir. Bu bilgiler ışığında örneğin çözümünü yapalım:

Birinci turda, $c_1$ okulu $t_1$ kategorisinde olduğu için bu kategorideki en yüksek puan alan öğrencilere ilk olarak başvuruda bulunacaktır. Kontenjan sayısı iki olduğu için sıralamada ilk iki sırada olan öğrenciye teklif götürür. Bunlar $s_1$ ve $s_2$ öğrencileridir. $c_2$ okulu $t_2$ kategorisinde olduğu için aynı sebeple $s_6$ ve $s_1$ öğrencilerine teklif götürmektedir. $c_3$ okulu, bir kontenjana sahip olduğu için sadece $s_6$ öğrencisine ve $t_1$ kategorisinde olan $c_4$ ise aynı sebeple $s_1$ öğrencisine ilk olarak başvuruda bulunurlar. Bu başvurular sonucuna ilk turda $s_1$ üç teklif, $s_6$ iki teklif ve $s_2$ ise tek teklif alır. $s_2$ öğrencisi $c_1$ okuluna geçici olarak atanır. Birden fazla teklif alan öğrencilerden olan $s_1$ kendi tercih sıralamasına göre en yüksek sıradaki okulu yani $c_2$ okulunun teklifini kabul eder ve diğer okulların

teklifini reddeder. Bu öğrenci $c_2$ okuluna geçici olarak atanır. Birden fazla teklif alan öğrencilerden diğeri olan $s_6$ ise kendine teklif götüren okullar arasından $c_3$ okulunun teklifini kabul eder ve diğer okulları reddeder. Bu okula geçici olarak atanır.

**Şekil 22**: Örnek 12 – Gale-Shapley Okul Optimal Durağan Mekanizması 1. Tur

| | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ | $s_6$ |
|---|---|---|---|---|---|---|
| *1.tur* | $c_1, c_2, c_4$ | $c_1$ | | | | $c_2, c_3$ |

İkinci turda, yaptığı iki tekliften biri reddedilen $c_1$ okulu boş kalan diğer kontenjanını doldurmak için sıralamada bir sonraki öğrenci olan $s_3$'e teklif götürür. Aynı sebeple $c_2$ okulu ise $s_4$ öğrencisine teklif götürür. Teklifi $s_1$ tarafından reddedilen $c_4$, sıralamada bir sonraki sırada olan $s_2$ öğrencisine başvuruda bulunur. Bu aşamada birden fazla teklif alan $s_2$ öğrencisi, $c_1$ okulunun teklifini reddeder ve $c_4$ okulunun teklifini kabul ederek bu okula geçici olarak atanır. $s_3$ öğrencisi $c_1$ okuluna, $s_4$ öğrencisi ise $c_2$ okuluna geçici olarak atanırlar. Atamalar sonucunda okula atanan her öğrenci için o okulun kontenjanı 1 azaltılır.

**Şekil 23**: Örnek 12 – Gale-Shapley Okul Optimal Durağan Mekanizması 1. ve 2. Tur

| | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ | $s_6$ |
|---|---|---|---|---|---|---|
| *1.tur* | $c_1, c_2, c_4$ | $c_1$ | | | | $c_2, c_3$ |
| *2.tur* | $c_2$ | $c_1, c_4$ | $c_1$ | $c_2$ | | $c_3$ |

Üçüncü turda, bir önceki turda teklifi reddedilen $c_1$, sıralamada bir sonraki öğrenciye yani $s_6$ öğrencisine teklif götürür. İki teklif alan $s_6$ öğrencisi kendi tercihlerinde $c_3$ okulu daha öncelikli olduğu için $c_1$ okulunun teklifini reddeder.

**Şekil 24**: Örnek 12 – Gale-Shapley Okul Optimal Durağan Mekanizması 1.– 2. ve 3. Tur

| | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ | $s_6$ |
|---|---|---|---|---|---|---|
| *1.tur* | $c_1, c_4, c_2$ | $c_1$ | | | | $c_2, c_3$ |
| *2.tur* | $c_2$ | $c_1, c_4$ | $c_1$ | $c_2$ | | $c_3$ |
| *3.tur* | $c_2$ | $c_4$ | $c_1$ | $c_2$ | | $c_3, c_1$ |

Dördüncü turda, bir önceki turda teklifi $s_6$ tarafından reddedilen $c_1$, $s_5$ öğrencisine teklifte bulunur. Daha önce hiç teklif almamış olan $s_5$ öğrencisi bu teklifi kabul eder ve geçici olarak bu okula yerleşir. Okulun kontenjanı bir azaltılır. Bu atama sonrası okulların bütün kontenjanları dolmuş ve atanamayan bir öğrenci kalmamıştır. Dolayısıyla atama nihai sonucuna varmıştır.

**Şekil 25**: Örnek 12 – Gale-Shapley Okul Optimal Durağan Mekanizması Nihai Atamalar

| | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ | $s_6$ |
|---|---|---|---|---|---|---|
| *1.tur* | $c_1$, $c_4$, $c_2$ | $c_1$ | | | | $c_2$, $c_3$ |
| *2.tur* | $c_2$ | $c_1$, $c_4$ | $c_1$ | $c_2$ | | $c_3$ |
| *3.tur* | $c_2$ | $c_4$ | $c_1$ | $c_2$ | | $c_3$, $c_1$ |
| *4.tur* | $c_2$ | $c_4$ | $c_1$ | $c_2$ | $c_1$ | $c_3$ |

$$\mu_{okul\ optimal} = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 \\ c_2 & c_4 & c_1 & c_2 & c_1 & c_3 \end{pmatrix}$$

*Çok Kategorili Seri Diktatörlük Mekanizması'na göre çözüm:*

Puan sıralamasına göre iki kategoride de birinci olan öğrencilerin tercihleri ile geçici atamalara başlanır. İlk olarak hangi kategoriden başlandığının bir önemi yoktur. Çünkü her kategori için okullar farklıdır. Ve dolayısıyla kontenjanları ayrıdır. $t_1$ kategorisi ile geçici atamalara başlayalım:

Birinci turda, $t_1$ kategorisinde birinci sırada yer alan $s_1$ öğrencisi ilk tercih hakkına sahiptir. $s_1$ öğrencisinin ilk tercihi olan $c_3$ ve bir sonraki tercihi olan $c_2$ okulları $t_2$ kategorisinde olduğu için bu tercihler atlanır. Bu işlem tercih sıralamasında bulunan $t_1$ kategorisindeki ilk okula kadar devam eder. $s_1$ öğrencisinin üçüncü tercihi olan $c_1$ okulu $t_1$ kategorisindedir ve bu öğrenci $s_1$ okuluna geçici olarak yerleştirilir. $c_1$ okulunun kontenjanı bir düşürülür ($q_{c_1} = 1$). $t_1$ kategorisinde puan sıralamasında ikinci Sıradaki öğrenciye ($s_2$) geçilir. $s_2$ öğrencisinin ilk tercihi $c_4$, $t_1$ kategorisinde olduğu için $s_2$ öğrencisi bu okula geçici olarak atanır. $c_4$ okulunun kontenjanı bir azaltılır ve sıfıra düşer ($q_{c_4} = 0$). Puan sıralamasında bir sonraki öğrenci olan $s_3$, sırasıyla ilk iki tercihi olan $c_2$ ve $c_3$ okulları farklı kategoride olmasından dolayı ve üçüncü tercihi olan $c_4$ okulun da kontenjanı bulunmamasından dolayı es geçildikten sonra dördüncü

tercihi olan $c_1$ okuluna geçici olarak atanır. $c_1$ okulunun kontenjanı bir azaltılır ve sıfıra düşer ($q_{c_1} = 0$). $t_1$ kategorisindeki tüm okulların kontenjanı dolduğu için $t_2$ kategorisine geçilir. $t_2$ kategorisinde birinci sırada yer alan $s_6$ ilk tercihi olan $c_3$ okuluna yerleştirilir. $c_3$ okulunun kontenjanı sıfıra düşer ($q_{c_3} = 0$). $t_2$ kategorisi puan sıralamasında ikinci sırada olan $s_1$ ilk tercihi olan $c_3$ okuluna okulun kontenjanı dolu olduğu için atanamaz. $s_1$ tercih sıralamasında bir sonraki sırada olan $c_2$ okuluna geçici olarak atanır ve okulun kontenjanı bir azaltılır ($q_{c_2} = 1$). Sıra $s_4$ öğrencisine gelir. $s_4$ öğrencisi ilk tercihine o okulun kontenjanı dolu olduğu için atanamaz. İkinci tercihi ($c_1$) ise kategori farklı olduğu için es geçilir. Öğrenci üçüncü tercihi olan $c_2$ okuluna geçici olarak atanır. Okulun kontenjanı bir eksilerek sıfıra düşer ($q_{c_2} = 0$). Geçici atamalar tüm okulların kontenjanları dolu olduğu için sonlanmıştır. Geçici atamalar sonucunda birden fazla okula atanan öğrenci olup olmadığı kontrol edilir. $s_1$ öğrencisi hem $c_1$ hem de $c_2$ okullarına geçici olarak atanmıştır. Birinci tur sonunda güncellenen tercih listeleri:

$P_{s_1} = c_3, c_2, c_0$ $\quad$ $P_{s_4} = c_3, c_1, c_2, c_0$

$P_{s_2} = c_4, c_0$ $\quad$ $P_{s_5} = c_1, c_2, c_3, c_4, c_0$

$P_{s_3} = c_2, c_3, c_4, c_1, c_0$ $\quad$ $P_{s_6} = c_3, c_0$

**Şekil 26**: Örnek 12 – Çok Kategorili Seri Diktatörlük Mekanizması 1. Tur

| | $c_1$ | $c_2$ | $c_3$ | $c_4$ |
|---|---|---|---|---|
| *1.tur* | $s_1, s_3$ | $s_1, s_4$ | $s_6$ | $s_2$ |

İkinci turda, $s_1$ öğrencisinin iki okul arasından hangisini daha önce tercih ettiği kontrol edilir. Öğrenci bu iki okul arasından hangisini daha önce tercih etti ise o okula atanır. $s_1$ öğrencisi $c_2$ okulunu daha önce tercih ettiği için o okula atanır. $c_1$ okulunun kontenjanı bir arttırılır ($q_{c_1} = 1$). $t_1$ kategorisindeki bir okulun kontenjanında yer açıldığı için $t_1$ kategorisinde yeniden atama yapılır. Puan sıralamasına göre ikinci sıradaki $s_2$ öğrencisi kontrol edilir. Bu öğrenci ilk tercihine yerleştirildiği için atlanır. Çünkü daha iyi duruma gelemeyecektir. Bir sonraki sırada olan $s_3$ öğrencisi zaten $c_1$ okulunda olduğu için ve sıralamada bir sonraki öğrenci $s_6$ öğrencisi de ilk tercihine atandığı için bu öğrenciler için tekrar bir atama gerekmez. Sıra $s_5$ öğrencisindedir. $s_5$ öğrencisinin ilk tercihi $c_1$

okuludur ve bu okulun kontenjanı mevcuttur. Dolayısıyla $s_1$ öğrencisi $c_1$ okuluna atanır. Okulun kontenjanı bir azaltılır ($q_{c_1} = 0$). Bu atamadan sonra bütün öğrenciler atanmış olup birden fazla okula atanan herhangi bir öğrenci bulunmadığı için atama nihai sonucuna ulaşmıştır.

**Şekil 27**: Örnek 12 – Çok Kategorili Seri Diktatörlük Mekanizması Nihai Atamalar

| | $c_1$ | $c_2$ | $c_3$ | $c_4$ |
|---|---|---|---|---|
| *1.tur* | $s_1, s_3$ | $s_1, s_4$ | $s_6$ | $s_2$ |
| *2.tur* | $s_3, s_5$ | $s_1, s_4$ | $s_6$ | $s_2$ |

$$\mu_{MCSD} = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 \\ c_2 & c_4 & c_1 & c_2 & c_1 & c_3 \end{pmatrix}$$

Aynı örneği üç farklı mekanizma ile çözümü sonrası atama sonuçlarını karşılaştırırsak Çok Kategorili Seri Diktatörlük Mekanizması ve Gale-Shapley Okul Optimal Durağan Mekanizması atama sonuçlarının tamamen aynı olduğunu görebilmekteyiz. Bu sonuçlara göre $s_1$ öğrencisi, $c_2$ okuluna, yani ikinci tercihine, $s_2$ öğrencisi, $c_4$ okuluna, birinci tercihine, $s_3$ öğrencisi, $c_1$ okuluna, dördüncü tercihine, $s_4$ öğrencisi, $c_2$ okuluna, üçüncü tercihine, $s_5$ öğrencisi $c_1$ okuluna, ilk tercihine ve $s_6$ öğrencisi, $c_3$ okuluna, ilk tercihine yerleşmiştir. Gale-Shapley Öğrenci Optimal Durağan Mekanizması'na göre yapılan atamalara baktığımızda; $s_1$ öğrencisi, $c_2$ okuluna, ikinci tercihine, $s_2$ öğrencisi, $c_4$ okuluna, ilk tercihine, $s_3$ öğrencisi, $c_2$ okuluna, ilk tercihine, $s_4$ öğrencisi, $c_1$ okuluna, ikinci tercihine, $s_5$ öğrencisi, $c_1$ okuluna, ilk tercihine ve $s_6$ öğrencisi, $c_3$ okuluna, ilk tercihine atanmıştır.

$R_{\mu_{\text{öğrenci optimal}(s_1)}} = 2$, $R_{\mu_{\text{öğrenci optimal}(s_2)}} = 1$, $R_{\mu_{\text{öğrenci optimal}(s_3)}} = 1$, $R_{\mu_{\text{öğrenci optimal}(s_4)}} = 2$, $R_{\mu_{\text{öğrenci optimal}(s_5)}} = 1$, $R_{\mu_{\text{öğrenci optimal}(s_6)}} = 1$

$R_{\mu_{okul\ optimal(s_1)}} = 2$, $R_{\mu_{okul\ optimal(s_2)}} = 1$, $R_{\mu_{okul\ optimal(s_3)}} = 4$, $R_{\mu_{okul\ optimal(s_4)}} = 3$, $R_{\mu_{okul\ optimal(s_5)}} = 1$, $R_{\mu_{okul\ optimal(s_6)}} = 1$

$R_{\mu_{MCSD(s_1)}} = 2$, $R_{\mu_{MCSD(s_2)}} = 1$, $R_{\mu_{MCSD(s_3)}} = 4$, $R_{\mu_{MCSD(s_4)}} = 3$, $R_{\mu_{MCSD(s_5)}} = 1$, $R_{\mu_{MCSD(s_6)}} = 1$

Yukarıda da görülebildiği üzere Çok Kategorili Seri Diktatörlük Mekanizması ve Gale-Shapley Okul Optimal Durağan Mekanizması'na göre yapılan atamalarda $s_3$ öğrencisi dördüncü tercihine, $s_4$ öğrencisi ise üçüncü tercihine yerleşebilmişken,

Gale-Shapley Öğrenci Optimal Durağan Mekanizması eşleşmesinde $s_3$ öğrencisi ilk tercihine ve $s_4$ öğrencisi ise ikinci tercihine yerleşmiştir. Her iki öğrenci de Gale-Shapley Öğrenci Optimal Durağan Mekanizması'na göre yapılan atamalarda diğer iki mekanizmada atandıkları okullardan daha fazla tercih ettikleri okullara yerleşmişlerdir. Ve diğer öğrencilerin atama sonuçlarını karşılaştırdığımızda diğer iki mekanizmadaki atamalarından daha kötü durumda olan herhangi bir öğrenci olmadığını görebilmekteyiz. Dolayısıyla Gale-Shapley Öğrenci Optimal Durağan Mekanizması'nın diğer mekanizmaları Pareto domine ettiğini örnek üzerinden kolayca görebilmekteyiz.

### 3.5.2 Gale-Shapley Öğrenci Optimal Durağan Mekanizması ve En Yüksek Değiş Tokuş Döngüleri Mekanizması

En Yüksek Değiş Tokuş Döngüleri Mekanizması'nın Pareto etkin dolduğunu ve hatta bazı eşleşmelerde diğerki mekanizmaları Pareto domine ettiğini bilmekteyiz. TTC'nin eşleşmelerinin, Gale-Shapley Öğrenci Optimal Durağan Mekanizması'nı Pareto domine etmesine rağmen en iyi ikinci mekanizma olarak Gale-Shapley öğrenci optimal durağan mekanizmasını kabul etmemizin sebebi, TTC'nin her eşleşmede haklı kıskançlığı ortadan kaldıramadığı için her zaman durağan eşleşme oluşturamamasıdır. Aşağıdaki örnek üzerinden bu durumu gösterelim.

*Örnek 13:* Altı okul $C=\{c_1, c_2, c_3, c_4, c_5, c_6\}$ ve bu okullara atanacak olan sekiz öğrenci $S=\{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8\}$ olsun. Bilgiler şu şekildedir;

$q_{c_1} = q_{c_2} = q_{c_3} = q_{c_4} = q_{c_5} = q_{c_6} = 2$

*tercihler:*

$P_{s_1} = c_1, c_2, c_3, c_4, c_5, c_6$ $\quad$ $P_{s_5} = c_5, c_2, c_1, c_4, c_3, c_6$

$P_{s_2} = c_1, c_4, c_6, c_3, c_2, c_5$ $\quad$ $P_{s_6} = c_6, c_4, c_5, c_1, c_2, c_3$

$P_{s_3} = c_5, c_6, c_4, c_1, c_3, c_2$ $\quad$ $P_{s_7} = c_6, c_2, c_3, c_4, c_5, c_1$

$P_{s_4} = c_5, c_4, c_3, c_2, c_1, c_6$ $\quad$ $P_{s_8} = c_3, c_1, c_2, c_4, c_6, c_5$

$P_{c_1} = s_3, s_4, s_5, s_8, s_1, s_2, s_7, s_6$ $\quad$ $P_{c_4} = s_7, s_5, s_3, s_2, s_1, s_8, s_4, s_6$

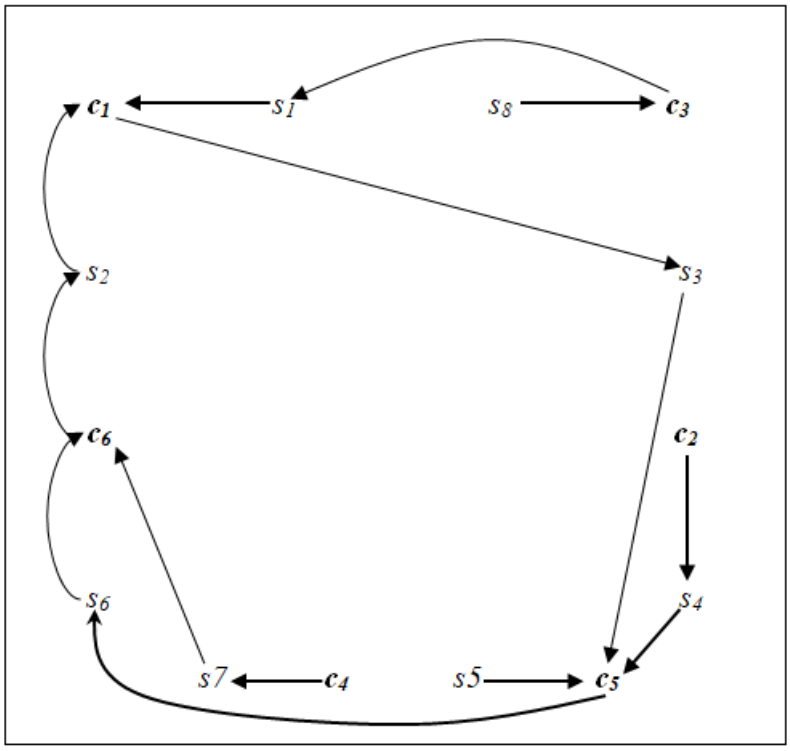
$P_{c_2} = s_4, s_7, s_8, s_1, s_6, s_2, s_3, s_5$ $\quad$ $P_{c_5} = s_6, s_1, s_5, s_4, s_7, s_8, s_3, s_2$

$P_{c_3} = s_1, s_2, s_3, s_6, s_5, s_4, s_7, s_8$ $\quad$ $P_{c_6} = s_2, s_5, s_8, s_1, s_7, s_6, s_4, s_3$

*En Yüksek Değiş Tokuş Döngüleri Mekanizması'na göre çözüm:*

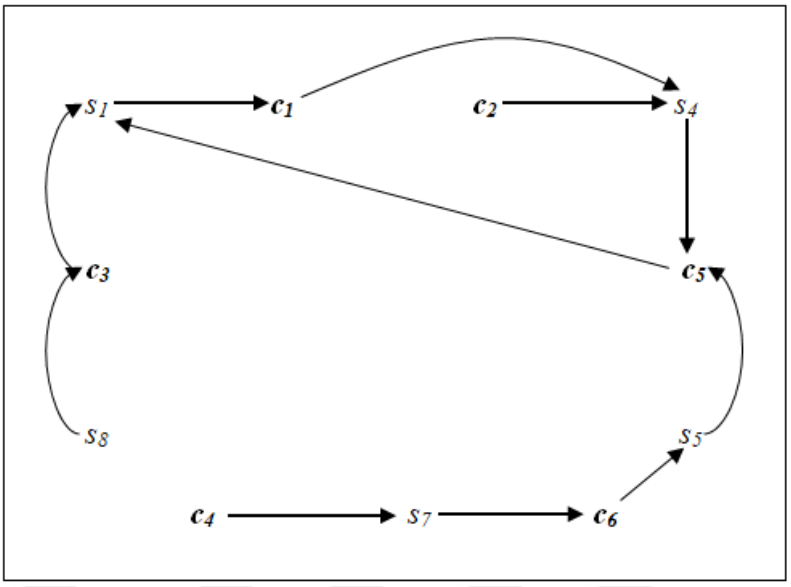
Birinci turda, her okul ve öğrenci kendi ilk tercihini belirtecektir. Bu durum aşağıdaki şekilde gösterilmektedir.

**Şekil 28**: Örnek 13 – En Yüksek Değiş Tokuş Döngüleri Mekanizması 1. Tur



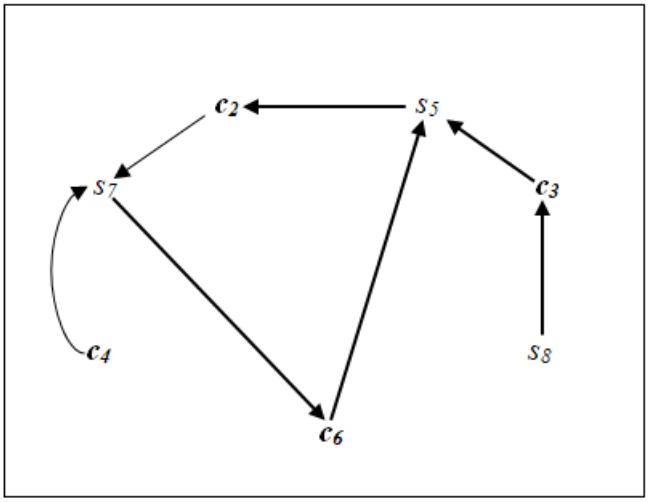

Her okul ve öğrencinin ilk tercihini gösterdiği birinci turda tek bir döngü ($s_2 \rightarrow c_1 \rightarrow s_3 \rightarrow c_5 \rightarrow s_6 \rightarrow c_6 \rightarrow s_2$) vardır. Bu döngü sonrası $s_2$, $c_1$ okuluna, $s_3$, $c_5$ okuluna ve $s_6$, $c_6$ okuluna atanır. Bu üç öğrenci döngüden çıkartılır ve kontenjanlar $q_{c_1} = 1$, $q_{c_5} = 1$, $q_{c_6} = 1$ şeklinde düzenlenir. Bir sonraki tura geçilir.

**Şekil 29**: Örnek 13 – En Yüksek Değiş Tokuş Döngüleri Mekanizması 2. Tur



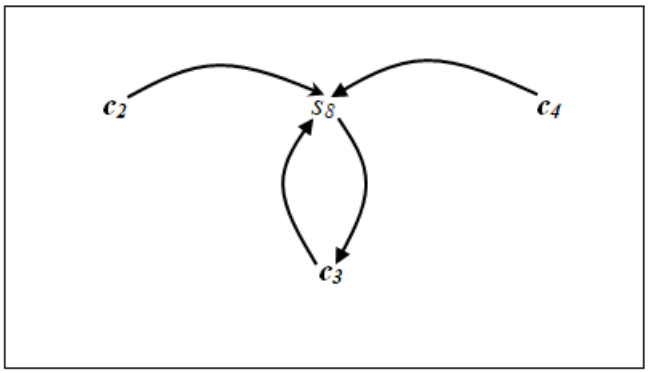

İkinci turda da tek bir döngü ($s_1 \rightarrow c_1 \rightarrow s_4 \rightarrow c_5 \rightarrow s_1$) vardır. Bu döngü sonrası $s_1$, $c_1$ okuluna ve $s_4$, $c_5$ okuluna atanır. Bu üç öğrenci döngüden çıkartılır ve kontenjanlar $q_{c_1} = 0$, $q_{c_5} = 0$ şeklinde düzenlenir. $c_1$ ve $c_5$ okullarının kontenjanı kalmadığı için döngüden çıkartılırlar. Bu işlemden sonra bir sonraki tura geçilir.

**Şekil 30**: Örnek 13 – En Yüksek Değiş Tokuş Döngüleri Mekanizması 3. Tur




Bu turda oluşan tek döngü ($s_5 \rightarrow c_2 \rightarrow s_7 \rightarrow c_6 \rightarrow s_5$) döngüsüdür. Her öğrenci işaret ettiği okula atanır. Dolayısıyla $s_5$, $c_2$ okuluna ve $s_7$, $c_6$ okuluna yerleştirilir, $s_5$ ve $s_7$ öğrencileri döngüden çıkartılır ve atandıkları okulların kontenjanları bir azaltılır. $q_{c_2} = 1$, $q_{c_6} = 0$'dır ve $c_6$ okulu döngüden çıkartılır. Bir sonraki tura geçilir.

**Şekil 31**: Örnek 13 – En Yüksek Değiş Tokuş Döngüleri Mekanizması 4. Tur




Bu turda bir okula atanamayan bir tek $s_8$ kalmıştır. Kontenjanı dolmamış her okul $s_8$ öğrencisini işaret ederken $s_8$ ise ilk tercihi olan $c_3$'ü kontenjanı dolmadığı için işaret eder. Dolayısıyla $s_8$, $c_3$ okuluna atanır ve döngüden çıkarılır. Okulun kontenjanı bir azaltılır. Bu atama sonrası atanamayan hiçbir öğrenci kalmadığı için mekanizma sona erer.

$$\mu_{TTC} = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 \\ c_1 & c_1 & c_5 & c_5 & c_2 & c_6 & c_6 & c_3 \end{pmatrix}$$

$R_{\mu_{TTC}(s_1)} = 1$, $R_{\mu_{TTC}(s_2)} = 1$, $R_{\mu_{TTC}(s_3)} = 1$, $R_{\mu_{TTC}(s_4)} = 1$, $R_{\mu_{TTC}(s_5)} = 2$, $R_{\mu_{TTC}(s_6)} = 1$, $R_{\mu_{TTC}(s_7)} = 1$, $R_{\mu_{TTC}(s_8)} = 1$

*Gale-Shapley Öğrenci Optimal Durağan Mekanizması:*

İlk turda bütün öğrenciler ilk tercihlerine başvuruda bulunur. Dolayısıyla $s_1$ ve $s_2$, $c_1$ okuluna, $s_3$, $s_4$ ve $s_5$, $c_5$ okuluna, $s_6$ ve $s_7$, $c_6$ okuluna, $s_8$ ise $c_3$ okuluna başvururlar. Bu turda üç teklif alan $c_5$, kontenjan kısıtı ($q_{c_2}$=2) sebebiyle bu üç öğrenciden ikisinin teklifini geçici olarak kabul edip diğer öğrencinin teklifini reddetmek zorundadır. $P_{c_5}$'e göre $c_5$, $s_3$ öğrencisinin teklifini reddeder ve diğer iki öğrencinin teklifini geçici olarak kabul eder. $s_3$ dışındaki bütün öğrenciler başvurdukları okullara geçici olarak atanır.

**Şekil 32**: Örnek 13 – Gale-Shapley Öğrenci Optimal Durağan Mekanizması 1. Tur

| | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ |
|---|---|---|---|---|---|---|
| *1. tur* | $s_1, s_2$ | | $s_8$ | | $s_3, s_4, s_5$ | $s_6, s_7$ |

İkinci turda, teklifi reddedilen $s_3$, ikinci tercihi olan $c_6$ okuluna başvuruda bulunur. Üç teklif alan $c_6$, öncelik sıralamasına göre $s_6$ ve $s_7$ öğrencilerinin başvurularını kabul ederken $s_3$ öğrencisinin başvurusunu reddeder.

**Şekil 33**: Örnek 13 – Gale-Shapley Öğrenci Optimal Durağan Mekanizması 1. ve 2. Tur

| | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ |
|---|---|---|---|---|---|---|
| *1. tur* | $s_1, s_2$ | | $s_8$ | | $s_3, s_4, s_5$ | $s_6, s_7$ |
| *2. tur* | $s_1, s_2$ | | $s_8$ | | $s_4, s_5$ | $s_6, s_7, s_3$ |

Üçüncü turda, bir önceki turda tekrar teklifi reddedilen $s_3$, bu sefer üçüncü tercihi olan $c_4$ okuluna başvuruda bulunur. $c_4$, $s_3$ öğrencisinin başvurusunu kabul eder ve öğrenci geçici olarak bu okula atanır. Her öğrenci bir okula yerleştiği ve hiçbir okulun kontenjan kısıtı aşılmadığı için bu turda mekanizma sona erer.

**Şekil 34**: Örnek 13 – Gale-Shapley Öğrenci Optimal Durağan Mekanizması Tüm Turlar ve Nihai Atamalar

| | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ |
|---|---|---|---|---|---|---|
| *1. tur* | $s_1, s_2$ | | $s_8$ | | $s_3, s_4, s_5$ | $s_6, s_7$ |
| *2. tur* | $s_1, s_2$ | | $s_8$ | | $s_4, s_5$ | $s_6, s_7, s_3$ |
| *3. tur* | $s_1, s_2$ | | $s_8$ | $s_3$ | $s_4, s_5$ | $s_6, s_7$ |

Üçüncü turda öğrencilerin geçici olan atamaları nihai hale gelir.

$$\mu_{\text{öğrenci optimal}} = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 \\ c_1 & c_1 & c_4 & c_5 & c_5 & c_6 & c_6 & c_3 \end{pmatrix}$$

$R_{\mu_{\text{öğrenci optimal}(s_1)}} = 1$, $R_{\mu_{\text{öğrenci optimal}(s_2)}} = 1$, $R_{\mu_{\text{öğrenci optimal}(s_3)}} = 3$, $R_{\mu_{\text{öğrenci optimal}(s_4)}} = 1$, $R_{\mu_{\text{öğrenci optimal}(s_5)}} = 2$, $R_{\mu_{\text{öğrenci optimal}(s_6)}} = 1$, $R_{\mu_{\text{öğrenci optimal}(s_7)}} = 1$, $R_{\mu_{\text{öğrenci optimal}(s_8)}} = 1$

İki mekanizmanın atamaları karşılaştırıldığında TTC'nin, Gale-Shapley Öğrenci Optimal Durağan Mekanizması'nı Pareto domine ettiğini görebilmekteyiz. Çünkü $\mu_{TTC}$'ye göre tek bir öğrenci hariç bütün öğrenciler ilk tercihlerine yerleşmişlerdir. $s_5$ ise ikinci tercihine yerleşmiştir. $\mu_{\text{öğrenci optimal}}$'e göre ise yine tek bir öğrenci hariç bütün öğrenciler ilk tercihine yerleşmiş olup, $s_3$ öğrencisi ise üçüncü tercihine yerleşmiştir. Bu da $\mu_{TTC}$'nin öğrenciler için $\mu_{\text{öğrenci optimal}}$'a göre daha iyi bir atama gerçekleştirdiğini göstermektedir. Fakat $\mu_{TTC}$'de ikinci tercihine yerleşen $s_5$

öğrencisine baktığımızda ilk tercihi olan $c_5$'e yerleşemediğini ve bu okula $s_3$ ve $s_4$ öğrencilerinin yerleştiğini biliyoruz. Ancak $P_{c_5}$'e göre, $s_5 \psi_{c_5} s_3$ ve $s_5 \psi_{c_5} s_4$ olduğunu ve $P_{s_5}$'göre $c_5 \psi_{s_5} c_2$ olduğunu görmekteyiz. Dolayısıyla $\mu_{TTC}$'nin haklı kıskançlığa yol açtığı ve durağanlıktan uzaklaştığı açıkça görülmektedir.

# DÖRDÜNCÜ BÖLÜM

# 4. GELİŞTİRİLEN PYTHON ALGORİTMASI İLE MEKANİZMALARIN ÇOK KATILIMCILI ÖRNEKLERİNİN ÇÖZÜMÜ

## 4.1 MEKANİZMALARIN EŞLEŞME SONUÇLARININ GÖSTERİMİ

Aşağıda dört farklı mekanizma için Pyhton programlama dili ile geliştirdiğimiz algoritma ile eşleşmelerin otomatik olarak yapıldığı örnekler mevcuttur. Bu mekanizmalar: Gale-Shapley Öğrenci Optimal Durağan Mekanizması, Gale-Shapley Okul Optimal Durağan Mekanizması, Çok Kategorili Seri Diktatörlük Mekanizması ve En Yüksek Değiş Tokuş Döngüleri Mekanizması'dır. Örneklerdeki öğrenci ve okul sayıları çalışmanın bütünlüğünü bozmaması adına sembolik olarak belirlenmiştir. Fakat her mekanizma için bu sayılar istenilen boyutlarda arttırılıp azaltılabilmektedir. Algoritma istenilen boyuttaki eşleşmeleri yapabilmektedir.

### 4.1.1 Gale-Shapley Öğrenci Optimal Durağan Mekanizması Python Algoritması Örneği

Bu örnekte algoritma öğrenci ve okul tercihlerini ve okul kotalarını (kontenjanlar) rastgele belirlemiştir. Ancak belirli bir piyasadaki gerçek tercih ve kota bilgileri algoritmaya işlenerek eşleşmeler yapılabilmektedir. Bir öğrenci veya okulun herhangi bir öğrenci-okul eşleşme piyasasına dahil olabilmesi için en az bir tercih yapması gerekmektedir. Dolayısıyla sadece tercih yapan öğrencilerin piyasaya dahil olması amacıyla algoritmada, teoride öğrencilerin son tercihinden bir sonraki sıraya yerleştirdiğimiz $c_0$, herhangi bir öğrencinin tercih sıralamasında ilk tercihe gelmeyecek şekilde ayarlanmıştır. Aynı şekilde piyasaya sadece en az bir öğrenciyi kabul eden okullar dahil olabilmesi için teoride okulların son tercihinden bir sonraki sıraya yerleştirdiğimiz $s_0$, herhangi bir okulun tercih sıralamasında ilk tercihe gelmeyecek şekilde ayarlanmıştır.

Örnekte otuz öğrenci ve sekiz okulun bulunduğu bir piyasadaki öğrenci-okul eşleşmesi yapılmıştır.

*Örnek 14*: Piyasa bilgileri Ekler kısmında tablolar halinde verilmiştir. Bakınız EK-4.1 ve EK-4.2.

Eşleşme sonucu turlar halinde Tablo 1'de gösterilmiştir. Her öğrencinin yanında bulunan değer o öğrencinin geçici veya nihai olarak atandığı okulu kaçıncı

sırada tercih ettiğini göstermektedir. Eğer değer “0” ise o öğrencinin o okula atanmak yerine boşta kalmayı tercih ettiğini belirtmektedir. Eşleşme on beşinci turda sona ermiştir. Tabloda bir öğrenci on beşinci turda hangi okulun sütununa yazılmış ise nihai olarak o okula atanmıştır. Eşleşme sonucunda oluşan atamalar şu şekildedir:

$$\mu_{\text{öğ}renci\ optimal} = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 & s_9 & s_{10} & s_{11} & s_{12} & s_{13} & s_{14} & s_{15} & s_{16} & s_{17} & s_{18} & s_{19} & s_{20} \\ c_1 & c_4 & - & - & c_8 & c_7 & - & c_3 & c_3 & - & c_1 & c_4 & c_7 & - & c_6 & c_7 & c_6 & c_2 & c_3 & c_3 \end{pmatrix}$$

$$\begin{pmatrix} s_{21} & s_{22} & s_{23} & s_{24} & s_{25} & s_{26} & s_{27} & s_{28} & s_{29} & s_{30} \\ c_8 & c_7 & - & c_1 & c_7 & c_3 & c_1 & c_1 & - & c_4 \end{pmatrix}$$

**Tablo 1:** Örnek 14 - Turlar ve Eşleşme Sonucu

| Okullar / Tur Sayısı | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ | $c_7$ | $c_8$ |
|---|---|---|---|---|---|---|---|---|
| **1** | [] | [('$s_5$', 1), ('$s_{25}$', 1)] | [('$s_{15}$', 1), ('$s_{24}$', 1), ('$s_{27}$', 1), ('$s_{26}$', 1), ('$s_8$', 1), ('$s_9$', 1), ('$s_{17}$', 1)] | [('$s_{10}$', 1), ('$s_{11}$', 1), ('$s_{12}$', 1), ('$s_{28}$', 1)] | [('$s_1$', 1), ('$s_4$', 1), ('$s_{16}$', 1), ('$s_{19}$', 1), ('$s_{21}$', 1)] | [('$s_{18}$', 1), ('$s_{23}$', 1)] | [('$s_{22}$', 1), ('$s_{13}$', 1), ('$s_6$', 1), ('$s_{29}$', 1), ('$s_7$', 1), ('$s_3$', 1)] | [('$s_2$', 1), ('$s_{30}$', 1), ('$s_{14}$', 1), ('$s_{20}$', 1)] |
| **2** | [('$s_{16}$', 2), ('$s_2$', 2), ('$s_{24}$', 2)] | [('$s_3$', 2), ('$s_{19}$', 2), ('$s_{10}$', 2)] | [('$s_4$', 2), ('$s_{26}$', 1), ('$s_8$', 1), ('$s_9$', 1), ('$s_{17}$', 1)] | [('$s_{12}$', 1), ('$s_{28}$', 1), ('$s_{18}$', 2)] | [('$s_{20}$', 2)] | [('$s_{11}$', 2), ('$s_{15}$', 2), ('$s_1$', 2), ('$s_{23}$', 1)] | [('$s_{22}$', 1), ('$s_{25}$', 2), ('$s_{13}$', 1), ('$s_6$', 1), ('$s_{29}$', 1), ('$s_7$', 1), ('$s_{27}$', 2)] | [('$s_5$', 2), ('$s_{21}$', 2), ('$s_{30}$', 1), ('$s_{14}$', 1)] |
| **3** | [('$s_{30}$', 2), ('$s_{27}$', 3), ('$s_{24}$', 2)] | [('$s_{20}$', 3), ('$s_{19}$', 2)] | [('$s_{14}$', 2), ('$s_{26}$', 1), ('$s_8$', 1), ('$s_9$', 1), ('$s_7$', 2), ('$s_{17}$', 1)] | [('$s_{12}$', 1), ('$s_{28}$', 1), ('$s_{23}$', 2), ('$s_{18}$', 2)] | [] | [('$s_2$', 3), ('$s_{15}$', 2), ('$s_1$', 2), ('$s_{16}$', 3)] | [('$s_{10}$', 3), ('$s_{22}$', 1), ('$s_{25}$', 2), ('$s_{13}$', 1), ('$s_6$', 1), ('$s_{29}$', 1), ('$s_4$', 3)] | [('$s_5$', 2), ('$s_{21}$', 2), ('$s_{11}$', 3)] |
| **4** | [('$s_{27}$', 3), ('$s_{24}$', 2)] | [('$s_2$', 4), ('$s_{19}$', 2)] | [('$s_{11}$', 4), ('$s_{26}$', 1), ('$s_8$', 1), ('$s_9$', 1), ('$s_7$', 2), ('$s_{17}$', 1)] | [('$s_{12}$', 1), ('$s_{28}$', 1), ('$s_{20}$', 4), ('$s_{23}$', 2), ('$s_{16}$', 4)] | [] | [('$s_{30}$', 3), ('$s_{15}$', 2), ('$s_1$', 2)] | [('$s_{14}$', 3), ('$s_{22}$', 1), ('$s_{25}$', 2), ('$s_{13}$', 1), ('$s_6$', 1), ('$s_{29}$', 1)] | [('$s_5$', 2), ('$s_{21}$', 2), ('$s_{18}$', 3)] |
| **5** | [('$s_{27}$', 3), ('$s_{24}$', 2), ('$s_{11}$', 5)] | [('$s_{16}$', 5), ('$s_{14}$', 4), ('$s_{18}$', 4), ('$s_{19}$', 2), ('$s_{30}$', 4)] | [('$s_{26}$', 1), ('$s_8$', 1), ('$s_9$', 1), ('$s_7$', 2), ('$s_{17}$', 1)] | [('$s_2$', 5), ('$s_{12}$', 1), ('$s_{28}$', 1), ('$s_{20}$', 4)] | [] | [('$s_{15}$', 2), ('$s_1$', 2)] | [('$s_{22}$', 1), ('$s_{25}$', 2), ('$s_{13}$', 1), ('$s_6$', 1), ('$s_{29}$', 1)] | [('$s_5$', 2), ('$s_{21}$', 2)] |
| **6** | [('$s_{27}$', 3), ('$s_{24}$', 2), ('$s_{11}$', 5)] | [('$s_{18}$', 4)] | [('$s_{26}$', 1), ('$s_8$', 1), ('$s_9$', 1), ('$s_7$', 2), ('$s_{17}$', 1)] | [('$s_{14}$', 5), ('$s_2$', 5), ('$s_{12}$', 1), ('$s_{30}$', 5), ('$s_{19}$', 3), ('$s_{28}$', 1)] | [] | [('$s_{20}$', 5), ('$s_{15}$', 2), ('$s_1$', 2)] | [('$s_{22}$', 1), ('$s_{25}$', 2), ('$s_{13}$', 1), ('$s_6$', 1), ('$s_{16}$', 6), ('$s_{29}$', 1)] | [('$s_5$', 2), ('$s_{21}$', 2)] |
| **7** | [('$s_{14}$', 6), ('$s_{27}$', 3), ('$s_{24}$', 2), ('$s_{11}$', 5)] | [('$s_{18}$', 4), ('$s_{29}$', 2)] | [('$s_{26}$', 1), ('$s_8$', 1), ('$s_{20}$', 6), ('$s_9$', 1), ('$s_7$', 2), ('$s_{17}$', 1)] | [('$s_2$', 5), ('$s_{12}$', 1), ('$s_{30}$', 5)] | [('$s_{28}$', 2)] | [('$s_{19}$', 4), ('$s_{15}$', 2), ('$s_1$', 2)] | [('$s_{22}$', 1), ('$s_{25}$', 2), ('$s_{13}$', 1), ('$s_6$', 1), ('$s_{16}$', 6)] | [('$s_5$', 2), ('$s_{21}$', 2)] |
| **8** | [('$s_{27}$', 3), ('$s_{24}$', 2), ('$s_{11}$', 5), ('$s_{28}$', 3)] | [('$s_{18}$', 4)] | [('$s_{19}$', 5), ('$s_{26}$', 1), ('$s_8$', 1), ('$s_{20}$', 6), ('$s_9$', 1), ('$s_7$', 2)] | [('$s_2$', 5), ('$s_{12}$', 1), ('$s_{30}$', 5)] | [] | [('$s_{14}$', 7), ('$s_{17}$', 2), ('$s_{15}$', 2), ('$s_1$', 2)] | [('$s_{22}$', 1), ('$s_{25}$', 2), ('$s_{13}$', 1), ('$s_6$', 1), ('$s_{16}$', 6)] | [('$s_5$', 2), ('$s_{21}$', 2)] |
| **9** | [('$s_{27}$', 3), ('$s_{24}$', 2), ('$s_{11}$', 5), ('$s_{28}$', 3)] | [('$s_{18}$', 4)] | [('$s_{19}$', 5), ('$s_{26}$', 1), ('$s_8$', 1), ('$s_{20}$', 6), ('$s_9$', 1)] | [('$s_2$', 5), ('$s_{12}$', 1), ('$s_{30}$', 5)] | [] | [('$s_{17}$', 2), ('$s_{15}$', 2), ('$s_7$', 3)] | [('$s_{22}$', 1), ('$s_{25}$', 2), ('$s_{13}$', 1), ('$s_6$', 1), ('$s_{16}$', 6)] | [('$s_5$', 2), ('$s_{21}$', 2), ('$s_1$', 3)] |
| **10** | [('$s_1$', 4), ('$s_{27}$', 3), ('$s_{24}$', 2), ('$s_{11}$', 5), ('$s_{28}$', 3)] | [('$s_7$', 4), ('$s_{18}$', 4)] | [('$s_{19}$', 5), ('$s_{26}$', 1), ('$s_8$', 1), ('$s_{20}$', 6), ('$s_9$', 1)] | [('$s_2$', 5), ('$s_{12}$', 1), ('$s_{30}$', 5)] | [] | [('$s_{17}$', 2), ('$s_{15}$', 2)] | [('$s_{22}$', 1), ('$s_{25}$', 2), ('$s_{13}$', 1), ('$s_6$', 1), ('$s_{16}$', 6)] | [('$s_5$', 2), ('$s_{21}$', 2)] |
| **11** | [('$s_1$', 4), ('$s_{27}$', 3), ('$s_{24}$', 2), ('$s_{11}$', 5), ('$s_{28}$', 3)] | [('$s_{18}$', 4)] | [('$s_{19}$', 5), ('$s_{26}$', 1), ('$s_8$', 1), ('$s_{20}$', 6), ('$s_9$', 1)] | [('$s_2$', 5), ('$s_{12}$', 1), ('$s_{30}$', 5)] | [('$s_7$', 5)] | [('$s_{17}$', 2), ('$s_{15}$', 2)] | [('$s_{22}$', 1), ('$s_{25}$', 2), ('$s_{13}$', 1), ('$s_6$', 1), ('$s_{16}$', 6)] | [('$s_5$', 2), ('$s_{21}$', 2)] |
| **12** | [('$s_1$', 4), ('$s_{27}$', 3), ('$s_{24}$', 2), ('$s_{11}$', 5), ('$s_{28}$', 3)] | [('$s_{18}$', 4)] | [('$s_{19}$', 5), ('$s_{26}$', 1), ('$s_8$', 1), ('$s_{20}$', 6), ('$s_9$', 1)] | [('$s_2$', 5), ('$s_{12}$', 1), ('$s_{30}$', 5), ('$s_7$', 6)] | [] | [('$s_{17}$', 2), ('$s_{15}$', 2)] | [('$s_{22}$', 1), ('$s_{25}$', 2), ('$s_{13}$', 1), ('$s_6$', 1), ('$s_{16}$', 6)] | [('$s_5$', 2), ('$s_{21}$', 2)] |
| **13** | [('$s_7$', 7), ('$s_1$', 4), ('$s_{27}$', 3), ('$s_{24}$', 2), ('$s_{11}$', 5), ('$s_{28}$', 3)] | [('$s_{18}$', 4)] | [('$s_{19}$', 5), ('$s_{26}$', 1), ('$s_8$', 1), ('$s_{20}$', 6), ('$s_9$', 1)] | [('$s_2$', 5), ('$s_{12}$', 1), ('$s_{30}$', 5)] | [] | [('$s_{17}$', 2), ('$s_{15}$', 2)] | [('$s_{22}$', 1), ('$s_{25}$', 2), ('$s_{13}$', 1), ('$s_6$', 1), ('$s_{16}$', 6)] | [('$s_5$', 2), ('$s_{21}$', 2)] |
| **14** | [('$s_1$', 4), ('$s_{27}$', 3), ('$s_{24}$', 2), ('$s_{11}$', 5), ('$s_{28}$', 3)] | [('$s_{18}$', 4)] | [('$s_{19}$', 5), ('$s_{26}$', 1), ('$s_8$', 1), ('$s_{20}$', 6), ('$s_9$', 1)] | [('$s_2$', 5), ('$s_{12}$', 1), ('$s_{30}$', 5)] | [] | [('$s_{17}$', 2), ('$s_{15}$', 2)] | [('$s_{22}$', 1), ('$s_{25}$', 2), ('$s_{13}$', 1), ('$s_6$', 1), ('$s_{16}$', 6)] | [('$s_5$', 2), ('$s_{21}$', 2), ('$s_7$', 8)] |
| **15** | [('$s_1$', 4), ('$s_{27}$', 3), ('$s_{24}$', 2), ('$s_{11}$', 5), ('$s_{28}$', 3)] | [('$s_{18}$', 4)] | [('$s_{19}$', 5), ('$s_{26}$', 1), ('$s_8$', 1), ('$s_{20}$', 6), ('$s_9$', 1)] | [('$s_2$', 5), ('$s_{12}$', 1), ('$s_{30}$', 5)] | [] | [('$s_{17}$', 2), ('$s_{15}$', 2)] | [('$s_{22}$', 1), ('$s_{25}$', 2), ('$s_{13}$', 1), ('$s_6$', 1), ('$s_{16}$', 6)] | [('$s_5$', 2), ('$s_{21}$', 2)] |

### 4.1.2 Gale-Shapley Okul Optimal Durağan Mekanizması Python Algoritması Örneği

Bu mekanizma için yazılan algoritma, özellik olarak Gale-Shapley Öğrenci Optimal Durağan Mekanizması için yazılan algoritma ile aynıdır. Bu örnekte de yirmi öğrenci ve sekiz okul mevcuttur.

*Örnek 15*: Piyasa bilgileri Ekler kısmında tablolar halinde verilmiştir. Bakınız EK-4.3 ve EK-4.4.

Eşleşme sonucu turlar halinde Tablo 2'de gösterilmiştir. Her öğrencinin sütununda bulunan okul her tur için o öğrenciye o turda hangi okulların teklif yaptığını gösterir. Eğer bir önceki tura geçici olarak kabul edilerek gelen bir okul var ise o okul da teklif yapan okullar ile birlikte gösterilmektedir. Okulların yanlarındaki değerler sütununda bulunan öğrencinin o okulu kaçıncı sırada tercih ettiğini belirtmektedir. Değer "0" ise o öğrenci o okula yerleşmektense boşta kalmayı tercih etmektedir. Mekanizma beşinci turda sona ermiştir. Beşinci turda bir okul hangi öğrencinin sütununda ise o öğrenci o okula nihai olarak atanmıştır. Eşleşme sonucunda oluşan atamalar şu şekildedir:

$$\mu_{okul\ optimal} = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 & s_9 & s_{10} & s_{11} & s_{12} & s_{13} & s_{14} & s_{15} & s_{16} & s_{17} & s_{18} & s_{19} & s_{20} \\ c_5 & c_7 & c_5 & - & c_7 & c_1 & c_4 & - & - & - & c_7 & c_3 & c_8 & - & - & c_2 & c_1 & c_6 & - & c_7 \end{pmatrix}$$

**Tablo 2**: Örnek 15 - Turlar ve Eşleşme Sonucu

| Okullar / Tur sayısı | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ | $s_6$ | $s_7$ | $s_8$ | $s_9$ | $s_{10}$ | $s_{11}$ | $s_{12}$ | $s_{13}$ | $s_{14}$ | $s_{15}$ | $s_{16}$ | $s_{17}$ | $s_{18}$ | $s_{19}$ | $s_{20}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | [('$c_6$', 0)] | [] | [('$c_5$', 7)] | [('$c_7$', 0)] | [('$c_5$', 0), ('$c_6$', 0)] | [('$c_7$', 0)] | [('$c_6$', 3)] | [('$c_3$', 0)] | [('$c_1$', 0), ('$c_7$', 0)] | [] | [('$c_3$', 0), ('$c_4$', 5)] | [('$c_1$', 0), ('$c_7$', 0), ('$c_3$', 1)] | [('$c_8$', 1)] | [] | [] | [('$c_6$', 0), ('$c_2$', 2)] | [] | [('$c_6$', 3)] | [] | [] |
| **2** | [] | [] | [('$c_5$', 7)] | [] | [('$c_7$', 3)] | [] | [('$c_6$', 3), ('$c_7$', 6)] | [] | [] | [] | [('$c_4$', 5)] | [('$c_5$', 0), ('$c_3$', 1)] | [('$c_8$', 1), ('$c_1$', 4)] | [] | [('$c_7$', 0)] | [('$c_3$', 0), ('$c_2$', 2)] | [] | [('$c_6$', 3)] | [] | [('$c_7$', 4), ('$c_1$', 6)] |
| **3** | [] | [('$c_7$', 2)] | [('$c_5$', 7)] | [] | [('$c_7$', 3)] | [('$c_1$', 1)] | [('$c_6$', 3)] | [] | [('$c_5$', 0)] | [] | [('$c_7$', 1), ('$c_4$', 5)] | [('$c_3$', 1)] | [('$c_8$', 1)] | [] | [] | [('$c_2$', 2)] | [] | [('$c_1$', 0), ('$c_6$', 3)] | [] | [('$c_7$', 4)] |
| **4** | [('$c_5$', 1)] | [('$c_7$', 2)] | [('$c_5$', 7)] | [] | [('$c_7$', 3)] | [('$c_1$', 1)] | [('$c_4$', 2), ('$c_6$', 3)] | [] | [] | [] | [('$c_7$', 1)] | [('$c_3$', 1)] | [('$c_8$', 1)] | [] | [] | [('$c_2$', 2)] | [('$c_1$', 2)] | [('$c_6$', 3)] | [] | [('$c_7$', 4)] |
| **5** | [('$c_5$', 1)] | [('$c_7$', 2)] | [('$c_5$', 7)] | [] | [('$c_7$', 3)] | [('$c_1$', 1)] | [('$c_4$', 2)] | [] | [] | [] | [('$c_7$', 1)] | [('$c_3$', 1)] | [('$c_8$', 1)] | [] | [] | [('$c_2$', 2)] | [('$c_1$', 2)] | [('$c_6$', 3)] | [] | [('$c_7$', 4)] |

### 4.1.3 En Yüksek Değiş Tokuş Döngüleri Mekanizması Python Algoritması Örneği

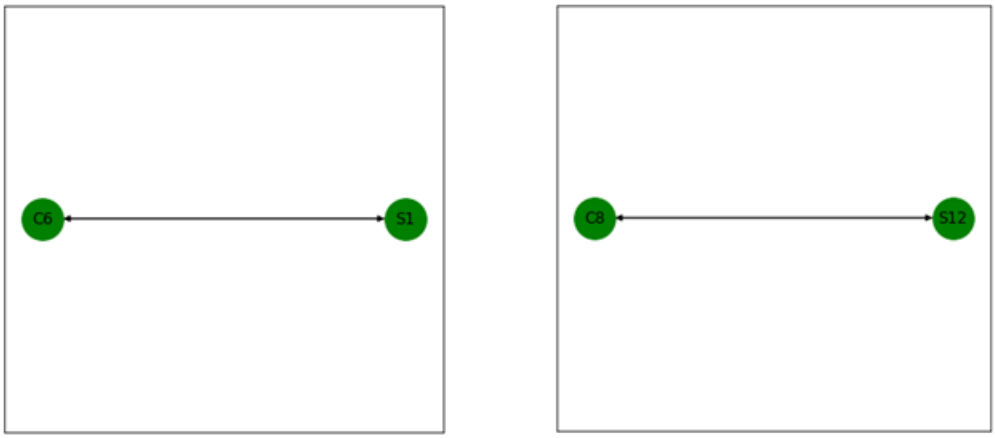
Algoritma özellik bakımından diğer iki mekanizma için yazılmış olan algoritmalar ile aynıdır. Tek fark bu algoritmada hiçbir öğrenci için $c_0$ ve hiçbir okul için $s_0$ tercih listelerine koyulmamıştır. Yani her öğrenci açıkta kalmaktansa bir okula yerleşmeyi tercih ederken her okul da kontenjanı boş kalmaktansa kendi tercih sıralamasına göre piyasadaki herhangi bir öğrenciyi kabul etmeyi tercih etmektedir.

Örnekte yirmi öğrenci ve sekiz okulun bulunduğu bir öğrenci-okul piyasası eşleşmesi yapılmıştır.

*Örnek 16:* Piyasaya ait bilgiler ekler kısmında verilmiştir. Bakınız EK-4.5 ve EK-4.6.

Mekanizmanın uygulanması ile birlikte birinci turda oluşan tüm döngüler aşağıdaki gibidir. (Her tur başında bütün okul ve öğrencilerin tercihlerini gösterdikleri şekiller ekler kısmında verilmiştir. Bakınız EK-4.7-14.)

**Şekil 35**: Örnek 16 – 1. Turda Oluşan Döngüler

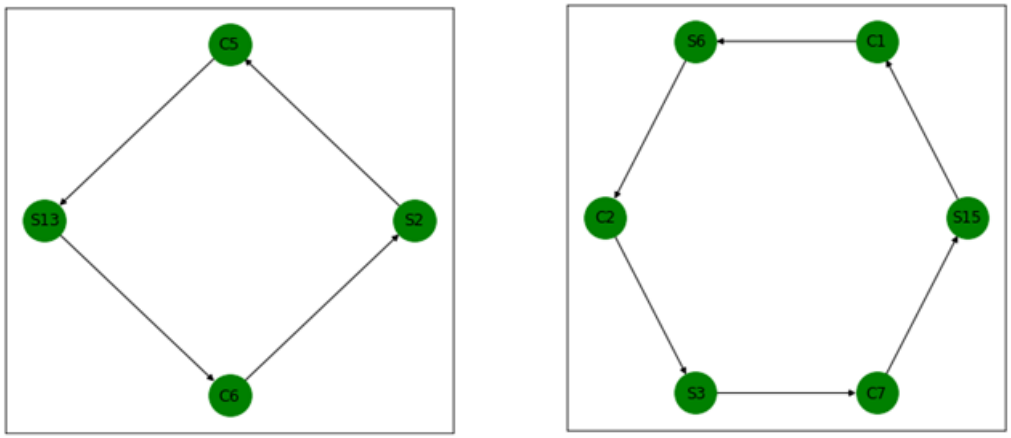



İlk turda yapılan atamalar:

$$\mu_{TTC_{(1.tur)}} = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 & s_9 & s_{10} & s_{11} & s_{12} & s_{13} & s_{14} & s_{15} & s_{16} & s_{17} & s_{18} & s_{19} & s_{20} \\ c_6 & - & - & - & - & - & - & - & - & - & - & c_8 & - & - & - & - & - & - & - & - \end{pmatrix}$$

İkinci turda oluşan tüm döngüler:

**Şekil 36**: Örnek 16 – 2. Turda Oluşan Döngüler

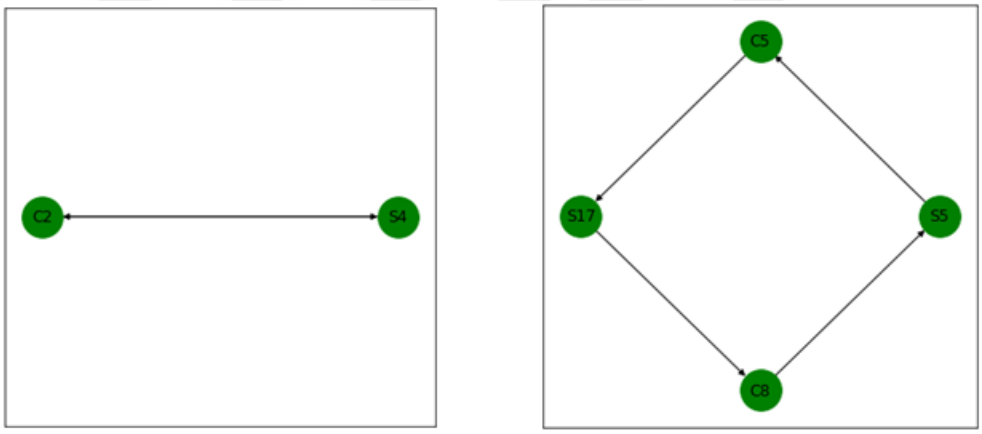



İkinci tur sonunda yapılmış tüm eşleşmeler:

$$\mu_{TTC_{(2.tur)}} = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 & s_9 & s_{10} & s_{11} & s_{12} & s_{13} & s_{14} & s_{15} & s_{16} & s_{17} & s_{18} & s_{19} & s_{20} \\ c_6 & c_5 & c_7 & - & - & c_2 & - & - & - & - & - & c_8 & c_6 & - & c_1 & - & - & - & - & - \end{pmatrix}$$

Üçüncü turda oluşan tüm döngüler:

**Şekil 37**: Örnek 16 – 3. Turda Oluşan Döngüler

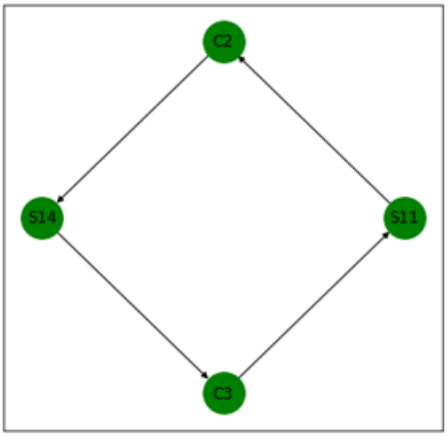



Üçüncü tur sonunda yapılmış tüm eşleşmeler:

$$\mu_{TTC_{(3.tur)}} = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 & s_9 & s_{10} & s_{11} & s_{12} & s_{13} & s_{14} & s_{15} & s_{16} & s_{17} & s_{18} & s_{19} & s_{20} \\ c_6 & c_5 & c_7 & c_2 & c_5 & c_2 & - & - & - & - & - & c_8 & c_6 & - & c_1 & - & c_8 & - & - & - \end{pmatrix}$$

Dördüncü turda tek bir döngü oluşmuştur. Dördüncü turda oluşan döngü aşağıdaki gibidir.

**Şekil 38**: Örnek 16 – 4. Turda Oluşan Döngü

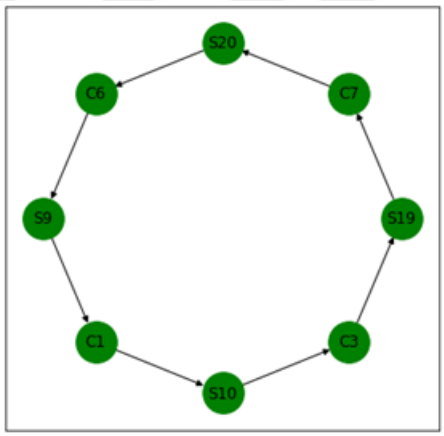



Dördüncü tur sonunda yapılmış tüm eşleşmeler:

$$\mu_{TTC_{(4.tur)}} = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 & s_9 & s_{10} & s_{11} & s_{12} & s_{13} & s_{14} & s_{15} & s_{16} & s_{17} & s_{18} & s_{19} & s_{20} \\ c_6 & c_5 & c_7 & c_2 & c_5 & c_2 & - & - & - & - & c_2 & c_8 & c_6 & c_3 & c_1 & - & c_8 & - & - & - \end{pmatrix}$$

Beşinci turda oluşan tek döngü:

**Şekil 39**: Örnek 16 – 5. Turda Oluşan Döngü

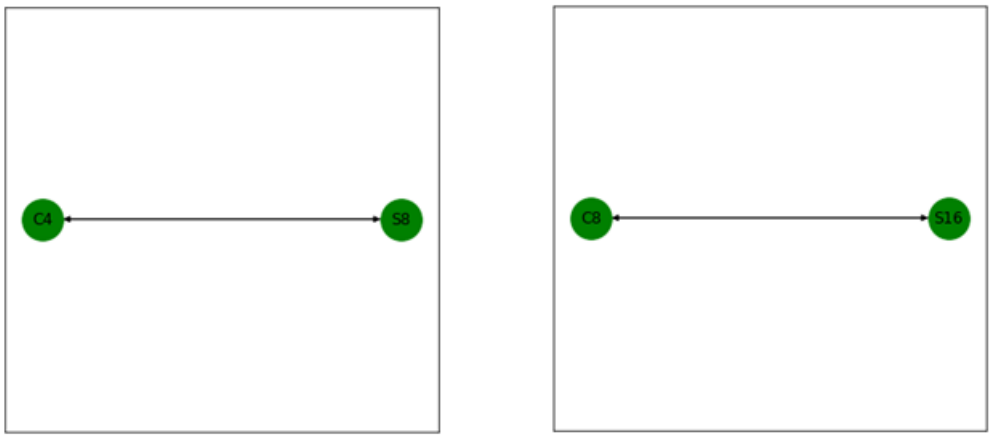



Beşinci tur sonunda yapılmış tüm eşleşmeler:

$$\mu_{TTC_{(5.tur)}} = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 & s_9 & s_{10} & s_{11} & s_{12} & s_{13} & s_{14} & s_{15} & s_{16} & s_{17} & s_{18} & s_{19} & s_{20} \\ c_6 & c_5 & c_7 & c_2 & c_5 & c_2 & - & - & c_1 & c_3 & c_2 & c_8 & c_6 & c_3 & c_1 & - & c_8 & - & c_7 & c_6 \end{pmatrix}$$

Altıncı turda oluşan tüm döngüler:

**Şekil 40**: Örnek 16 – 6. Turda Oluşan Döngüler

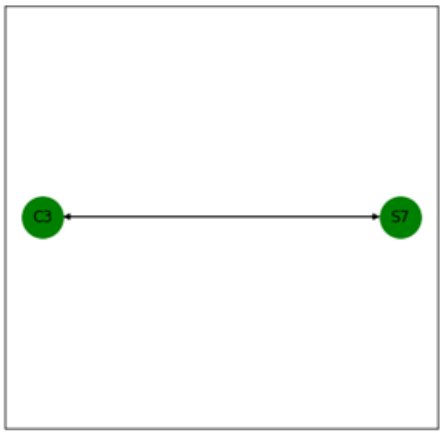



Altıncı tur sonunda yapılmış tüm eşleşmeler:

$$\mu_{TTC_{(6.tur)}} = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 & s_9 & s_{10} & s_{11} & s_{12} & s_{13} & s_{14} & s_{15} & s_{16} & s_{17} & s_{18} & s_{19} & s_{20} \\ c_6 & c_5 & c_7 & c_2 & c_5 & c_2 & - & c_4 & c_1 & c_3 & c_2 & c_8 & c_6 & c_3 & c_1 & c_8 & c_8 & - & c_7 & c_6 \end{pmatrix}$$

Yedinci turda oluşan tek döngü:

**Şekil 41**: Örnek 16 – 7. Turda Oluşan Döngü

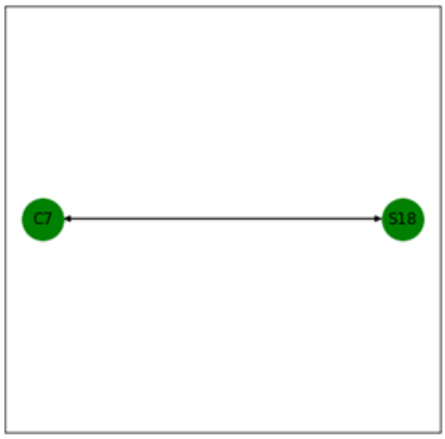



Yedinci tur sonunda yapılmış tüm eşleşmeler:

$$\mu_{TTC_{(7.tur)}} = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 & s_9 & s_{10} & s_{11} & s_{12} & s_{13} & s_{14} & s_{15} & s_{16} & s_{17} & s_{18} & s_{19} & s_{20} \\ c_6 & c_5 & c_7 & c_2 & c_5 & c_2 & c_3 & c_4 & c_1 & c_3 & c_2 & c_8 & c_6 & c_3 & c_1 & c_8 & c_8 & - & c_7 & c_6 \end{pmatrix}$$

Sekizince turda oluşan tek döngü:

**Şekil 42**: Örnek 16 – 8. Turda Oluşan Döngü




Sekizince turda daha önce atanmayan son öğrenci de atanmıştır ve böylece mekanizma 8. turda sona ermiştir. Mekanizma sonucunda oluşan eşleşmeler şu şekildedir.

$$\mu_{TTC} = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 & s_9 & s_{10} & s_{11} & s_{12} & s_{13} & s_{14} & s_{15} & s_{16} & s_{17} & s_{18} & s_{19} & s_{20} \\ c_6 & c_5 & c_7 & c_2 & c_5 & c_2 & c_3 & c_4 & c_1 & c_3 & c_2 & c_8 & c_6 & c_3 & c_1 & c_8 & c_8 & c_7 & c_7 & c_6 \end{pmatrix}$$

### 4.1.4 Çok Kategorili Seri Diktatörlük Mekanizması Python Algoritması Örneği

Algoritmada, okul tipleri manuel olarak belirlenmiştir ve istenilen miktarda okul tipi algoritmaya eklenebilmektedir. Öğrenci tercihleri, kotalar, öğrencilerin her tipten aldığı puanlar ve hangi okulun hangi tipte olacağı algoritma tarafından rastgele belirlenmektedir. Fakat piyasadan alınan bilgiler algoritmaya işlenerek eşleşmeler yapılabilmektedir. Algoritma her tipte en az bir okul olacak şekilde düzenlenmiş olup öğrencilerin her tipten aldıkları puanlara göre puan sıralaması algoritma tarafından otomatik olarak yapılmaktadır.

Örnekte otuz öğrenci sekiz okul mevcuttur.

*Örnek 17:* Piyasaya ait bilgiler ekler kısmında verilmiştir. Bakınız: EK-4.15 ve EK-4.16.

Eşleşme sonucu turlar halinde Tablo 3'de gösterilmiştir. Tabloda her öğrencinin yanında bulunan rakam o öğrencinin geçici veya nihai olarak atandığı okulu kaçıncı sırada tercih ettiğini göstermektedir. Mekanizma dördüncü turda sona ermiştir ve dördüncü turda her öğrenci hangi okulun sütununda bulunuyor ise nihai olarak o okula atanmıştır.

$$\mu_{MCSD} = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 & s_9 & s_{10} & s_{11} & s_{12} & s_{13} & s_{14} & s_{15} & s_{16} & s_{17} & s_{18} & s_{19} & s_{20} \\ c_6 & c_6 & c_6 & c_6 & c_5 & c_2 & c_1 & c_1 & c_3 & c_5 & c_3 & c_2 & c_2 & c_5 & c_7 & c_3 & c_7 & c_5 & c_5 & c_4 \end{pmatrix}$$

$$\begin{pmatrix} s_{21} & s_{22} & s_{23} & s_{24} & s_{25} & s_{26} & s_{27} & s_{28} & s_{29} & s_{30} \\ c_7 & c_2 & c_6 & c_5 & c_8 & c_4 & c_8 & c_2 & c_4 & c_6 \end{pmatrix}$$

**Tablo 3:** Örnek 17 - Turlar ve Eşleşme Sonucu

| Okullar / Tur Sayısı | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ | $c_7$ | $c_8$ |
|---|---|---|---|---|---|---|---|---|
| **1** | [('$s_5$', 4), ('$s_7$', 2)] | [('$s_{25}$', 3), ('$s_3$', 3), ('$s_{12}$', 3), ('$s_4$', 2), ('$s_{21}$', 2)] | [('$s_{20}$', 2), ('$s_{11}$', 4), ('$s_9$', 2)] | [('$s_{29}$', 1), ('$s_{30}$', 6), ('$s_{20}$', 1)] | [('$s_4$', 6), ('$s_{19}$', 1), ('$s_{29}$', 2), ('$s_{10}$', 1), ('$s_6$', 2), ('$s_{24}$', 2)] | [('$s_8$', 4), ('$s_1$', 2), ('$s_{29}$', 4), ('$s_4$', 1), ('$s_3$', 1), ('$s_{28}$', 4)] | [('$s_{17}$', 1), ('$s_3$', 2), ('$s_{15}$', 1)] | [('$s_{25}$', 2), ('$s_{27}$', 1)] |
| **2** | [('$s_5$', 4), ('$s_7$', 2)] | [('$s_{12}$', 3), ('$s_{21}$', 2), ('$s_{28}$', 2), ('$s_{26}$', 4), ('$s_{13}$', 1)] | [('$s_{11}$', 4), ('$s_9$', 2), ('$s_{16}$', 1)] | [('$s_{29}$', 1), ('$s_{30}$', 6), ('$s_{20}$', 1)] | [('$s_{19}$', 1), ('$s_{10}$', 1), ('$s_6$', 2), ('$s_{24}$', 2), ('$s_5$', 1), ('$s_{30}$', 4)] | [('$s_8$', 4), ('$s_1$', 2), ('$s_4$', 1), ('$s_3$', 1), ('$s_{28}$', 4), ('$s_{26}$', 3)] | [('$s_{17}$', 1), ('$s_{15}$', 1), ('$s_{21}$', 1)] | [('$s_{25}$', 2), ('$s_{27}$', 1)] |
| **3** | [('$s_7$', 2), ('$s_8$', 1)] | [('$s_{12}$', 3), ('$s_{28}$', 2), ('$s_{13}$', 1), ('$s_6$', 1), ('$s_{22}$', 2)] | [('$s_{11}$', 4), ('$s_9$', 2), ('$s_{16}$', 1)] | [('$s_{29}$', 1), ('$s_{20}$', 1), ('$s_{26}$', 1)] | [('$s_{19}$', 1), ('$s_{10}$', 1), ('$s_6$', 2), ('$s_{24}$', 2), ('$s_5$', 1), ('$s_{30}$', 4)] | [('$s_1$', 2), ('$s_4$', 1), ('$s_3$', 1), ('$s_{26}$', 3), ('$s_2$', 1), ('$s_{30}$', 1)] | [('$s_{17}$', 1), ('$s_{15}$', 1), ('$s_{21}$', 1)] | [('$s_{25}$', 2), ('$s_{27}$', 1)] |
| **4** | [('$s_7$', 2), ('$s_8$', 1)] | [('$s_{12}$', 3), ('$s_{28}$', 2), ('$s_{13}$', 1), ('$s_6$', 1), ('$s_{22}$', 2)] | [('$s_{11}$', 4), ('$s_9$', 2), ('$s_{16}$', 1)] | [('$s_{29}$', 1), ('$s_{20}$', 1), ('$s_{26}$', 1)] | [('$s_{19}$', 1), ('$s_{10}$', 1), ('$s_{24}$', 2), ('$s_5$', 1), ('$s_{14}$', 5), ('$s_{18}$', 2)] | [('$s_1$', 2), ('$s_4$', 1), ('$s_3$', 1), ('$s_2$', 1), ('$s_{30}$', 1), ('$s_{23}$', 3)] | [('$s_{17}$', 1), ('$s_{15}$', 1), ('$s_{21}$', 1)] | [('$s_{25}$', 2), ('$s_{27}$', 1)] |

### 4.1.5 Aynı Piyasa Örneği Üzerinden Üç Farklı Mekanizmanın Python Algoritması ile Yapılan Atama Sonuçları ve Fayda Endeksinin Hesaplanması

Gale-Shapley Öğrenci Optimal Durağan Mekanizması diğer durağan mekanizmaları Pareto domine ettiğini bilinmektedir. Bir önceki bölümde mekanizmaların aynı örnek üzerinden atamalarını yaptığımızda Gale-Shapley Okul Optimal Durağan Mekanizması ve Çok Kategorili Seri Diktatörlük Mekanizması'nın aynı sonuçları verdiğini ve Gale-Shapley Öğrenci Optimal Durağan Mekanizması'nın sonuçlarının diğer iki mekanizmadan öğrenci açısından daha iyi sonuç verdiğini gösterdik. Bu gösterimi öğrencilerin yerleştikleri okulları kaçıncı sırada tercih ettiğini karşılaştırarak yapmıştık.

Bu bölümde, geliştirilen *fayda endeksi* ile piyasadaki ataması yapılan öğrencilerin faydası üzerinden mekanizmaların atama sonuçlarının karşılaştırılması yapılacaktır.

#### 4.1.5.1 Fayda Endeksi

Fayda endeksi, atama sonucunda okula yerleşmiş olan öğrencilerin her birinin faydasının hesaplanması temeline dayanmaktadır ve oluşan farklı mekanizma ataması sonuçlarının değerlendirilmesinde kullanılmaktadır. Her öğrencinin tercih sıralaması faydaları birbirlerine eşittir. Yani piyasadaki her öğrenci için ilk tercihine atanması durumunda elde edeceği fayda aynıdır. Aynı şekilde ikinci tercihine atanan öğrenciler aynı fayda seviyesini elde eder. Bu durum her tercih için geçerlidir. Fayda endeksi sıfır ile bir arasında bir değer alır. Bir öğrenci ilk tercihine atanması durumunda "1" birim fayda elde eder. Eğer bir atamada her öğrenci ilk tercihine yerleşirse atamanın fayda endeksi bire eşit olur. Fayda endeksi bire ne kadar yakın ise o atama piyasadaki öğrenciler için daha fazla fayda sağlamaktadır.

Atanmayan öğrencilerin faydası "0" olarak kabul edilmiştir.

*Fayda endeksinin hesaplanması*:

*m=Okul sayısı*

*n=Öğrenci sayısı*

$s_k$ , $k=\{1, 2, 3... n\}$

$c_z$ , $z=\{1, 2, 3... m\}$

$R_{S_k}$=*Öğrencinin atandığı okul için tercih sırası*

$s_{util_k}$=*k öğrencisinin faydası*

$$s_{util_k} = \frac{1}{R_{s_k}}$$

$$\textit{Fayda Endeksi}(\varepsilon) = \frac{\sum_{k=1}^{n}\left(s_{util_k}\right)}{n}$$

Şimdi mekanizmaların aynı örnek (*Örnek 18*) üzerinden çözerek fayda endekslerini hesaplayalım:

*Örnek 18:* Örnekte okullar üç farklı kategoride (sosyal, matematik ve dil) olabilmektedir. Öğrenci tercihleri rastgele olarak ilk tercihe $c_0$ gelmeyecek ve her okul için $s_0$, tercih sırasının sonunda olacak şekilde algoritma tarafından otomatik olarak belirlenmektedir. Öğrencilerin algoritma tarafından her kategori için otomatik olarak atanmış puanları ile oluşan puan sıralamaları okul tercih sıralamalarını göstermektedir. Okul hangi kategori tipinde ise tercih sıralaması öğrencilerin o okul tipindeki puan sıralamasına eşittir. Örnekte yüz öğrenci ve otuz okul bulunmaktadır. Gösterimde parantez içinde öğrencinin yanında bulunan rakam o öğrencinin yerleştiği okulu kaçıncı sırada tercih ettiğini belirtmektedir.

*Gale-Shapley Öğrenci Optimal Durağan Mekanizması Atama Sonuçları* ($\mu_{\ddot{o}\breve{g}renci\ \text{optimal}}$):

$c_1$ : [('$s_{73}$', 2), ('$s_{30}$', 2), ('$s_{27}$', 2), ('$s_{49}$', 1), ('$s_{100}$', 1), ('$s_{53}$', 1)]

$c_2$ : [('$s_{9}$', 3), ('$s_{24}$', 1), ('$s_{60}$', 2), ('$s_{54}$', 1)]

$c_3$ : [('$s_{37}$', 1), ('$s_{1}$', 1), ('$s_{79}$', 1), ('$s_{56}$', 3)]

$c_4$ : [('$s_{67}$', 1), ('$s_{74}$', 1), ('$s_{18}$', 1), ('$s_{77}$', 6)]

$c_5$ : [('$s_{47}$', 1), ('$s_{43}$', 4)]

$c_6$ : [('$s_{36}$', 1), ('$s_{78}$', 1), ('$s_{12}$', 2), ('$s_{11}$', 2), ('$s_{76}$', 2)]

$c_7$ : [('$s_{92}$', 2), ('$s_{83}$', 1), ('$s_{80}$', 1)]

$c_8$ : [('$s_{72}$', 3)]

$c_9$ : [('$s_{99}$', 1), ('$s_{70}$', 4), ('$s_{22}$', 2), ('$s_{82}$', 4)]

$c_{10}$ : [('$s_6$', 2)]

$c_{11}$ : [('$s_{98}$', 1)]

$c_{12}$ : [('$s_{71}$', 1), ('$s_{38}$', 1)]

$c_{13}$ : [('$s_{34}$', 2), ('$s_{19}$', 1), ('$s_{46}$', 3), ('$s_{55}$', 1)]

$c_{14}$ : [('$s_{88}$', 2), ('$s_{84}$', 1)]

$c_{15}$ : [('$s_{50}$', 3), ('$s_{59}$', 1), ('$s_{41}$', 3), ('$s_{26}$', 2)]

$c_{16}$ : [('$s_{51}$', 1), ('$s_{87}$', 2), ('$s_{52}$', 1)]

$c_{17}$ : [('$s_{94}$', 1), ('$s_{44}$', 1)]

$c_{18}$ : [('$s_{32}$', 1), ('$s_{31}$', 2), ('$s_{57}$', 1), ('$s_{91}$', 1)]

$c_{19}$ : [('$s_{40}$', 2), ('$s_{68}$', 4), ('$s_4$', 1), ('$s_{33}$', 2), ('$s_{93}$', 3), ('$s_{85}$', 2)]

$c_{20}$ : [('$s_{45}$', 1), ('$s_{42}$', 2)]

$c_{21}$ : [('$s_{86}$', 1)]

$c_{22}$ : [('$s_{16}$', 1), ('$s_{69}$', 1), ('$s_2$', 2), ('$s_{63}$', 2)]

$c_{23}$ : [('$s_8$', 1), ('$s_{96}$', 1), ('$s_{65}$', 1), ('$s_{13}$', 2), ('$s_{81}$', 1)]

$c_{24}$ : [('$s_{23}$', 1), ('$s_{28}$', 1), ('$s_{97}$', 2)]

$c_{25}$ : [('$s_{64}$', 1), ('$s_{66}$', 1), ('$s_{14}$', 2), ('$s_{20}$', 1), ('$s_5$', 2)]

$c_{26}$ : [('$s_{17}$', 1), ('$s_{90}$', 4), ('$s_{39}$', 1), ('$s_{25}$', 1)]

$c_{27}$ : [('$s_{61}$', 1), ('$s_{95}$', 1), ('$s_7$', 1), ('$s_{58}$', 3), ('$s_{15}$', 1)]

$c_{28}$ : [('$s_{21}$', 1), ('$s_3$', 4), ('$s_{75}$', 1)]

$c_{29}$ : [('$s_{62}$', 2), ('$s_{10}$', 1)]

$c_{30}$ : [('$s_{89}$', 1), ('$s_{48}$', 5)]

*Gale-Shapley Okul Optimal Durağan Mekanizması Atama sonuçları* ($\mu_{okul\ optimal}$):

$c_1$ : [('$s_{73}$', 2), ('$s_{30}$', 2), ('$s_{27}$', 2), ('$s_{49}$', 1), ('$s_{100}$', 1), ('$s_{52}$', 2)]

$c_2$ : [('$s_9$', 3), ('$s_{24}$', 1), ('$s_{60}$', 2), ('$s_{54}$', 1)]

$c_3$ : [('$s_{37}$', 1), ('$s_1$', 1), ('$s_{79}$', 1), ('$s_{56}$', 3)]

$c_4$ : [('$s_{67}$', 1), ('$s_{74}$', 1), ('$s_{18}$', 1), ('$s_{77}$', 6)]

$c_5$ : [('$s_{47}$', 1), ('$s_{43}$', 4)]

$c_6$ : [('$s_{36}$', 1), ('$s_{78}$', 1), ('$s_{12}$', 2), ('$s_{11}$', 2), ('$s_{76}$', 2)]

$c_7$ : [('$s_{92}$', 2), ('$s_{83}$', 1), ('$s_{80}$', 1)]

$c_8$ : [('$s_{72}$', 3)]

$c_9$ : [('$s_{99}$', 1), ('$s_{70}$', 4), ('$s_{22}$', 2), ('$s_{82}$', 4)]

$c_{10}$ : [('$s_{6}$', 2)]

$c_{11}$ : [('$s_{98}$', 1)]

$c_{12}$ : [('$s_{71}$', 1), ('$s_{38}$', 1)]

$c_{13}$ : [('$s_{34}$', 2), ('$s_{19}$', 1), ('$s_{46}$', 3), ('$s_{55}$', 1)]

$c_{14}$ : [('$s_{88}$', 2), ('$s_{84}$', 1)]

$c_{15}$ : [('$s_{50}$', 3), ('$s_{59}$', 1), ('$s_{41}$', 3), ('$s_{26}$', 2)]

$c_{16}$ : [('$s_{51}$', 1), ('$s_{85}$', 4), ('$s_{87}$', 2)]

$c_{17}$ : [('$s_{94}$', 1), ('$s_{44}$', 1)]

$c_{18}$ : [('$s_{32}$', 1), ('$s_{31}$', 2), ('$s_{57}$', 1), ('$s_{91}$', 1)]

$c_{19}$ : [('$s_{40}$', 2), ('$s_{68}$', 4), ('$s_{4}$', 1), ('$s_{33}$', 2), ('$s_{93}$', 3), ('$s_{53}$', 2)]

$c_{20}$ : [('$s_{45}$', 1), ('$s_{42}$', 2)]

$c_{21}$ : [('$s_{86}$', 1)]

$c_{22}$ : [('$s_{16}$', 1), ('$s_{97}$', 3), ('$s_{69}$', 1), ('$s_{2}$', 2)]

$c_{23}$ : [('$s_{8}$', 1), ('$s_{96}$', 1), ('$s_{65}$', 1), ('$s_{13}$', 2), ('$s_{81}$', 1)]

$c_{24}$ : [('$s_{23}$', 1), ('$s_{63}$', 3), ('$s_{28}$', 1)]

$c_{25}$ : [('$s_{64}$', 1), ('$s_{66}$', 1), ('$s_{14}$', 2), ('$s_{20}$', 1), ('$s_{5}$', 2)]

$c_{26}$ : [('$s_{17}$', 1), ('$s_{90}$', 4), ('$s_{39}$', 1), ('$s_{25}$', 1)]

$c_{27}$ : [('$s_{61}$', 1), ('$s_{95}$', 1), ('$s_{7}$', 1), ('$s_{58}$', 3), ('$s_{15}$', 1)]

$c_{28}$ : [('$s_{21}$', 1), ('$s_{3}$', 4), ('$s_{75}$', 1)]

$c_{29}$ : [('$s_{62}$', 2), ('$s_{10}$', 1)]

$c_{30}$ : [('$s_{89}$', 1), ('$s_{48}$', 5)]

*Çok Kategorili Seri Diktatörlük Mekanizması Atama Sonuçları ($\mu_{MCSD}$)*:

$c_{1}$ : [('$s_{73}$', 2), ('$s_{30}$', 2), ('$s_{27}$', 2), ('$s_{49}$', 1), ('$s_{100}$', 1), ('$s_{52}$', 2)]

$c_{2}$ : [('$s_{9}$', 3), ('$s_{24}$', 1), ('$s_{60}$', 2), ('$s_{54}$', 1)]

$c_{3}$ : [('$s_{37}$', 1), ('$s_{1}$', 1), ('$s_{79}$', 1), ('$s_{56}$', 3)]

$c_{4}$ : [('$s_{67}$', 1), ('$s_{74}$', 1), ('$s_{18}$', 1), ('$s_{77}$', 6)]

$c_{5}$ : [('$s_{47}$', 1), ('$s_{43}$', 4)]

$c_{6}$ : [('$s_{36}$', 1), ('$s_{78}$', 1), ('$s_{12}$', 2), ('$s_{11}$', 2), ('$s_{76}$', 2)]

$c_{7}$ : [('$s_{92}$', 2), ('$s_{83}$', 1), ('$s_{80}$', 1)]

$c_{8}$ : [('$s_{72}$', 3)]

$c_{9}$ : [('$s_{99}$', 1), ('$s_{70}$', 4), ('$s_{22}$', 2), ('$s_{82}$', 4)]

$c_{10}$ : [('$s_{6}$', 2)]

$c_{11}$ : [('$s_{98}$', 1)]

$c_{12}$ : [('$s_{71}$', 1), ('$s_{38}$', 1)]

$c_{13}$ : [('$s_{34}$', 2), ('$s_{19}$', 1), ('$s_{46}$', 3), ('$s_{55}$', 1)]

$c_{14}$ : [('$s_{88}$', 2), ('$s_{84}$', 1)]

$c_{15}$ : [('$s_{50}$', 3), ('$s_{59}$', 1), ('$s_{41}$', 3), ('$s_{26}$', 2)]

$c_{16}$ : [('$s_{51}$', 1), ('$s_{85}$', 4), ('$s_{87}$', 2)]

$c_{17}$ : [('$s_{94}$', 1), ('$s_{44}$', 1)]

$c_{18}$ : [('$s_{32}$', 1), ('$s_{31}$', 2), ('$s_{57}$', 1), ('$s_{91}$', 1)]

$c_{19}$ : [('$s_{40}$', 2), ('$s_{68}$', 4), ('$s_{4}$', 1), ('$s_{33}$', 2), ('$s_{93}$', 3), ('$s_{53}$', 2)]

$c_{20}$ : [('$s_{45}$', 1), ('$s_{42}$', 2)]

$c_{21}$ : [('$s_{86}$', 1)]

$c_{22}$ : [('$s_{16}$', 1), ('$s_{97}$', 3), ('$s_{69}$', 1), ('$s_{2}$', 2)]

$c_{23}$ : [('$s_8$', 1), ('$s_{96}$', 1), ('$s_{65}$', 1), ('$s_{13}$', 2), ('$s_{81}$', 1)]

$c_{24}$ : [('$s_{23}$', 1), ('$s_{63}$', 3), ('$s_{28}$', 1)]

$c_{25}$ : [('$s_{64}$', 1), ('$s_{66}$', 1), ('$s_{14}$', 2), ('$s_{20}$', 1), ('$s_5$', 2)]

$c_{26}$ : [('$s_{17}$', 1), ('$s_{90}$', 4), ('$s_{39}$', 1), ('$s_{25}$', 1)]

$c_{27}$ : [('$s_{61}$', 1), ('$s_{95}$', 1), ('$s_7$', 1), ('$s_{58}$', 3), ('$s_{15}$', 1)]

$c_{28}$ : [('$s_{21}$', 1), ('$s_3$', 4), ('$s_{75}$', 1)]

$c_{29}$ : [('$s_{62}$', 2), ('$s_{10}$', 1)]

$c_{30}$ : [('$s_{89}$', 1), ('$s_{48}$', 5)]

**Tablo 4**: Fayda Endeks Değerleri

| Mekanizma | $\varepsilon$ |
|---|---|
| Gale-Shapley Öğrenci Optimal Durağan Mekanizması | $0.735\overline{3}$ |
| Gale-Shapley Okul Optimal Durağan Mekanizması | 0.7195 |
| Çok Kategorili Seri Diktatörlük Mekanizması | 0.7195 |

Mekanizmaların atama sonuçları karşılaştırıldığında $\mu_{okul\ \text{optimal}}$'in ve $\mu_{MCSD}$'nin aynı olduğu görülmektedir. Bu iki mekanizma atamaları ile Gale-Shapley Öğrenci Optimal Durağan Mekanizması ataması arasında ise farklılıklar mevcuttur. Farklı okullara yerleşen öğrenciler $s_{52}$, $s_{53}$, $s_{63}$, $s_{85}$, $s_{97}$'dir. Bu durum incelendiğinde $\mu_{öğrenci\ \text{optimal}}$'a göre ilk tercihine yerleşen $s_{52}$ ve $s_{53}$ diğer iki mekanizmada ikinci tercihine, $\mu_{öğrenci\ \text{optimal}}$'de ikinci tercihine yerleşen $s_{63}$, $s_{85}$ ve $s_{97}$ diğer iki mekanizmada sırasıyla üçüncü, dördüncü ve üçüncü tercihine yerleşmiştir. Dolayısıyla $\mu_{öğrenci\ \text{optimal}}$'de beş öğrenci diğer iki atamaya göre daha iyi duruma gelmiştir. Üç atama sonucunda da iki öğrenci açıkta kalmıştır.

Ayrıca üç mekanizmanın atama sonuçları için hesaplanan endeks değerlerine bakıldığında $\mu_{öğrenci\ \text{optimal}}$'in fayda endeks değeri $0.735\overline{3}$ iken, $\mu_{MCSD}$ ve $\mu_{okul\ \text{optimal}}$'in hesaplanan endeks değerlerinin 0.7195 olduğu ve Gale-Shapley Öğrenci Optimal Durağan Mekanizması'nın fayda endeks değerinin bire daha yakın olduğu görülmektedir. Dolayısıyla $\mu_{öğrenci\ \text{optimal}}$'in öğrenciler için diğer iki mekanizmadan daha fazla fayda sağladığı görülmektedir.

## SONUÇ

Eşleşme Teorisi'nin piyasa düzenlemelerinde başvurulan en önemli teorilerden biri olduğu yadsınamaz bir gerçektir. 1962 yılından bu yana gelişimine devam etmekte olan Eşleşme Teorisi içerisinde birçok teoriler ortaya atılmış ve ispatlanmıştır. Gerekli literatür taraması gerçekleştirildiğinde bu teorilerden bazılarının matematiksel olarak ispatlandığı fakat örnekler ile uygulamalı olarak gösterilmediği görülebilmektedir.

Bu çalışmada, daha önce geliştirilmiş olan eşleşme piyasası mekanizmalarının özellikleri bakımından birbirlerinden ne derece ve nasıl farklılaştıkları örnekler üzerinden gösterilerek, daha önce matematiksel olarak ispatlanmış olan teoriler ile uyumluluğu gözlemlenmiştir ve bu örneklerin bazılarının çözümünde çalışmamızda geliştirilmiş olan Python programlama dili temelli algoritma kullanılmıştır.

Piyasa örnekleri üzerinden hem üniversiteye girişte hem de okul seçiminde kullanılan Gale-Shapley Öğrenci Optimal Durağan Mekanizması, Gale-Shapley Okul Optimal Durağan Mekanizması, Çok Kategorili Seri Diktatörlük Mekanizması ve En Yüksek Değiş Tokuş Döngüleri Mekanizması karşılaştırılmıştır. Çalışmamızda *Örnek 12* ile Gale-Shapley Öğrenci Optimal Durağan Mekanizması, Gale-Shapley Okul Optimal Durağan Mekanizması, Çok Kategorili Seri Diktatörlük Mekanizması'nın karşılaştırılması yapılmıştır. Aynı piyasa örneği üç farklı mekanizma uygulanarak çözümlenmiş olup her mekanizmanın atama sonuçları değerlendirilmiştir. Değerlendirme sonucunda Gale-Shapley Okul Optimal Durağan Mekanizması ve Çok Kategorili Seri Diktatörlük Mekanizması'nın atama sonuçlarının aynı olduğu gözlemlenmiştir. Aynı örnek üzerinden Gale-Shapley Öğrenci Optimal Durağan Mekanizması eşleşmesinin diğer iki mekanizma eşleşmesini Pareto domine ettiği de gözlemlenebilmiştir. Ve bu iki sonucun daha önce ispatlanmış olan teoriler ile uyumlu olduğu görülmüştür.

*Örnek 13* ile Gale-Shapley Öğrenci Optimal Durağan Mekanizması ile En Yüksek Değiş Tokuş Döngüleri Mekanizması karşılaştırılmıştır. İki farklı mekanizmanın aynı piyasa örneğine uygulanması sonucunda oluşan farklı atama sonuçları değerlendirildiğinde En Yüksek Değiş Tokuş Döngüleri Mekanizması eşleşmesinin, Gale-Shapley Öğrenci Optimal Durağan Mekanizması eşleşmesini Pareto domine ettiği fakat eşleşmenin haklı kıskançlığa yol açtığı dolayısıyla durağan

olmadığı gözlemlenmiştir. Bu durumun da daha önce ispatlanmış olan teoriler ile uyumlu olduğu görülmüştür.

Mekanizma uygulamalarının, az sayıda oyuncu ve/veya nesnenin bulunduğu piyasalarda uygulanması kolay olmasına karşın oyuncu ve/veya nesne sayısının fazla olduğu piyasalarda mekanizma atamalarının yapılması kolay değildir. Bu zorluğun ortadan kaldırılması adına bu çalışmada Ekler kısmında kodları verilen, dört farklı mekanizmanın istenilen oyuncu ve/veya nesne sayısına sahip olan piyasalarda eşleşmelerini yapabilen Pyhton algoritması geliştirilmiştir. Algoritma istenilen piyasa (firma-işçi, stajyer doktor-hastane, öğrenci-okul yurt odaları vb.) için eşleşmeleri yapabilecek özelliğe sahiptir. Aynı zamanda çalışmamızda, farklı mekanizmaların atamalarının daha iyi karşılaştırılabilmesi adına "fayda endeksi" adıyla bir endeks geliştirilmiştir. Fayda endeksi her mekanizma atama sonuçları için ayrı olarak hesaplanabilmektedir. Endeks sıfır ile bir arasında bir değer almakta ve bire yaklaştıkça o piyasadaki öğrenciler için daha fazla fayda sağlamaktadır.

Çalışmamızda yüz öğrenci ve otuz okulun bulunduğu piyasa örneği için Gale-Shapley Öğrenci Optimal Durağan Mekanizması, Gale-Shapley Okul Optimal Durağan Mekanizması ve Çok Kategorili Seri Diktatörlük Mekanizması atamaları yapılmış olup atama sonuçlarına göre endeks değerleri hesaplanmıştır. Oluşan eşleşmeler teori ile tutarlı olup hesaplanan endeks değerlerine göre Gale-Shapley Okul Optimal Durağan Mekanizması eşleşmesinin beklenildiği gibi bire daha yakın bir değer aldığı görülmüştür. Dolayısıyla diğer iki mekanizma eşleşmesine göre Gale-Shapley Öğrenci Optimal Durağan Mekanizması eşleşmesinin öğrenciler için daha fazla fayda sağladığı gözlemlenmiştir.

Sonuç olarak, geliştirilmiş olan Python algoritması ile dört farklı mekanizmanın oyuncu ve/veya nesne sayısının fazla olduğu piyasalarda uygulamasının kolayca yapılabilmesine olanak sağlanmıştır. Ayrıca daha önce örnek piyasalar üzerinden gösterimi yapılmamış fakat matematiksel olarak ispatlanmış olan birçok teori çalışmada uygulaması yapılmış örnekler ile gösterilmiştir. Teorilerle uyumlu olan örnek sonuçlarında da görüldüğü gibi Gale-Shapley Öğrenci Optimal Durağan Mekanizması durağan olan diğer tüm mekanizma eşleşme sonuçlarını Pareto domine etmektedir. Dolayısıyla öğrenci-okul eşleşme piyasasında eşleşmelerin yapılmasında kullanılabilecek en uygun mekanizmadır. Örnekler

sonucunda görülebilen ve daha önce ispatlanmış olan diğer teoriler ise şu şekildedir: Gale-Shapley Öğrenci Optimal Durağan Mekanizması ve Çok Kategorili Seri Diktatörlük Mekanizması eşdeğer mekanizmalardır; En Yüksek Değiş Tokuş Döngüleri Mekanizması, Gale-Shapley Öğrenci Optimal Durağan Mekanizması'nı bazı eşleşmelerde Pareto domine etmektedir fakat durağan olmayan bir mekanizmadır. Bu iki mekanizma arasında hangisinin kullanılacağına ise karar alıcılar karar vermelidir. Eğer karar alıcılar haklı kıskançlığın ortadan kaldırılmasına, Pareto etkinlikten daha fazla önem veriyorlar ise Gale-Shapley Öğrenci Optimal Durağan Mekanizması, eğer Pareto etkinliğe, haklı kıskançlığın ortadan kaldırılmasına daha fazla önem veriyorlar ise En Yüksek Değiş Tokuş Döngüleri Mekanizması daha uygun bir mekanizmadır (Abdulkadiroğlu ve Sönmez, 2014: 306).

Ayrıca bu çalışmada geliştirilmiş olan "fayda endeksi" ile atama sonuçlarının değerlendirilmesine yeni bir bakış açısı getirilmiştir.

**KAYNAKÇA**

## Kitaplar

Hirsch, D. (1994). *School: A Matter of Choice.* Organization for Economic Co-operation and Development (OECD).

## Makaleler, Bildiriler, Diğer Basılı Yayınlar

Abdulkadiroğlu, A., & Sönmez, T. (2003). School Choice: A Mechanism Design Approach. *American Economic Review,* 93(3): 729-747.

Abdulkadiroğlu, A., & Sönmez, T. (2014). Okul Seçimi: Bir Mekanizma Tasarımı Yaklaşımı. *Journal of Economics and Political Economy,* 1(2): 302-326.

Abdulkadiroğlu, A., Parag, P. A., & Roth, A. E. (2009). Strategy-Proofness versus Efficiency in Matching with Indifferences: Redesigning the NYC High School Match. *American Economic Review,* 99(5): 1954–1978.

Abdulkadiroğlu, A., Pathak, P. A., & Roth, A. E. (2005). The New York City High School Match. *The American Economic Review,* 95(2): 364-367.

Alcalde , J., & Barberà, S. (1993). Top Dominance and the Possibility of Strategy-Proof Stable Solutions to Matching Problems. *Economic Theory,* 4(3): 417-435.

Balinski, M., & Sönmez, T. (1999). A Tale of Two Mechanisms: Student Placement. *Journal of Economic Theory,* 84(1): 73-94.

Calsamiglia, C., Haeringer, G., & Klijn, F. (2010). Constrained School Choice: An Experimental Study. *American Economic Review,* 100(4): 1860-1874.

Doğan, K. M., & Yuret , T. (2010). Üniversitelere Öğrenci Yerleştirme Sisteminde Tercih Bildirimindeki Kısıtlamanın Etkileri. 65(2): *Ankara Üniversitesi SBF Dergisi,* 59-88.

Doğan, M. K. (2009). Matching with Restricted Preferences. *Sosyoekonomi,* 10(10): 109-120.

Doğan, M. K. (2014). Eşleşme Teorisi ve Piyasa Tasarımı. *Ankara Üniversitesi SBF Dergisi*, 69(2): 379-405.

Erdil, A., & Ergin, H. (2008). What's the Matter with Tie-Breaking? Improving Efficiency in School Choice. *American Economic Review*, 98(3): 669-689.

Eşme, İ. (2014). Türkiye'de Yükseköğretime Geçiş Sistemi. *Yükseköğretim Dergisi*, 4(3): 148–157.

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

Kesten, O. (2010). School Choice with Consent. *The Quarterly Journal of Economics*, 125(3): 1297-1348.

Kılıç, R. (1999), Türkiye'de Yükseköğretimin Kapsamı ve Tarihsel Gelişimi. *Dumlupınar Üniversitesi Sosyal Bilimler Dergisi*, (3): 289:310

Roth, A. E. (1984). The Evolution of the Labor Market for Medical Interns and Residents: A Case Study in Game Theory. *Journal of Political Economy*, 92(6): 991-1016.

Roth, A. E., Sönmez, T., & Ünver, M. U. (2004). Kidney Exchange. *The Quarterly Journal of Economics*, 119(2): 457-488.

Shapley, L., & Scarf, H. (1974). On Cores and Indivisibility. *Journal of Mathematical Economics*, 1(1): 23-37.

Sönmez, T., & Ünver, M. U. (2010). Matching, Allocation, and Exchange of Discrete Resources. J. Benhabib, A. Bisin, & M. Jackson içinde. *Handbook of Social Economics.* North Holland, (s. 781-852).

Willie, C. V., & Alves, M. J. (1996). *Controlled Choice: A New Approach to School Desegregated Education and School Improvement.* Washington: New England Desegregation Assistance Center for Equity in Education, Providence, RI.; Brown Univ., Providence, RI. Education Alliance for Equity in the Nation's Schools.

**Elektronik Kaynaklar**

Economic Sciences Prize Committee of the Royal Swedish Academy of Sciences. (2012, Ekim 15). Stable Allocations and The Practice of Market Design. Stokholm, İsviçre.

# EKLER
## EK-4.1: Örnek 14 – Öğrenci Tercihleri

| Öğrenciler / Tercih sırası | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ | $s_6$ | $s_7$ | $s_8$ | $s_9$ | $s_{10}$ | $s_{11}$ | $s_{12}$ | $s_{13}$ | $s_{14}$ | $s_{15}$ | $s_{16}$ | $s_{17}$ | $s_{18}$ | $s_{19}$ | $s_{20}$ | $s_{21}$ | $s_{22}$ | $s_{23}$ | $s_{24}$ | $s_{25}$ | $s_{26}$ | $s_{27}$ | $s_{28}$ | $s_{29}$ | $s_{30}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | $c_5$ | $c_8$ | $c_7$ | $c_5$ | $c_2$ | $c_7$ | $c_7$ | $c_3$ | $c_3$ | $c_4$ | $c_4$ | $c_4$ | $c_7$ | $c_8$ | $c_3$ | $c_5$ | $c_3$ | $c_6$ | $c_5$ | $c_8$ | $c_5$ | $c_7$ | $c_6$ | $c_3$ | $c_2$ | $c_3$ | $c_3$ | $c_4$ | $c_7$ | $c_8$ |
| **2** | $c_6$ | $c_1$ | $c_2$ | $c_3$ | $c_8$ | $c_2$ | $c_3$ | $c_2$ | $c_7$ | $c_2$ | $c_6$ | $c_5$ | $c_8$ | $c_3$ | $c_6$ | $c_1$ | $c_6$ | $c_4$ | $c_2$ | $c_5$ | $c_8$ | $c_2$ | $c_4$ | $c_1$ | $c_7$ | $c_7$ | $c_7$ | $c_5$ | $c_2$ | $c_1$ |
| **3** | $c_8$ | $c_6$ | $c_0$ | $c_7$ | $c_1$ | $c_4$ | $c_6$ | $c_0$ | $c_1$ | $c_7$ | $c_8$ | $c_7$ | $c_5$ | $c_7$ | $c_7$ | $c_6$ | $c_2$ | $c_8$ | $c_4$ | $c_2$ | $c_1$ | $c_1$ | $c_0$ | $c_5$ | $c_1$ | $c_4$ | $c_1$ | $c_1$ | $c_0$ | $c_6$ |
| **4** | $c_1$ | $c_2$ | $c_4$ | $c_0$ | $c_4$ | $c_3$ | $c_2$ | $c_6$ | $c_0$ | $c_0$ | $c_3$ | $c_1$ | $c_0$ | $c_2$ | $c_8$ | $c_4$ | $c_0$ | $c_2$ | $c_6$ | $c_4$ | $c_7$ | $c_6$ | $c_7$ | $c_2$ | $c_6$ | $c_5$ | $c_4$ | $c_0$ | $c_3$ | $c_2$ |
| **5** | $c_4$ | $c_4$ | $c_6$ | $c_6$ | $c_6$ | $c_1$ | $c_5$ | $c_7$ | $c_2$ | $c_8$ | $c_1$ | $c_3$ | $c_1$ | $c_4$ | $c_4$ | $c_2$ | $c_8$ | $c_1$ | $c_3$ | $c_6$ | $c_3$ | $c_0$ | $c_3$ | $c_7$ | $c_4$ | $c_1$ | $c_0$ | $c_2$ | $c_4$ | $c_4$ |
| **6** | $c_0$ | $c_5$ | $c_3$ | $c_2$ | $c_3$ | $c_0$ | $c_4$ | $c_8$ | $c_8$ | $c_5$ | $c_7$ | $c_0$ | $c_3$ | $c_1$ | $c_2$ | $c_7$ | $c_5$ | $c_7$ | $c_7$ | $c_3$ | $c_2$ | $c_4$ | $c_2$ | $c_0$ | $c_3$ | $c_6$ | $c_6$ | $c_8$ | $c_5$ | $c_0$ |
| **7** | $c_2$ | $c_3$ | $c_1$ | $c_1$ | $c_5$ | $c_6$ | $c_1$ | $c_1$ | $c_6$ | $c_1$ | $c_5$ | $c_2$ | $c_6$ | $c_6$ | $c_5$ | $c_0$ | $c_7$ | $c_3$ | $c_0$ | $c_0$ | $c_0$ | $c_5$ | $c_5$ | $c_8$ | $c_5$ | $c_2$ | $c_8$ | $c_7$ | $c_8$ | $c_7$ |
| **8** | $c_3$ | $c_0$ | $c_8$ | $c_8$ | $c_7$ | $c_8$ | $c_8$ | $c_4$ | $c_5$ | $c_3$ | $c_2$ | $c_8$ | $c_4$ | $c_0$ | $c_0$ | $c_8$ | $c_4$ | $c_0$ | $c_8$ | $c_1$ | $c_4$ | $c_8$ | $c_1$ | $c_6$ | $c_0$ | $c_0$ | $c_2$ | $c_3$ | $c_6$ | $c_5$ |
| **9** | $c_7$ | $c_7$ | $c_5$ | $c_4$ | $c_0$ | $c_5$ | $c_0$ | $c_5$ | $c_4$ | $c_6$ | $c_0$ | $c_6$ | $c_2$ | $c_5$ | $c_1$ | $c_3$ | $c_1$ | $c_5$ | $c_1$ | $c_7$ | $c_6$ | $c_3$ | $c_8$ | $c_4$ | $c_8$ | $c_8$ | $c_5$ | $c_6$ | $c_1$ | $c_3$ |

## EK-4.2: Örnek 14 – Okul Tercihleri ve Kota Bilgisi

| | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ | $c_7$ | $c_8$ |
|---|---|---|---|---|---|---|---|---|
| **Kota** | 5 | 1 | 5 | 3 | 7 | 2 | 5 | 2 |

| Okullar / Tercih sırası | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ | $c_7$ | $c_8$ |
|---|---|---|---|---|---|---|---|---|
| **1** | $s_5$ | $s_{11}$ | $s_{12}$ | $s_2$ | $s_{27}$ | $s_{17}$ | $s_{22}$ | $s_9$ |
| **2** | $s_9$ | $s_{13}$ | $s_{21}$ | $s_{26}$ | $s_5$ | $s_{26}$ | $s_{26}$ | $s_5$ |
| **3** | $s_{18}$ | $s_{17}$ | $s_{19}$ | $s_{12}$ | $s_0$ | $s_{29}$ | $s_{18}$ | $s_{19}$ |
| **4** | $s_1$ | $s_{18}$ | $s_2$ | $s_{30}$ | $s_9$ | $s_{15}$ | $s_{19}$ | $s_{10}$ |
| **5** | $s_{12}$ | $s_{15}$ | $s_{22}$ | $s_{22}$ | $s_{20}$ | $s_3$ | $s_{25}$ | $s_{21}$ |
| **6** | $s_{27}$ | $s_{19}$ | $s_{10}$ | $s_8$ | $s_3$ | $s_4$ | $s_1$ | $s_4$ |
| **7** | $s_{23}$ | $s_{30}$ | $s_{23}$ | $s_{29}$ | $s_7$ | $s_{24}$ | $s_{30}$ | $s_{12}$ |
| **8** | $s_{22}$ | $s_{10}$ | $s_{18}$ | $s_{19}$ | $s_{24}$ | $s_{10}$ | $s_{13}$ | $s_{15}$ |
| **9** | $s_{17}$ | $s_4$ | $s_{26}$ | $s_{17}$ | $s_{21}$ | $s_{27}$ | $s_{28}$ | $s_1$ |
| **10** | $s_{15}$ | $s_{29}$ | $s_8$ | $s_5$ | $s_{11}$ | $s_{22}$ | $s_6$ | $s_{11}$ |
| **11** | $s_{24}$ | $s_0$ | $s_{20}$ | $s_{28}$ | $s_{14}$ | $s_8$ | $s_9$ | $s_{25}$ |
| **12** | $s_{11}$ | $s_1$ | $s_9$ | $s_{24}$ | $s_{17}$ | $s_{28}$ | $s_{16}$ | $s_{26}$ |
| **13** | $s_{19}$ | $s_9$ | $s_7$ | $s_{15}$ | $s_{19}$ | $s_1$ | $s_{29}$ | $s_3$ |
| **14** | $s_{28}$ | $s_{14}$ | $s_{17}$ | $s_{20}$ | $s_{25}$ | $s_7$ | $s_8$ | $s_{16}$ |
| **15** | $s_{29}$ | $s_{27}$ | $s_0$ | $s_4$ | $s_{12}$ | $s_{16}$ | $s_{23}$ | $s_{30}$ |
| **16** | $s_0$ | $s_{16}$ | $s_{29}$ | $s_{21}$ | $s_{23}$ | $s_{25}$ | $s_7$ | $s_7$ |
| **17** | $s_{14}$ | $s_8$ | $s_{24}$ | $s_6$ | $s_{13}$ | $s_{13}$ | $s_{27}$ | $s_{24}$ |
| **18** | $s_2$ | $s_6$ | $s_{14}$ | $s_3$ | $s_8$ | $s_{12}$ | $s_{17}$ | $s_{17}$ |
| **19** | $s_{26}$ | $s_{28}$ | $s_{27}$ | $s_{23}$ | $s_4$ | $s_9$ | $s_{12}$ | $s_{29}$ |
| **20** | $s_{16}$ | $s_{12}$ | $s_{16}$ | $s_{16}$ | $s_{26}$ | $s_5$ | $s_2$ | $s_{14}$ |
| **21** | $s_{21}$ | $s_{22}$ | $s_1$ | $s_7$ | $s_2$ | $s_{23}$ | $s_5$ | $s_{20}$ |
| **22** | $s_4$ | $s_{21}$ | $s_6$ | $s_9$ | $s_{30}$ | $s_0$ | $s_{24}$ | $s_{22}$ |
| **23** | $s_3$ | $s_{23}$ | $s_4$ | $s_{27}$ | $s_1$ | $s_{20}$ | $s_{15}$ | $s_{23}$ |
| **24** | $s_{30}$ | $s_5$ | $s_{13}$ | $s_{18}$ | $s_{18}$ | $s_{11}$ | $s_{21}$ | $s_{13}$ |
| **25** | $s_8$ | $s_3$ | $s_{28}$ | $s_1$ | $s_{22}$ | $s_6$ | $s_{11}$ | $s_{27}$ |
| **26** | $s_7$ | $s_7$ | $s_3$ | $s_0$ | $s_{29}$ | $s_{30}$ | $s_4$ | $s_{18}$ |
| **27** | $s_6$ | $s_{24}$ | $s_{15}$ | $s_{25}$ | $s_{15}$ | $s_{18}$ | $s_3$ | $s_8$ |
| **28** | $s_{25}$ | $s_{25}$ | $s_{11}$ | $s_{13}$ | $s_{10}$ | $s_{21}$ | $s_0$ | $s_{28}$ |
| **29** | $s_{20}$ | $s_{20}$ | $s_{25}$ | $s_{14}$ | $s_{28}$ | $s_{14}$ | $s_{14}$ | $s_0$ |
| **30** | $s_{10}$ | $s_{26}$ | $s_5$ | $s_{11}$ | $s_{16}$ | $s_{19}$ | $s_{20}$ | $s_6$ |
| **31** | $s_{13}$ | $s_2$ | $s_{30}$ | $s_{10}$ | $s_6$ | $s_2$ | $s_{10}$ | $s_2$ |

## EK-4.3: Örnek 15 – Öğrenci Tercihleri

| Öğrenciler / Tercih sırası | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ | $s_6$ | $s_7$ | $s_8$ | $s_9$ | $s_{10}$ | $s_{11}$ | $s_{12}$ | $s_{13}$ | $s_{14}$ | $s_{15}$ | $s_{16}$ | $s_{17}$ | $s_{18}$ | $s_{19}$ | $s_{20}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | $c_5$ | $c_8$ | $c_4$ | $c_1$ | $c_8$ | $c_1$ | $c_1$ | $c_4$ | $c_3$ | $c_3$ | $c_7$ | $c_3$ | $c_8$ | $c_6$ | $c_2$ | $c_5$ | $c_6$ | $c_4$ | $c_3$ | $c_2$ |
| **2** | $c_8$ | $c_7$ | $c_2$ | $c_3$ | $c_4$ | $c_4$ | $c_4$ | $c_2$ | $c_8$ | $c_2$ | $c_2$ | $c_8$ | $c_5$ | $c_7$ | $c_5$ | $c_2$ | $c_1$ | $c_8$ | $c_2$ | $c_4$ |
| **3** | $c_0$ | $c_3$ | $c_7$ | $c_0$ | $c_7$ | $c_6$ | $c_6$ | $c_0$ | $c_4$ | $c_4$ | $c_8$ | $c_0$ | $c_4$ | $c_3$ | $c_4$ | $c_8$ | $c_4$ | $c_6$ | $c_6$ | $c_8$ |
| **4** | $c_1$ | $c_6$ | $c_8$ | $c_8$ | $c_0$ | $c_0$ | $c_8$ | $c_3$ | $c_2$ | $c_0$ | $c_1$ | $c_6$ | $c_1$ | $c_0$ | $c_0$ | $c_0$ | $c_2$ | $c_2$ | $c_0$ | $c_7$ |
| **5** | $c_7$ | $c_5$ | $c_6$ | $c_6$ | $c_2$ | $c_5$ | $c_2$ | $c_5$ | $c_0$ | $c_8$ | $c_4$ | $c_7$ | $c_2$ | $c_4$ | $c_8$ | $c_6$ | $c_5$ | $c_3$ | $c_1$ | $c_6$ |
| **6** | $c_6$ | $c_1$ | $c_3$ | $c_4$ | $c_1$ | $c_2$ | $c_7$ | $c_1$ | $c_1$ | $c_5$ | $c_0$ | $c_5$ | $c_0$ | $c_2$ | $c_3$ | $c_4$ | $c_7$ | $c_0$ | $c_5$ | $c_1$ |
| **7** | $c_2$ | $c_4$ | $c_5$ | $c_5$ | $c_3$ | $c_7$ | $c_5$ | $c_6$ | $c_6$ | $c_6$ | $c_3$ | $c_4$ | $c_7$ | $c_5$ | $c_1$ | $c_7$ | $c_0$ | $c_5$ | $c_8$ | $c_3$ |
| **8** | $c_4$ | $c_0$ | $c_0$ | $c_2$ | $c_6$ | $c_8$ | $c_0$ | $c_7$ | $c_7$ | $c_7$ | $c_6$ | $c_2$ | $c_6$ | $c_8$ | $c_6$ | $c_3$ | $c_8$ | $c_1$ | $c_7$ | $c_0$ |
| **9** | $c_3$ | $c_2$ | $c_1$ | $c_7$ | $c_5$ | $c_3$ | $c_3$ | $c_8$ | $c_5$ | $c_1$ | $c_5$ | $c_1$ | $c_3$ | $c_1$ | $c_7$ | $c_1$ | $c_3$ | $c_7$ | $c_4$ | $c_5$ |

## EK-4.4: Örnek 15 - Okul Tercihleri ve Kota Bilgisi

| | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ | $c_7$ | $c_8$ |
|---|---|---|---|---|---|---|---|---|
| **Kota** | 2 | 1 | 3 | 1 | 2 | 6 | 4 | 1 |

| Okullar / Tercih sırası | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ | $c_7$ | $c_8$ |
|---|---|---|---|---|---|---|---|---|
| **1** | $s_9$ | $s_{16}$ | $s_8$ | $s_{11}$ | $s_5$ | $s_1$ | $s_4$ | $s_{13}$ |
| **2** | $s_{12}$ | $s_{18}$ | $s_{11}$ | $s_7$ | $s_3$ | $s_5$ | $s_6$ | $s_{10}$ |
| **3** | $s_{20}$ | $s_1$ | $s_{12}$ | $s_1$ | $s_{12}$ | $s_7$ | $s_{12}$ | $s_3$ |
| **4** | $s_{13}$ | $s_{11}$ | $s_{16}$ | $s_9$ | $s_9$ | $s_{16}$ | $s_9$ | $s_{11}$ |
| **5** | $s_6$ | $s_{20}$ | $s_0$ | $s_5$ | $s_1$ | $s_{18}$ | $s_5$ | $s_2$ |
| **6** | $s_{18}$ | $s_{12}$ | $s_9$ | $s_6$ | $s_{11}$ | $s_0$ | $s_{15}$ | $s_{12}$ |
| **7** | $s_{17}$ | $s_4$ | $s_{14}$ | $s_{10}$ | $s_0$ | $s_{11}$ | $s_{20}$ | $s_{18}$ |
| **8** | $s_2$ | $s_{13}$ | $s_3$ | $s_4$ | $s_{15}$ | $s_9$ | $s_7$ | $s_8$ |
| **9** | $s_{19}$ | $s_0$ | $s_4$ | $s_2$ | $s_8$ | $s_{19}$ | $s_{11}$ | $s_0$ |
| **10** | $s_{11}$ | $s_8$ | $s_{13}$ | $s_{20}$ | $s_6$ | $s_{17}$ | $s_2$ | $s_6$ |
| **11** | $s_{15}$ | $s_7$ | $s_6$ | $s_0$ | $s_{17}$ | $s_{14}$ | $s_0$ | $s_{20}$ |
| **12** | $s_7$ | $s_2$ | $s_{19}$ | $s_{16}$ | $s_{18}$ | $s_{12}$ | $s_{14}$ | $s_{19}$ |
| **13** | $s_{16}$ | $s_5$ | $s_{10}$ | $s_3$ | $s_{20}$ | $s_4$ | $s_{18}$ | $s_4$ |
| **14** | $s_5$ | $s_{15}$ | $s_1$ | $s_{14}$ | $s_{16}$ | $s_{13}$ | $s_1$ | $s_5$ |
| **15** | $s_1$ | $s_9$ | $s_7$ | $s_{12}$ | $s_{19}$ | $s_{20}$ | $s_{19}$ | $s_7$ |
| **16** | $s_{10}$ | $s_6$ | $s_{15}$ | $s_{19}$ | $s_{10}$ | $s_8$ | $s_{17}$ | $s_1$ |
| **17** | $s_8$ | $s_{17}$ | $s_2$ | $s_{17}$ | $s_{13}$ | $s_{10}$ | $s_{13}$ | $s_9$ |
| **18** | $s_0$ | $s_3$ | $s_{18}$ | $s_8$ | $s_7$ | $s_{15}$ | $s_{10}$ | $s_{14}$ |
| **19** | $s_{14}$ | $s_{19}$ | $s_{17}$ | $s_{15}$ | $s_2$ | $s_2$ | $s_3$ | $s_{15}$ |
| **20** | $s_4$ | $s_{10}$ | $s_{20}$ | $s_{13}$ | $s_4$ | $s_3$ | $s_8$ | $s_{16}$ |
| **21** | $s_3$ | $s_{14}$ | $s_5$ | $s_{18}$ | $s_{14}$ | $s_6$ | $s_{16}$ | $s_{17}$ |

## EK-4.5: Örnek 16 – Öğrenci Tercihleri

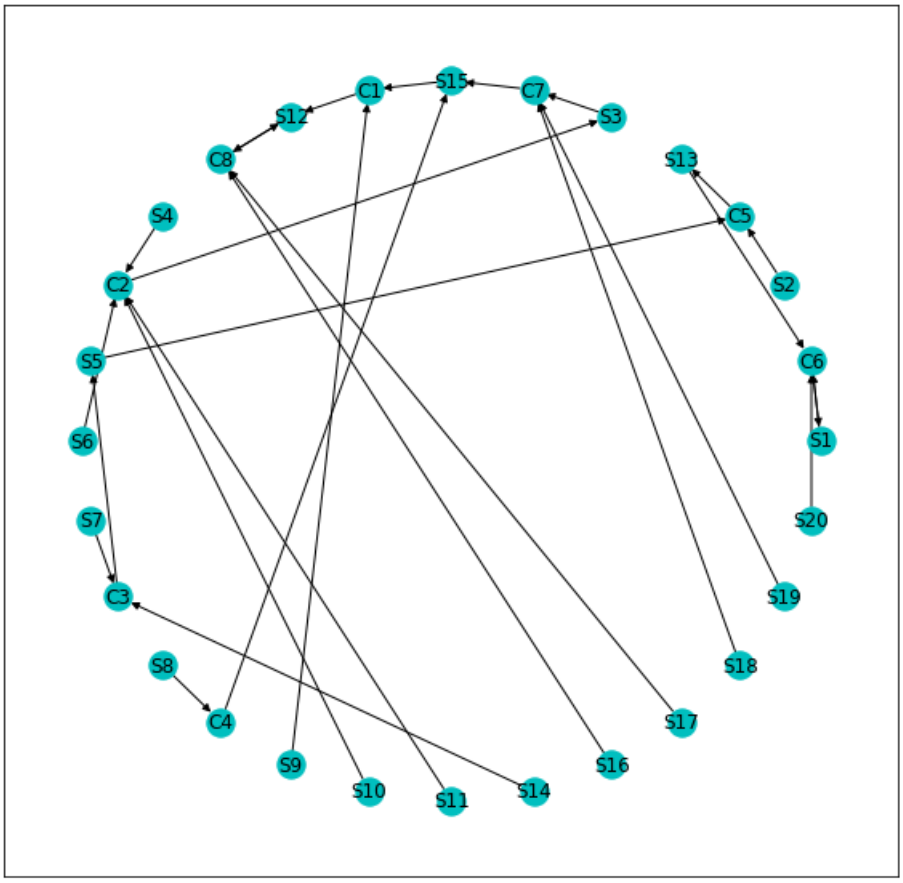
| Öğrenciler / Tercih sırası | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ | $s_6$ | $s_7$ | $s_8$ | $s_9$ | $s_{10}$ | $s_{11}$ | $s_{12}$ | $s_{13}$ | $s_{14}$ | $s_{15}$ | $s_{16}$ | $s_{17}$ | $s_{18}$ | $s_{19}$ | $s_{20}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | $c_6$ | $c_5$ | $c_7$ | $c_2$ | $c_5$ | $c_2$ | $c_3$ | $c_4$ | $c_1$ | $c_2$ | $c_2$ | $c_8$ | $c_6$ | $c_3$ | $c_1$ | $c_8$ | $c_8$ | $c_7$ | $c_7$ | $c_6$ |
| **2** | $c_4$ | $c_7$ | $c_6$ | $c_1$ | $c_1$ | $c_4$ | $c_8$ | $c_1$ | $c_5$ | $c_3$ | $c_7$ | $c_6$ | $c_4$ | $c_5$ | $c_5$ | $c_7$ | $c_1$ | $c_2$ | $c_5$ | $c_3$ |
| **3** | $c_5$ | $c_6$ | $c_2$ | $c_7$ | $c_6$ | $c_8$ | $c_2$ | $c_6$ | $c_3$ | $c_5$ | $c_6$ | $c_5$ | $c_5$ | $c_4$ | $c_4$ | $c_5$ | $c_3$ | $c_1$ | $c_4$ | $c_7$ |
| **4** | $c_1$ | $c_3$ | $c_8$ | $c_8$ | $c_4$ | $c_7$ | $c_4$ | $c_7$ | $c_6$ | $c_8$ | $c_5$ | $c_1$ | $c_3$ | $c_1$ | $c_2$ | $c_4$ | $c_7$ | $c_3$ | $c_1$ | $c_5$ |
| **5** | $c_8$ | $c_4$ | $c_1$ | $c_4$ | $c_2$ | $c_6$ | $c_5$ | $c_5$ | $c_8$ | $c_7$ | $c_4$ | $c_4$ | $c_7$ | $c_7$ | $c_3$ | $c_3$ | $c_2$ | $c_5$ | $c_8$ | $c_4$ |
| **6** | $c_2$ | $c_8$ | $c_5$ | $c_6$ | $c_3$ | $c_5$ | $c_7$ | $c_3$ | $c_2$ | $c_1$ | $c_1$ | $c_2$ | $c_1$ | $c_6$ | $c_8$ | $c_1$ | $c_5$ | $c_8$ | $c_6$ | $c_1$ |
| **7** | $c_7$ | $c_2$ | $c_4$ | $c_5$ | $c_8$ | $c_1$ | $c_1$ | $c_8$ | $c_7$ | $c_6$ | $c_8$ | $c_7$ | $c_8$ | $c_8$ | $c_7$ | $c_6$ | $c_4$ | $c_4$ | $c_2$ | $c_8$ |
| **8** | $c_3$ | $c_1$ | $c_3$ | $c_3$ | $c_7$ | $c_3$ | $c_6$ | $c_2$ | $c_4$ | $c_4$ | $c_3$ | $c_3$ | $c_2$ | $c_2$ | $c_6$ | $c_2$ | $c_6$ | $c_6$ | $c_3$ | $c_2$ |

## EK-4.6: Örnek 16 - Okul Kota Bilgileri ve Okul Tercihleri

| | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ | $c_7$ | $c_8$ |
|---|---|---|---|---|---|---|---|---|
| **kota** | 4 | 3 | 5 | 2 | 3 | 5 | 3 | 3 |

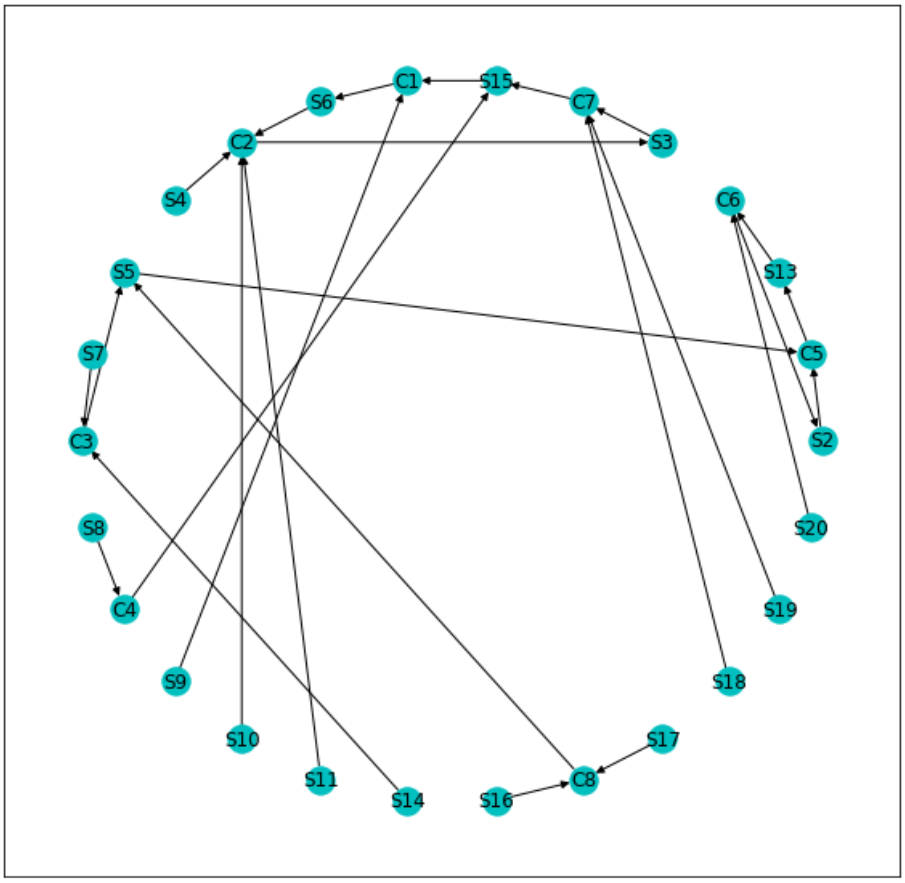
| Okullar / Tercih sırası | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ | $c_7$ | $c_8$ |
|---|---|---|---|---|---|---|---|---|
| **1** | $s_{12}$ | $s_3$ | $s_5$ | $s_{15}$ | $s_{13}$ | $s_1$ | $s_{15}$ | $s_{12}$ |
| **2** | $s_6$ | $s_4$ | $s_{11}$ | $s_2$ | $s_{17}$ | $s_2$ | $s_{20}$ | $s_5$ |
| **3** | $s_{10}$ | $s_{14}$ | $s_{13}$ | $s_{20}$ | $s_{18}$ | $s_6$ | $s_{10}$ | $s_{13}$ |
| **4** | $s_2$ | $s_{10}$ | $s_{19}$ | $s_{14}$ | $s_{12}$ | $s_{15}$ | $s_9$ | $s_{19}$ |
| **5** | $s_{19}$ | $s_5$ | $s_6$ | $s_{12}$ | $s_7$ | $s_9$ | $s_{19}$ | $s_6$ |
| **6** | $s_1$ | $s_9$ | $s_{17}$ | $s_{10}$ | $s_1$ | $s_{12}$ | $s_5$ | $s_4$ |
| **7** | $s_8$ | $s_{19}$ | $s_1$ | $s_4$ | $s_2$ | $s_{13}$ | $s_1$ | $s_{10}$ |
| **8** | $s_3$ | $s_2$ | $s_{12}$ | $s_{17}$ | $s_{19}$ | $s_{17}$ | $s_{11}$ | $s_2$ |
| **9** | $s_4$ | $s_7$ | $s_4$ | $s_6$ | $s_8$ | $s_{10}$ | $s_2$ | $s_1$ |
| **10** | $s_7$ | $s_{20}$ | $s_{16}$ | $s_1$ | $s_{15}$ | $s_4$ | $s_7$ | $s_{16}$ |
| **11** | $s_{13}$ | $s_{15}$ | $s_{20}$ | $s_{19}$ | $s_5$ | $s_5$ | $s_6$ | $s_9$ |
| **12** | $s_{20}$ | $s_8$ | $s_{14}$ | $s_8$ | $s_3$ | $s_{14}$ | $s_{12}$ | $s_7$ |
| **13** | $s_{16}$ | $s_{17}$ | $s_2$ | $s_7$ | $s_4$ | $s_7$ | $s_8$ | $s_{18}$ |
| **14** | $s_{17}$ | $s_6$ | $s_8$ | $s_{11}$ | $s_{14}$ | $s_{20}$ | $s_3$ | $s_{17}$ |
| **15** | $s_{14}$ | $s_{12}$ | $s_3$ | $s_5$ | $s_9$ | $s_{11}$ | $s_{17}$ | $s_{20}$ |
| **16** | $s_5$ | $s_{11}$ | $s_{15}$ | $s_{13}$ | $s_6$ | $s_8$ | $s_4$ | $s_{11}$ |
| **17** | $s_{11}$ | $s_1$ | $s_7$ | $s_{18}$ | $s_{16}$ | $s_{16}$ | $s_{13}$ | $s_{14}$ |
| **18** | $s_{18}$ | $s_{18}$ | $s_{18}$ | $s_3$ | $s_{10}$ | $s_3$ | $s_{18}$ | $s_8$ |
| **19** | $s_{15}$ | $s_{13}$ | $s_9$ | $s_{16}$ | $s_{20}$ | $s_{19}$ | $s_{16}$ | $s_3$ |
| **20** | $s_9$ | $s_{16}$ | $s_{10}$ | $s_9$ | $s_{11}$ | $s_{18}$ | $s_{14}$ | $s_{15}$ |

## EK-4.7: Örnek 16 – 1. Tur Tüm Oyuncuların Tercihleri



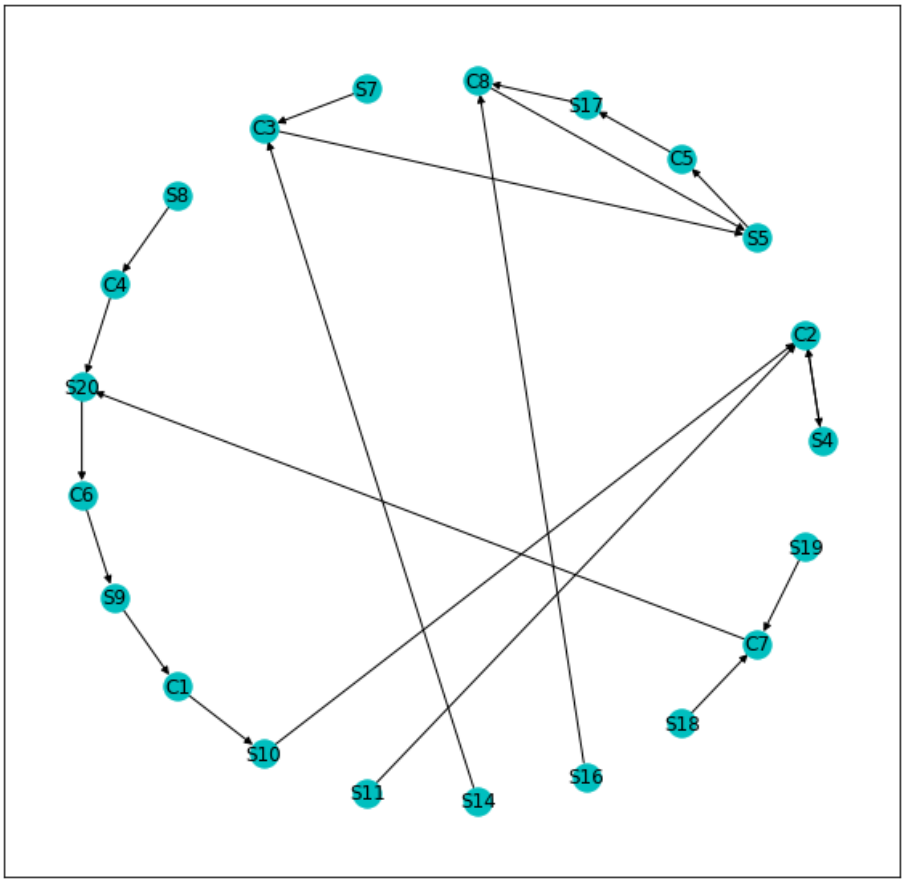

**Dipnot**: Bütün oyuncuların tercihlerinin her tur için ayrı olarak gösterildiği genel şekillerde, öğrenciler tarafından o turda tercih edilmeyen okullar o turun genel şekline dahil edilmemiştir

**EK-4.8: Örnek 16 – 2. Tur Tüm Oyuncuların Tercihleri**

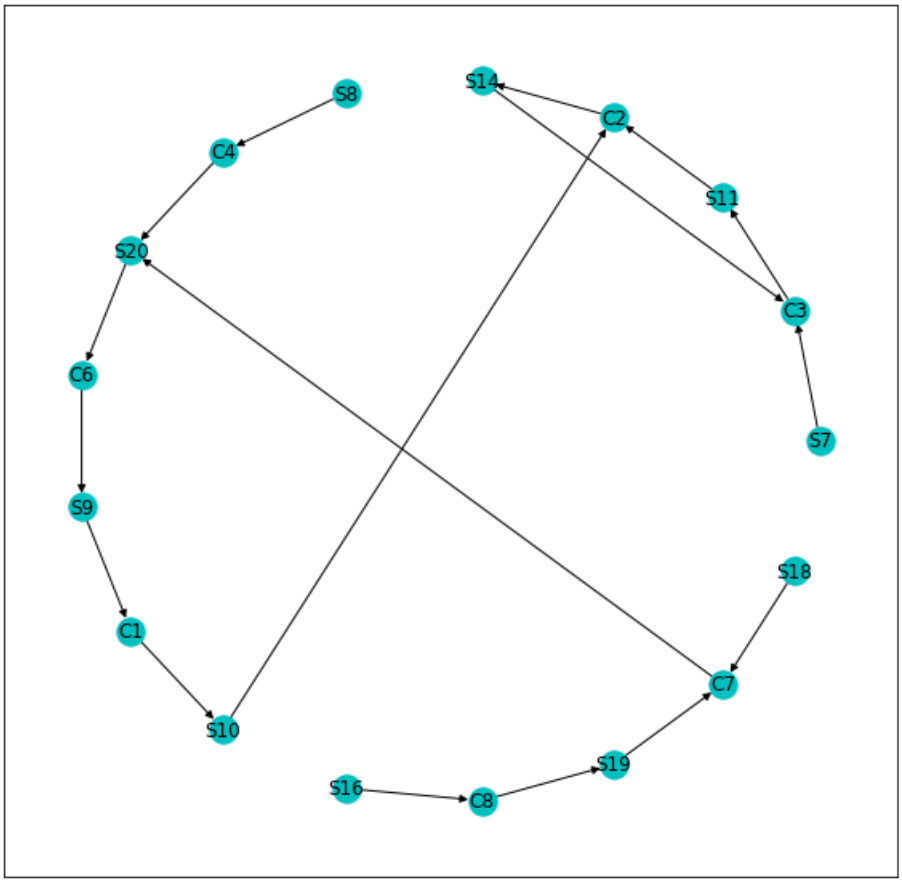



.

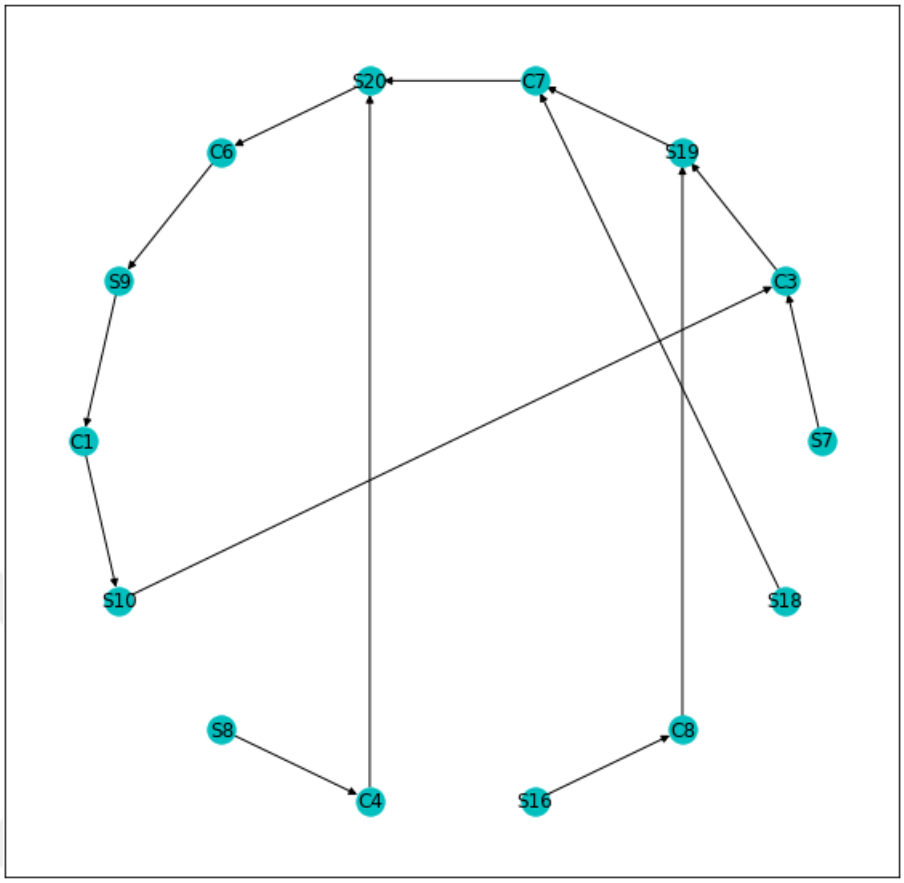
**EK-4.9: Örnek 16 – 3. Tur Tüm Oyuncuların Tercihleri**



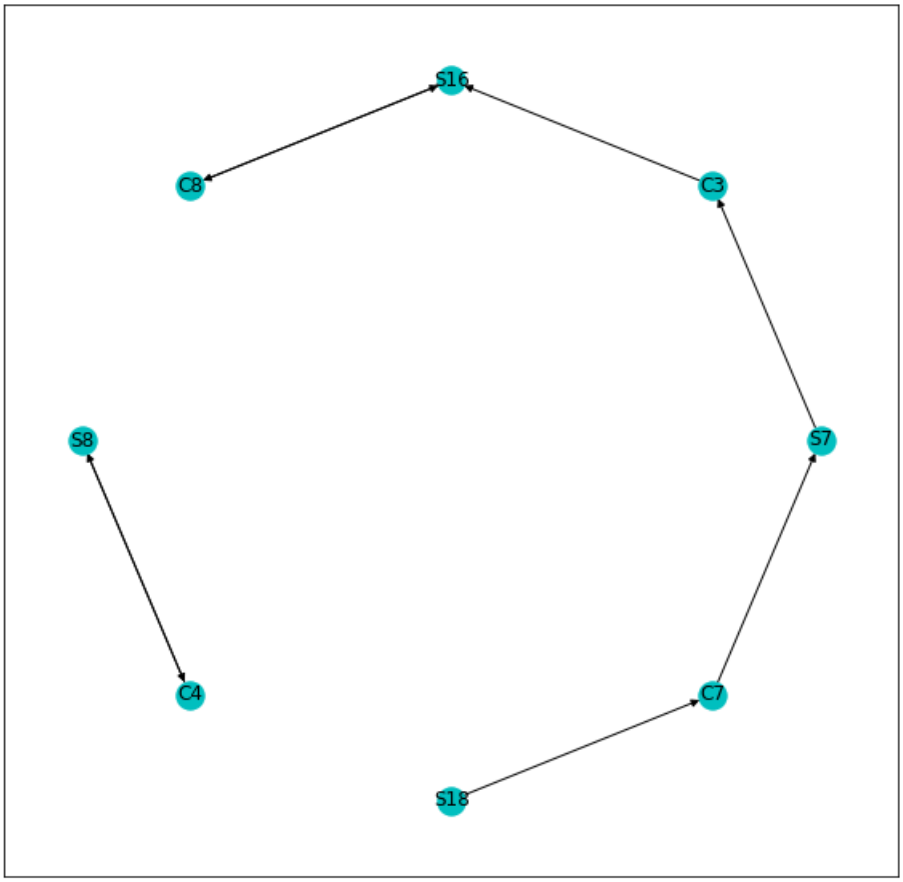
**EK-4.10: Örnek 16 – 4. Tur Tüm Oyuncuların Tercihleri**



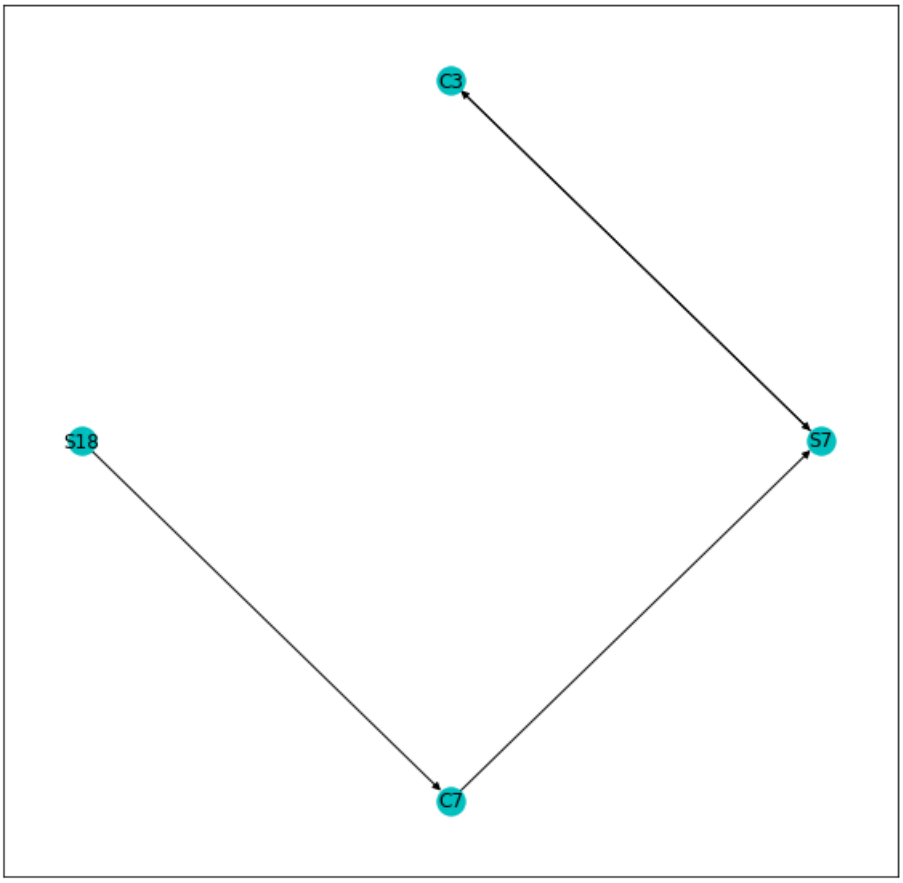
**EK-4.11: Örnek 16 – 5. Tur Tüm Oyuncuların Tercihleri**



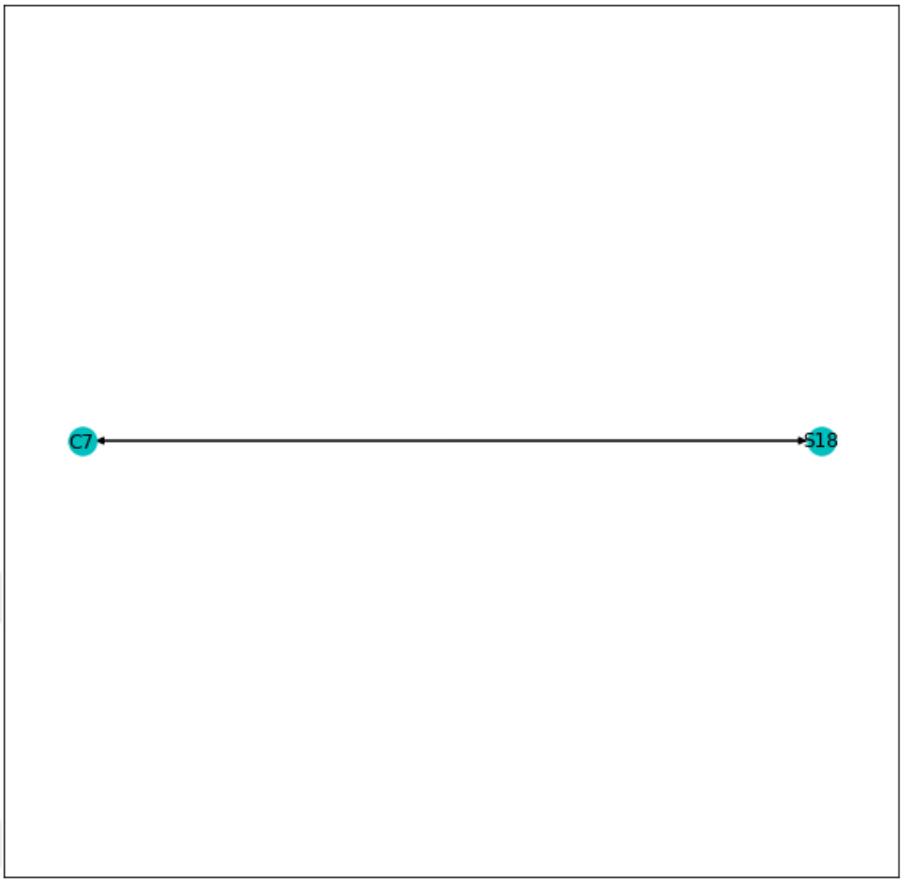
**EK-4.12: Örnek 16 – 6. Tur Tüm Oyuncuların Tercihleri**



**EK-4.13: Örnek 16 – 7. Tur Tüm Oyuncuların Tercihleri**



**EK-4.14: Örnek 16 – 8. Tur Tüm Oyuncuların Tercihleri**



## EK-4.15: Örnek 17 - Öğrenci Tercihleri ve Okul Bilgileri

| Öğrenciler / Tercih sırası | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ | $s_6$ | $s_7$ | $s_8$ | $s_9$ | $s_{10}$ | $s_{11}$ | $s_{12}$ | $s_{13}$ | $s_{14}$ | $s_{15}$ | $s_{16}$ | $s_{17}$ | $s_{18}$ | $s_{19}$ | $s_{20}$ | $s_{21}$ | $s_{22}$ | $s_{23}$ | $s_{24}$ | $s_{25}$ | $s_{26}$ | $s_{27}$ | $s_{28}$ | $s_{29}$ | $s_{30}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | $c_4$ | $c_6$ | $c_6$ | $c_6$ | $c_5$ | $c_2$ | $c_7$ | $c_1$ | $c_4$ | $c_5$ | $c_8$ | $c_5$ | $c_2$ | $c_6$ | $c_7$ | $c_3$ | $c_7$ | $c_8$ | $c_5$ | $c_4$ | $c_7$ | $c_4$ | $c_8$ | $c_4$ | $c_1$ | $c_4$ | $c_8$ | $c_3$ | $c_4$ | $c_6$ |
| **2** | $c_6$ | $c_7$ | $c_7$ | $c_2$ | $c_3$ | $c_5$ | $c_1$ | $c_8$ | $c_3$ | $c_7$ | $c_1$ | $c_8$ | $c_4$ | $c_7$ | $c_6$ | $c_1$ | $c_6$ | $c_5$ | $c_4$ | $c_3$ | $c_2$ | $c_2$ | $c_1$ | $c_5$ | $c_8$ | $c_8$ | $c_1$ | $c_2$ | $c_5$ | $c_7$ |
| **3** | $c_3$ | $c_3$ | $c_2$ | $c_8$ | $c_7$ | $c_1$ | $c_2$ | $c_4$ | $c_5$ | $c_3$ | $c_2$ | $c_2$ | $c_1$ | $c_3$ | $c_8$ | $c_6$ | $c_8$ | $c_4$ | $c_1$ | $c_7$ | $c_3$ | $c_8$ | $c_6$ | $c_7$ | $c_2$ | $c_6$ | $c_4$ | $c_1$ | $c_3$ | $c_1$ |
| **4** | $c_7$ | $c_0$ | $c_3$ | $c_4$ | $c_1$ | $c_0$ | $c_0$ | $c_6$ | $c_6$ | $c_6$ | $c_3$ | $c_0$ | $c_0$ | $c_8$ | $c_5$ | $c_7$ | $c_2$ | $c_0$ | $c_2$ | $c_0$ | $c_6$ | $c_5$ | $c_0$ | $c_1$ | $c_7$ | $c_2$ | $c_0$ | $c_6$ | $c_6$ | $c_5$ |
| **5** | $c_8$ | $c_5$ | $c_8$ | $c_1$ | $c_8$ | $c_3$ | $c_8$ | $c_5$ | $c_0$ | $c_2$ | $c_0$ | $c_3$ | $c_3$ | $c_5$ | $c_4$ | $c_0$ | $c_5$ | $c_7$ | $c_7$ | $c_5$ | $c_4$ | $c_0$ | $c_7$ | $c_0$ | $c_6$ | $c_0$ | $c_3$ | $c_5$ | $c_1$ | $c_3$ |
| **6** | $c_0$ | $c_4$ | $c_5$ | $c_5$ | $c_4$ | $c_4$ | $c_5$ | $c_2$ | $c_1$ | $c_1$ | $c_5$ | $c_4$ | $c_7$ | $c_4$ | $c_3$ | $c_4$ | $c_1$ | $c_6$ | $c_8$ | $c_2$ | $c_8$ | $c_1$ | $c_3$ | $c_6$ | $c_4$ | $c_1$ | $c_2$ | $c_7$ | $c_2$ | $c_4$ |
| **7** | $c_5$ | $c_1$ | $c_1$ | $c_7$ | $c_0$ | $c_8$ | $c_4$ | $c_7$ | $c_2$ | $c_8$ | $c_6$ | $c_7$ | $c_6$ | $c_2$ | $c_0$ | $c_2$ | $c_4$ | $c_1$ | $c_6$ | $c_6$ | $c_5$ | $c_6$ | $c_2$ | $c_8$ | $c_5$ | $c_7$ | $c_5$ | $c_8$ | $c_0$ | $c_0$ |
| **8** | $c_2$ | $c_2$ | $c_4$ | $c_3$ | $c_2$ | $c_7$ | $c_3$ | $c_0$ | $c_8$ | $c_4$ | $c_4$ | $c_1$ | $c_5$ | $c_1$ | $c_1$ | $c_8$ | $c_3$ | $c_3$ | $c_3$ | $c_8$ | $c_0$ | $c_7$ | $c_5$ | $c_3$ | $c_0$ | $c_3$ | $c_6$ | $c_0$ | $c_8$ | $c_8$ |
| **9** | $c_1$ | $c_8$ | $c_0$ | $c_0$ | $c_6$ | $c_6$ | $c_6$ | $c_3$ | $c_7$ | $c_0$ | $c_7$ | $c_6$ | $c_8$ | $c_0$ | $c_2$ | $c_5$ | $c_0$ | $c_2$ | $c_0$ | $c_1$ | $c_1$ | $c_3$ | $c_4$ | $c_2$ | $c_3$ | $c_5$ | $c_7$ | $c_4$ | $c_7$ | $c_2$ |

| | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ | $c_7$ | $c_8$ |
|---|---|---|---|---|---|---|---|---|
| **Kota** | 2 | 5 | 3 | 3 | 6 | 6 | 3 | 2 |
| **Okul Tipi** | Sözel | Sayısal | Dil | Eşit Ağırlık | Dil | Sözel | Dil | Eşit Ağırlık |

## EK-4.16: Örnek 17 - Öğrenci Puanları ve Sıralamalar

| | Sözel | Sayısal | Dil | Eşit Ağırlık |
|---|---|---|---|---|
| $s_1$ | 435,485 | 204,565 | 231,624 | 253,642 |
| $s_2$ | 230,466 | 261,158 | 205,642 | 207,863 |
| $s_3$ | 322,648 | 419,655 | 442,325 | 125,654 |
| $s_4$ | 344,617 | 397,298 | 442,125 | 335,714 |
| $s_5$ | 491,359 | 124,688 | 323,841 | 161,904 |
| $s_6$ | 174,289 | 158,392 | 339,756 | 131,482 |
| $s_7$ | 482,731 | 253,168 | 186,899 | 392,025 |
| $s_8$ | 442,743 | 195,943 | 194,487 | 283,589 |
| $s_9$ | 256,583 | 136,462 | 371,952 | 308,486 |
| $s_{10}$ | 273,613 | 121,528 | 358,611 | 249,756 |
| $s_{11}$ | 129,556 | 104,956 | 466,741 | 249,321 |
| $s_{12}$ | 416,694 | 401,186 | 149,333 | 427,885 |
| $s_{13}$ | 376,238 | 312,917 | 434,456 | 422,777 |
| $s_{14}$ | 147,987 | 108,951 | 171,888 | 122,841 |
| $s_{15}$ | 266,419 | 369,486 | 369,186 | 370,563 |
| $s_{16}$ | 180,628 | 106,726 | 256,173 | 406,253 |
| $s_{17}$ | 138,354 | 162,817 | 476,312 | 232,775 |
| $s_{18}$ | 403,352 | 203,984 | 156,652 | 142,633 |
| $s_{19}$ | 295,421 | 311,686 | 431,581 | 194,985 |
| $s_{20}$ | 291,682 | 372,556 | 481,932 | 422,988 |
| $s_{21}$ | 174,831 | 364,165 | 268,429 | 230,112 |
| $s_{22}$ | 271,284 | 131,587 | 169,592 | 170,948 |
| $s_{23}$ | 172,827 | 267,943 | 482,482 | 247,486 |
| $s_{24}$ | 331,185 | 271,854 | 329,189 | 185,385 |
| $s_{25}$ | 276,682 | 474,265 | 152,845 | 479,321 |
| $s_{26}$ | 261,588 | 321,984 | 238,664 | 348,621 |
| $s_{27}$ | 247,698 | 189,848 | 249,877 | 447,117 |
| $s_{28}$ | 312,198 | 338,786 | 123,557 | 103,315 |
| $s_{29}$ | 382,687 | 205,581 | 378,222 | 458,232 |
| $s_{30}$ | 193,586 | 126,197 | 185,685 | 447,389 |

| | Sözel | Sayısal | Dil | Eşit Ağırlık |
|---|---|---|---|---|
| 1 | $s_5$ | $s_{25}$ | $s_{23}$ | $s_{25}$ |
| 2 | $s_7$ | $s_3$ | $s_{20}$ | $s_{29}$ |
| 3 | $s_8$ | $s_{12}$ | $s_{17}$ | $s_{30}$ |
| 4 | $s_1$ | $s_4$ | $s_{11}$ | $s_{27}$ |
| 5 | $s_{12}$ | $s_{20}$ | $s_3$ | $s_{12}$ |
| 6 | $s_{18}$ | $s_{15}$ | $s_4$ | $s_{20}$ |
| 7 | $s_{29}$ | $s_{21}$ | $s_{13}$ | $s_{13}$ |
| 8 | $s_{13}$ | $s_{28}$ | $s_{19}$ | $s_{16}$ |
| 9 | $s_4$ | $s_{26}$ | $s_{29}$ | $s_7$ |
| 10 | $s_{24}$ | $s_{13}$ | $s_9$ | $s_{15}$ |
| 11 | $s_3$ | $s_{19}$ | $s_{15}$ | $s_{26}$ |
| 12 | $s_{28}$ | $s_{24}$ | $s_{10}$ | $s_4$ |
| 13 | $s_{19}$ | $s_{23}$ | $s_6$ | $s_9$ |
| 14 | $s_{20}$ | $s_2$ | $s_{24}$ | $s_8$ |
| 15 | $s_{25}$ | $s_7$ | $s_5$ | $s_1$ |
| 16 | $s_{10}$ | $s_{29}$ | $s_{21}$ | $s_{10}$ |
| 17 | $s_{22}$ | $s_1$ | $s_{16}$ | $s_{11}$ |
| 18 | $s_{15}$ | $s_{18}$ | $s_{27}$ | $s_{23}$ |
| 19 | $s_{26}$ | $s_8$ | $s_{26}$ | $s_{17}$ |
| 20 | $s_9$ | $s_{27}$ | $s_1$ | $s_{21}$ |
| 21 | $s_{27}$ | $s_{17}$ | $s_2$ | $s_2$ |
| 22 | $s_2$ | $s_6$ | $s_8$ | $s_{19}$ |
| 23 | $s_{30}$ | $s_9$ | $s_7$ | $s_{24}$ |
| 24 | $s_{16}$ | $s_{22}$ | $s_{30}$ | $s_{22}$ |
| 25 | $s_{21}$ | $s_{30}$ | $s_{14}$ | $s_5$ |
| 26 | $s_6$ | $s_5$ | $s_{22}$ | $s_{18}$ |
| 27 | $s_{23}$ | $s_{10}$ | $s_{18}$ | $s_6$ |
| 28 | $s_{14}$ | $s_{14}$ | $s_{25}$ | $s_3$ |
| 29 | $s_{17}$ | $s_{16}$ | $s_{12}$ | $s_{14}$ |
| 30 | $s_{11}$ | $s_{11}$ | $s_{28}$ | $s_{28}$ |

## EK-4.17: Gale-Shapley Öğrenci Optimal Durağan Mekanizması Geliştirilen Python Algoritması Kodları

### Import Necessary Library

In [ ]:

```
import pandas as pd
import numpy as np
import copy
```

### Structure of the Model

In [ ]:

```
Number_Of_Students = 30
Number_Of_Pref = 8
Number_of_School = Number_Of_Pref
```

### Student Pref

In [ ]:

```
np.random.seed(9)

students = np.arange(1,Number_Of_Students+1) #number of student
std_pref = pd.DataFrame(index = range(1, Number_Of_Pref+2))

for std in students:

    pref_list = []

    while len(pref_list) < Number_Of_Pref+1:    #number of pref
        pref = np.random.randint(0,Number_Of_Pref+1)    #number of pref
        if "C" + str(pref) in pref_list:
            continue
        else:
            pref_list.append(("C" + str(pref)))
            if pref_list[0] == "C0":
                del pref_list[0]

    std_pref["S"+ str(std)] = pref_list

std_pref
```

### School Pref

In [ ]:

```
sch_pref= pd.DataFrame(index = range(1, Number_Of_Students + 2))
school = np.arange(1,Number_of_School + 1) #number of school
for sch in school:

    pref_list = []

    while len(pref_list) < Number_Of_Students + 1:    #number of pref
        pref = np.random.randint(0, Number_Of_Students + 1)    #number of pref
        if "S" + str(pref) in pref_list:
            continue
        else:
            pref_list.append(("S" + str(pref)))
            if pref_list[0] == "S0":
                del pref_list[0]
    sch_pref["C"+ str(sch)] = pref_list

sch_pref
```

### Student School Pref Table

In [ ]:

```
std_sch_pref = pd.DataFrame(index = std_pref.index, columns = std_pref.columns)

for row in std_sch_pref.index:
    std_sch_pref.loc[row] = [ {} for i in range(len(std_sch_pref.columns))]


for col in std_pref.columns:

    for row in std_pref.index:

        school = std_pref.loc[row, col]

        if school == "C0":
            std_sch_pref.loc[row, col][school] = [row]
            continue



        S0_index   = pd.Index(sch_pref[school]).get_loc("S0") + 1
        pref_index = pd.Index(sch_pref[school]).get_loc(col) + 1


        if pref_index < S0_index:
            index = pref_index
        else:
            index = 0

        std_sch_pref.loc[row, col][school] = [index, row, col]

for col in std_sch_pref.columns:

    dictionary = {}

    for row in std_sch_pref.index:

        school = std_sch_pref.loc[row, col]

        dictionary.update(school)

    C0_index = dictionary["C0"][0]

    for row in std_sch_pref.index:

        schandpref = std_sch_pref.loc[row, col]
        sch = list(schandpref.keys())[0]


        if len(list(schandpref.values())[0]) > 1 and list(schandpref.values())[0][1] >

            std_sch_pref.loc[row, col][sch][1] = 0

std_sch_pref
```

### School Information

In [ ]:

```
while True:

    random_quota_list = np.random.multinomial(Number_Of_Students, [1/int(Number_Of_Pref
    quota_list = random_quota_list[0]

    if 0 in quota_list:
        continue
    else:
        break

sch_info = pd.DataFrame(columns =sch_pref.columns)
sch_info.loc["quota"] = quota_list
sch_info.loc[1] = [ [] for i in range(len(sch_pref.columns))]
sch_info
```

### First Assigments Codes

In [ ]:

```
for student in std_sch_pref.columns:  # S1, S2, S3, S4, S5 ......

    studentPref = std_sch_pref.loc[1, student]
    school      = list(studentPref.keys())[0]
    stdpref     = list(studentPref.values())[0][1]
    schpref     = list(studentPref.values())[0][0]

    sch_info.loc[1, school].append({student: [stdpref, schpref, school]})
    sch_info.loc[1, school] = sorted(sch_info.loc[1,school], key = lambda e: list(e.val


del_list = []
for school in sch_info.columns:  # C1, C2, C3, ......üzerinde dolaşıyor.

    del_list2 = []
    placed = sch_info.loc[1, school]

    for std_s0 in placed:

        if list(std_s0.values())[0][1] == 0:

            del_list2.append(std_s0)


    numberofstudent = len(placed)
    exceptS0student = numberofstudent - len(del_list2)
    quota           = sch_info.loc["quota", school]
    excessquota     = exceptS0student - quota

    for i in range(excessquota):

        del_list2.append(placed[-(i+1)])

    del_list.append(del_list2)

sch_info
```

## Model Algorithms

In [ ]:

```
model_round = 2

while True:

    Current_Placed = []

    for col in sch_info.columns:
        p_placed = sch_info.loc[model_round - 1, col]
        Current_Placed.append(p_placed)


    New_Placed = []
    for x in range(len(Current_Placed)):

        if len(Current_Placed[x]) > 0:

            p_placed = Current_Placed[x]
            d_list   = del_list[x]
            n_placed = [k for k in p_placed if k not in d_list]
            New_Placed.append(n_placed)

        else:
            New_Placed.append(Current_Placed[x])

    if Current_Placed == New_Placed:

        sch_info_update = copy.deepcopy(sch_info)
        for col in sch_info_update.columns:
                for row in sch_info_update.index[1:]:
                    sch_info_update.loc[row, col] = []

        for col in sch_info.columns:
            for row in sch_info.index[1:]:
                ass = sch_info.loc[row, col]
                if len(ass) > 0:

                    for x in ass:
                        student = list(x.keys())[0]
                        order   = list(x.values())[0][1]

                        value = (student, order)

                        sch_info_update.loc[row, col].append(value)

        break

    else:

        new_row = [ [] for s in range(len(sch_info.columns))]
        sch_info.loc[model_round] = new_row

        # New Placed'i yukarıda oluşturulan yeni satıra ekliyoruz.
        for x in New_Placed:

            if len(x) > 0:

                for y in x:
                    school = list(y.values())[0][2]
                    sch_info.loc[model_round, school].append(y)
```

```
60
61
62          # Del list'in içindekiler için sonraki tercihlerine atama yapılması
63          for k in del_list:
64
65              if len(k) > 0:
66
67                  for t in k:
68                      n_pref_order = list(t.values())[0][0] + 1
69                      student      = list(t.keys())[0]
70                      n_pref       = std_sch_pref.loc[n_pref_order, student]
71
72                      if list(n_pref.keys())[0] == "C0":
73                          continue
74
75                      else:
76                          sch = list(n_pref.keys())[0]
77                          school_pref = list(n_pref.values())[0][0]
78                          n_placed = {student: [n_pref_order, school_pref, sch]}
79                          sch_info.loc[model_round, sch].append(n_placed)
80                          sch_info.loc[model_round, sch] = sorted(sch_info.loc[model_rou
81                                                                  key = lambda e: list(e.valu
82
83
84
85          del_list = []
86
87
88          for school in sch_info.columns:  # C1, C2, C3, ......üzerinde dolaşıyor.
89
90              del_list2 = []
91              placed = sch_info.loc[model_round, school]
92
93
94              for std_s0 in placed:
95
96                  if list(std_s0.values())[0][1] == 0:
97
98                      del_list2.append(std_s0)
99
100
101             numberofstudent = len(placed)
102             exceptS0student = numberofstudent - len(del_list2)
103             quota           = sch_info.loc["quota", school]
104             excessquota     = exceptS0student - quota
105
106
107             for i in range(excessquota):
108
109                 del_list2.append(placed[-(i+1)])
110
111             del_list.append(del_list2)
112
113     model_round += 1
114
115 sch_info_update
```

## EK-4.18: Gale-Shapley Okul Optimal Durağan Mekanizması Geliştirilen Python Algoritması Kodları

### Import Necessary Library

In [ ]:

```
import pandas as pd
import numpy as np
import copy
```

### Structure of the Model

In [ ]:

```
Number_Of_Students = 20
Number_Of_Pref = 8
Number_of_School = Number_Of_Pref
```

### Student Preference

In [ ]:

```
np.random.seed(1)

students = np.arange(1,Number_Of_Students+1) #number of student
std_pref_table = pd.DataFrame(index = range(1, Number_Of_Pref+2))

for std in students:

    pref_list = []

    while len(pref_list) < Number_Of_Pref+1:    #number of pref
        pref = np.random.randint(0,Number_Of_Pref+1)    #number of pref
        if "C" + str(pref) in pref_list:
            continue
        else:
            pref_list.append(("C" + str(pref)))
            if pref_list[0] == "C0":
                del pref_list[0]

    std_pref_table["S"+ str(std)] = pref_list

school = np.arange(1,Number_of_School + 1) #number of student

std_pref_table
```

### School Preference

In [ ]:

```
sch_pref_table = pd.DataFrame(index = range(1, Number_Of_Students + 2))
for sch in school:

    pref_list = []

    while len(pref_list) < Number_Of_Students + 1:    #number of pref
        pref = np.random.randint(0, Number_Of_Students + 1)    #number of pref
        if "S" + str(pref) in pref_list:
            continue
        else:
            pref_list.append(("S" + str(pref)))
            if pref_list[0] == "S0":
                del pref_list[0]
    sch_pref_table["C"+ str(sch)] = pref_list



sch_pref_table
```

### School Student Pref Table

In [ ]:

```
sch_std_pref_table = pd.DataFrame(index = sch_pref_table.index, columns = sch_pref_tabl

# Tüm hücrelere boş sözlük atanaması
for row in sch_std_pref_table.index:
    sch_std_pref_table.loc[row] = [ {} for i in range(len(sch_std_pref_table.columns))]


for col in sch_pref_table.columns:

    for row in sch_pref_table.index:

        student = sch_pref_table.loc[row, col]

        if student == "S0":
            sch_std_pref_table.loc[row, col][student] = [row]
            continue

        S0_index       = pd.Index(sch_pref_table[col]).get_loc("S0") + 1
        sch_pref_index = pd.Index(sch_pref_table[col]).get_loc(student) + 1



        C0_index       = pd.Index(std_pref_table[student]).get_loc("C0") + 1
        std_pref_index = pd.Index(std_pref_table[student]).get_loc(col) + 1

        if std_pref_index < C0_index:
            index = std_pref_index
        else:
            index = 0


        if sch_pref_index < S0_index:


            sch_std_pref_table.loc[row, col][student] = [index, row, col]

        else:
            sch_std_pref_table.loc[row, col]["S0"] = [row]

sch_std_pref_table
```

### School Information

In [ ]:

```
while True:

    random_quota_list = np.random.multinomial(Number_Of_Students, [1/int(Number_Of_Pref
    quota_list = random_quota_list[0]

    if 0 in quota_list:
        continue
    else:
        break

sch_info = pd.DataFrame(columns =sch_pref_table.columns)
sch_info.loc["quota"] = quota_list
sch_info.loc["assigments"] = [[] for k in range(len(sch_pref_table.columns))]
sch_info
```

### Student Information

In [ ]:

```
std_info = pd.DataFrame(columns =std_pref_table.columns)
std_info.loc[1] = [[] for k in range(len(std_pref_table.columns))]
std_info
```

## Model Algorithms

In [ ]:

```
m_round = 1

while True:

    # Okulların mevcut kotalarına göre tercih listelerinin school_prefs
    # listesinin içerisine atanması
    for school in sch_std_pref_table.columns:

        if m_round > 1 and  school not in refused_school:
            continue

        school_pref_list = [pref for pref in list(sch_std_pref_table[school]) if pref
        school_quota     = sch_info.loc["quota", school] - len(sch_info.loc["assigment
        school_prefs     = school_pref_list[:(school_quota)]


        # school_prefs listesi içerisindeki okul tercihlerine göre
        # ilgili öğrencilere tercihlerin iletilmesi
        for preferred_student in school_prefs:

            student            = list(preferred_student.keys())[0]
            if student == "S0":
                break

            student_pref_order = list(preferred_student.values())[0][0]
            value = (school, student_pref_order, student)
            std_info.loc[m_round, student].append(value)
            sch_info.loc["assigments", school].append(student)   #Kota kontrolü için o


        # Teklif yapan okulların yapmış oldukları teklifler için
        # sch_std_pref_table tablasunun güncellenmesi

        school_pref_list2 = [pref for pref in list(sch_std_pref_table[school])]

        count_0 = school_pref_list2.count(0)
        count = 0

        for x in range(school_quota):

            if (count_0 + x) > len(school_pref_list2)-1:
                break
            school_pref_list2[count_0 + x] = 0



        sch_std_pref_table[school] = school_pref_list2

    # Teklif yapan okulların öğrenci tercih sırasına göre sıralanması
    # İlgili okul Öğrenci tercihlerinde C0'dan sonra ise öğrenci tercih sırası 0 oluyo

    for student in std_info.columns:
        std_info.loc[m_round, student] = sorted(std_info.loc[m_round, student],
                                                key = lambda e: e[1])

    # Teklifleri reddedilen ve geçici kabul edilen öğrencilerin
    # bilgilierinin iki ayrı liste şeklinde elde edilmesi

    refused_school  = []
```

```
    accepted_school = []

    for student in std_info.columns:

        list1 = []
        biddings = std_info.loc[m_round, student]


        if len(biddings) == 0:
            accepted_school.append([])
            continue

        if len(biddings) == 1:

            if biddings[0][1] == 0:

                refused_school.append(biddings[0])
                accepted_school.append([])
            else:
                list1.append(biddings[0])
                accepted_school.append(list1)

        if len(biddings) > 1:

            if biddings[0][1] != 0:
                list1.append(biddings[0])
                accepted_school.append(list1)
                refused_school.extend(biddings[1:])
                continue

            else:

                for bid in biddings:

                    if bid[1] == 0:
                        refused_school.append(bid)
                    else:
                        list1.append(bid)
                        accepted_school.append(list1)

                        index = biddings.index(bid)
                        refused_school.extend(biddings[index+1:])
                        break

                # Tekliflerin hepsinin öğrenci tarafından
                # reddedilme durumu
                except_0 = 0  # Teklifler içerisinde 0 hariç olanlar için sayaç
                for bid in biddings:
                    if bid[1] == 0:
                        continue
                    else:
                        except_0 +=1

                if except_0 == 0:  # Sayaç 0'a eşit ise tüm teklifler 0 demektir. KAbu
                    accepted_school.append([])

                #----------------------------------------

    if accepted_school == list(std_info.loc[m_round]):

        std_info_update = copy.deepcopy(std_info)
```

```
121
122         for col in std_info_update.columns:
123                 for row in std_info_update.index:
124                     std_info_update.loc[row, col] = []
125
126         for col in std_info.columns:
127             for row in std_info.index:
128
129                 ass = std_info.loc[row, col]
130
131                 if len(ass) > 0:
132
133                     for x in ass:
134                         school = x[0]
135                         order  = x[1]
136
137                         value = (school, order)
138
139                         std_info_update.loc[row, col].append(value)
140         break
141
142     else:
143         std_info.loc[m_round + 1] = [[] for k in range(len(std_pref_table.columns))]
144         std_info.loc[m_round + 1] = accepted_school
145
146
147     # Reddedilen okulların kotalarının güncellenmesi
148     for refused in refused_school:
149         res_school = refused[0]
150         std        = refused[2]
151         sch_info.loc["assigments",res_school ].remove(std)
152
153     # refused_school listesinin sadece okulları içerecek şekilde
154     # güncellenmesi, çünkü kad satırın başlangıcındaki şarta uyum sağlanması gerekiyor
155     for k in range(len(refused_school)):
156         refused_school[k] =  refused_school[k][0]
157
158     m_round +=1
159
160 std_info_update
```

## EK-4.19: En Yüksek Değiş Tokuş Döngüleri Mekanizması Geliştirilen Python Algoritması Kodları

### Import Necessary Library

In [ ]:

```
import pandas as pd
import numpy as np
import copy
```

### Structure of the Model

In [ ]:

```
Number_Of_Students = 20
Number_Of_Pref = 8
Number_of_School = Number_Of_Pref
```

### Student Preference

In [ ]:

```
np.random.seed(1)
students = np.arange(1, Number_Of_Students+1) #number of student
std_pref = pd.DataFrame(index = range(1, Number_Of_Pref+1))

for std in students:

    pref_list = []

    while len(pref_list) < Number_Of_Pref:    #number of pref
        pref = np.random.randint(1, Number_Of_Pref + 1)    #number of pref
        if "C" + str(pref) in pref_list:
            continue
        else:
            pref_list.append(("C" + str(pref)))

    std_pref["S"+ str(std)] = pref_list

std_pref_table = std_pref
std_pref_table
```

### School Preference

In [ ]:

```
school = np.arange(1,Number_of_School + 1) #number of school
sch_pref= pd.DataFrame(index = range(1, Number_Of_Students + 1))

for sch in school:

    pref_list = []

    while len(pref_list) < Number_Of_Students:    # number of pref
        pref = np.random.randint(1, Number_Of_Students + 1)    #number of pref
        if "S" + str(pref) in pref_list:
            continue
        else:
            pref_list.append(("S" + str(pref)))

    sch_pref["C"+ str(sch)] = pref_list

sch_pref_table = sch_pref
sch_pref_table
```

### School Information

In [ ]:

```
while True:

    random_quota_list = np.random.multinomial(Number_Of_Students*1.4, [1/int(Number_Of_
    quota_list = random_quota_list[0]

    if 0 in quota_list:
        continue
    else:
        break

sch_info = pd.DataFrame(columns =sch_pref_table.columns)
sch_info.loc["quota"] = quota_list
sch_info
```

## Model Algorithms _ 1

In [ ]:

```
assigned_students = []
student_list = list(std_pref_table.columns)
all_cycles = {}

m_round = 1
df_list = []

while True:

    cycle_list = []

    # Bu döngüye ait öğrenci ve okulların uygun ilk tercihlerinin oluşturulması-------
    StudentFirstPref = {}
    SchoolFirstPref  = {}
    ## Öğrenci tercihleri (Şart: okul kotası uygun olan ilk tercih)
    for student in std_pref_table.columns:

        for index_std_pref in std_pref_table.index:

            student_pref  = std_pref_table.loc[index_std_pref, student]
            pref_quota    = sch_info.loc["quota" , student_pref]

            if pref_quota > 0:

                StudentFirstPref[student] = student_pref
                break

    ## Okul Tercihleri (Daha önce atanmamış olan öğrenciler)
    ## Bu öğrencier sch_pref_table'ın kolonundan alınıyor
    for school in sch_pref_table.columns:

        for index_sch_pref in sch_pref_table.index:

            school_pref  = sch_pref_table.loc[index_sch_pref, school]

            if school_pref not in assigned_students:

                SchoolFirstPref[school] = school_pref
                break

    preflist = {**StudentFirstPref, **SchoolFirstPref}  # Toplam öğrenci ve okul terci


    # Bu döngüye ait Cycle_Table oluşturulması ------------------------------------
    cycle_table = pd.DataFrame(columns = list(StudentFirstPref.keys()))
    cycle_table.loc[1]                 = list(StudentFirstPref.keys())

    loop_counter = len(preflist)


    for x in range(loop_counter+1):
        cycle_table.loc[x+2] = 0

        for row in range(len(list(cycle_table.loc[2]))):

            value = cycle_table.iloc[x, row ]
            pref  = preflist[value]
            cycle_table.iloc[x+1, row] = pref

```

```
60
61      # Bir önceki döngü sisteminde oluşturulan cycle_table içerisindeki
62      # döngülerin bulunması-----------------------------------------------
63      cycle_list = []
64      for col in cycle_table.columns:
65
66          cycle_values = list(cycle_table[col])
67
68
69          for count in range(len(cycle_values)):
70              if count == len(cycle_values) - 1:
71                  break
72
73              reference_value = cycle_values[0]
74              benc_values     = cycle_values[count + 1]
75              if reference_value == benc_values:
76                  cycle = cycle_values[:count+2]
77                  cycle_list.append(cycle)
78                  break
79
80
81      # Bir önceki döngü sisteminde oluşturulan cycle_list içerisindeki
82      # tekrar eden döngüelerin silinmesi-----------------------------------------------
83      for x in cycle_list:
84          value1 = x
85
86          for y in cycle_list:
87              value2 = y
88
89              if x == y:
90                  continue
91
92              if set(value1) == set(value2):
93                  cycle_list.remove(value2)
94
95      all_cycles[m_round] = cycle_list
96
97      # Atanan öğrenci ve okulların cycle_list'ten çekilerek
98      # iki ayrı listede ayrıştırılması
99      ass_pref_list_std = []
100     ass_pref_list_sch = []
101
102     for ass_pref in cycle_list:
103
104         for ass in ass_pref:
105
106             if ass[0] == "S" and ass not in ass_pref_list_std:
107
108                 ass_pref_list_std.append(ass)
109             elif ass[0] == "C":
110                 ass_pref_list_sch.append(ass)
111
112     # Okul kotalarının güncellenmesi
113     for ass_sch in ass_pref_list_sch:
114         sch_info.loc["quota" , ass_sch] -=1
115
116
117     for ass_std in ass_pref_list_std:
118         assigned_students.append(ass_std)
119         del std_pref_table[ass_std]
120
```

```
121
122     df_list.append(cycle_table)
123     m_round +=1
124
125     if len(assigned_students) == len(student_list) or len(cycle_table.columns) ==0:
126         break
127
128
129
130 all_cycles
```

## Model Algorithms _ 2

In [ ]:

```
1  all_cycles_new = {}
2  for m_round in all_cycles.keys():
3
4      if len(all_cycles[m_round]) == 1:
5
6          pref_list = all_cycles[m_round][0]
7          edge_list = []
8
9          for pref_index in range(len(pref_list)):
10
11             if pref_index == len(pref_list) - 1:
12                 break
13
14             edge = (pref_list[pref_index], pref_list[pref_index + 1])
15             edge_list.append(edge)
16
17         all_cycles_new[m_round] =    edge_list
18
19     else:
20
21         cycle_number = 1
22
23         for pref_list in all_cycles[m_round]:
24             edge_list = []
25
26             for pref_index in range(len(pref_list)):
27
28                 if pref_index == len(pref_list) - 1:
29                     continue
30
31                 edge = (pref_list[pref_index], pref_list[pref_index + 1])
32                 edge_list.append(edge)
33                 alfa = str(m_round) + "." + str(cycle_number)
34
35             all_cycles_new[str(m_round) + "." + str(cycle_number)] =    edge_list
36             cycle_number +=1
37 all_cycles_new
```

In [ ]:

```
1 for key in all_cycles_new:
2     print(key, "---->", all_cycles_new[key] )
```

### Graphical Representation_1

In [ ]:

```
import networkx as nx
import matplotlib.pyplot as plt

for keys in all_cycles_new:

    g = nx.DiGraph()

    g.add_edges_from(all_cycles_new[keys])
    pos = nx.circular_layout(g)

    plt.figure(1)

    fig = plt.figure(facecolor="w")
    ax = fig.add_subplot(111)
    plt.title("Round" + ": " + str(keys), fontsize = 15, fontweight = "bold")

    # node_color = ('b' as blue) ('g' as green) ('r' as red) ('c' as cyan) ('m' as mage
    # ('m' as magenta) ('y' as yellow )('k' as black) ('w' as white)

    nx.draw_networkx(g, pos, with_labels = True, ax = ax, node_size = 1000, node_color

#     color = "k", linewidth = 0.5
#     ax.grid()  color = "k", linewidth = 0.5
    plt.show()
```

### Graphical Representation_2

In [ ]:

```
m_round = len(df_list)
all_cycles_new2 = {}

for i in range(m_round):

    cycl_list = []
    df = df_list[i]

    for col in df.columns:
        for row in df.index:

            if  row == df.index[-1]:
                break


            value1 = df.loc[row, col]

            value2 = df.loc[row+1, col]
            edges = (value1, value2)
            cycl_list.append(edges)

    all_cycles_new2[i+1] =  cycl_list
```

In [ ]:

```
import networkx as nx
import matplotlib.pyplot as plt

for keys in all_cycles_new2:

    g = nx.DiGraph()

    g.add_edges_from(all_cycles_new2[keys])
    print(list(nx.find_cycle(g, orientation="original")))
    pos = nx.circular_layout(g)

    plt.figure(1)

    fig = plt.figure(facecolor="w", figsize = (9,9))
    ax = fig.add_subplot(111)
    plt.title("Round" + ": " + str(keys), fontsize = 15, fontweight = "bold")
    nx.draw_networkx(g, pos, with_labels = True, ax = ax, node_color = "c")

#     ax.tick_params(left=True, bottom=True)
#     ax.grid(color = "k", linewidth = 0.5)
    plt.show()
```

## EK-4.20: Çok Kategorili Seri Diktatörlük Mekanizması Geliştirilen Python Algoritması Kodları

### Import Necessary Library

In [ ]:

```
import pandas as pd
import numpy as np
import copy
import time
```

### Structure of the Model

In [ ]:

```
Number_Of_Students = 30
Number_Of_Pref = 8
Number_of_School = Number_Of_Pref
Type_List = ["Social","Math","Language","Equal"]
```

### Student Prefs

In [ ]:

```
np.random.seed(7)
students = np.arange(1,Number_Of_Students+1) #number of student
std_pref = pd.DataFrame(index = range(1, Number_Of_Pref+2))

for std in students:

    pref_list = []

    while len(pref_list) < Number_Of_Pref+1:    #number of pref
        pref = np.random.randint(0,Number_Of_Pref+1)    #number of pref
        if "C" + str(pref) in pref_list:
            continue
        else:
            pref_list.append(("C" + str(pref)))
            if pref_list[0] == "C0":
                del pref_list[0]

    std_pref["S"+ str(std)] = pref_list

std_pref_table = copy.deepcopy(std_pref)

std_pref_table
```

In [ ]:

```
score_table = pd.DataFrame(index = std_pref.columns, columns = Type_List) #number of Ty

for a in score_table.index:
    for b in score_table.columns:
        score_table.at[a,b]= np.random.uniform(150,500)
score_table
```

### Student Order Table

In [ ]:

```
std_order_table = pd.DataFrame(index = range(1, Number_Of_Students + 1), columns = Type

for typ in std_order_table.columns:
    type_columns = score_table.sort_values(typ, ascending = False)
    std_order_table[typ] = type_columns.index

std_order_table
```

### School Information

In [ ]:

```
school = np.arange(1,Number_of_School+1)   #number of school / same with pref
school_list = ["C"+str(i) for i in school]


# Random quotas for school ------------------
while True:

    random_quota_list = np.random.multinomial(Number_Of_Students, [1/int(Number_Of_Pref
    quota_list = random_quota_list[0]

    if 0 in quota_list:
        continue
    else:
        break


sch_info = pd.DataFrame(index = school_list, columns=["Quota","School Type"])
sch_info["Quota"] = quota_list

School_Type = Type_List.copy()

while len(School_Type) < len(sch_info.index):
    random_variable = np.random.randint(0,len(Type_List))
    School_Type.append(Type_List[random_variable])

sch_info["School Type"] = School_Type
sch_info = sch_info.T
sch_info
```

## Model Algorithms

In [ ]:

```
m_round = 1
assigment_list = []


while True:

    sch_info.loc[m_round] = [[] for k in range(len(sch_info.columns)) ]

    for sch_type in std_order_table.columns:


        for std_order in std_order_table.index:

            student = std_order_table.loc[std_order, sch_type]


            for std_pref in std_pref_table[student]:

                # mevcut tercih C0 ise sonraki öğrenciye geç.-----
                if std_pref == "C0":
                    break

                # mevcut tercihe ait okul tipi başlangıçtan gelen sch_type'ye
                # eşit değilse sonraki tercihe geç.
                current_school_type = sch_info.loc["School Type", std_pref]
                if current_school_type != sch_type:
                    continue


                # mevcut std_pref yani okula ait açık otayı kontrol et !!!!
                # Eğer açık kota varsa atamayı yap.

                quota           = sch_info.loc["Quota", std_pref]
                numberOfStudent = len(sch_info.loc[m_round, std_pref])

                if quota > numberOfStudent:

                    current_pref_index = pd.Index(std_pref_table[student]).get_loc(std
                    assig_std_inf = {student: [std_pref,current_pref_index ]}
                    sch_info.loc[m_round,std_pref ].append(assig_std_inf)
                    assigment_list.append(assig_std_inf)
                    break

    # İlk tur atamaları yapıldıktan sonra !!!! tüm öğrenciler için
    # atandıkları okul sonrasındaki tercihler C0 yapıldı.

    for ass in assigment_list:

        student1           = list(ass.keys())[0]
        preference1        = list(ass.values())[0][0]
        preference_order1  = list(ass.values())[0][1]

        for col in std_pref_table.columns:

            if col == student1:

                for row in std_pref_table.index:

                    preference2 = std_pref_table.loc[row, col]
```

```
                    preference_order2 = row

                    if preference_order1 < preference_order2:
                        std_pref_table.loc[row, col] = "C0"


    # Bir önceki atamalar ile bu turdaki atamaları karşılaştır.
    # Aynı atamalar yapıldı ise son yapılam atamaları sil ve döngüyü durdur.

    if m_round > 1:

        previous_assigments = list(sch_info.loc[m_round - 1])
        current_assigments  = list(sch_info.loc[m_round])

        if previous_assigments == current_assigments:
            sch_info = sch_info[:-1]

            #SCH INFO DAHA SADE BİR ŞEKİLDE SCH INFO UPDATE TABLOSUNA DÖNÜŞTÜRÜLÜYOR.
            sch_info_update = copy.deepcopy(sch_info)
            for col in sch_info_update.columns:
                for row in sch_info_update.index[2:]:
                    sch_info_update.loc[row, col] = []

            for col in sch_info.columns:
                for row in sch_info.index[2:]:
                    ass = sch_info.loc[row, col]
                    if len(ass) > 0:

                        for x in ass:
                            student = list(x.keys())[0]
                            order   = list(x.values())[0][1]

                            value = (student, order)

                            sch_info_update.loc[row, col].append(value)

            break


    m_round += 1

sch_info
```

# ÖZGEÇMİŞ

**Kişisel Bilgiler**

Adı Soyadı : Uğur ÇUHALILAR

**Öğrenim Bilgileri**

Lisans Eğitimi : Kocaeli Üniversitesi, İktisadi ve İdari Bilimler Fakültesi, İktisat Programı

**Görevler** : Araştırma Görevlisi

Görev Yeri : Kocaeli Üniversitesi, İktisadi ve İdari Bilimler Fakültesi, İktisat Bölümü, İktisat Teorisi Anabilim Dalı