

Eleme Turnuvalarında Seribaşı Eşlemelerinin Sayılması ve Sayımlanması

Counting and Enumerating Seedings in Knockout Tournaments

Can ATILGAN
Dokuz Eylül Üniversitesi,
Bilgisayar Bilimleri Bölümü, İzmir, Türkiye
can.atilgan@deu.edu.tr
ORCID: 0000-0002-1680-6207

Mehmet Emin DALKILIÇ
Ege Üniversitesi,
Uluslararası Bilgisayar Enstitüsü, İzmir, Türkiye
mehmet.emin.dalkilic@ege.edu.tr
ORCID: 0000-0003-3932-5155

Öz

Eleme turnuvaları, yarışma tasarımlarının muhtemelen en yaygın olarak kullanılanıdır. Bu turnuvalarda, her turda tüm yarışmacılar bir eleme karşılaşması oynamak üzere eşlenirler ve kaybeden turnuva dışında kalır. Seribaşı eşlemesi olarak adlandırılan ilk tur maçlarının seçimi, turnuvanın geri kalanını da doğrudan etkiler. Seribaşı eşlemelerini çeşitli açılardan değerlendiren ve inceleyen dikkate değer sayıda çalışma olmasına karşın, bildiğimiz kadarıyla seribaşı eşlemelerini sayımlayan bir algoritma sunulmamıştır. Ayrıca, literatürde farklı seribaşı eşlemelerinin sayısının bulunması da yalnızca kısaca tartışılmıştır. Bu makalede farklı seribaşı eşlemelerinin nasıl sayılacağı tartışılmış ve eleme turnuvalarında seribaşı eşlemelerinin sayımlanması için etkin bir algoritma önerilmiştir.

Anahtar Sözcükler: Hesaplamalı Turnuva Tasarımı, Eleme Turnuvası, Seribaşı Eşleme, Sayma, Sayımlama

Abstract

Knockout tournaments are probably the most common of all competition designs. In those tournaments, all competitors are paired to play an elimination match at each round, where the loser is

Gönderme ve kabul tarihi: 14.11.2021 – 17.12.2021

Makale türü: Araştırma

immediately knocked out of the tournament. The setting of the first round of matches, named as seeding, directly influences the rest of the tournament. While there is a considerable amount of work that evaluate and examine different seedings from various perspectives, an algorithm that enumerates seedings has not been given to the best of our knowledge. Also, counting the number of unique seedings has been only briefly discussed in the literature. This paper discusses how to count the number of unique seedings and proposes an efficient algorithm for enumerating seedings in a knockout tournament.

Keywords: Computational Tournament Design, Knockout Tournament, Seeding, Counting, Enumeration

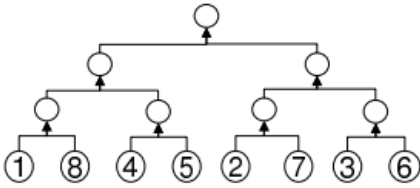
1. Giriş

Eleme turnuvaları, hemen hemen tüm spor yarışmaları için kullanılan, basit ve iyi bilinen yarışma şemalarıdır. Bir yarışmacının kaç kez kaybettikten sonra eleneceğine göre farklı eleme turnuvaları düzenlenebilir. Öte yandan, açık ara en yaygın eleme turnuvaları, tek eleme turnuvalarıdır. Tek eleme turnuvalarında yarışmacılar seribaşı eşlemesi olarak adlandırılan ilk tur eşlemelerine göre rakipleri belirlenerek turnuvaya başlarlar. Bir karşılaşmanın kazananı sonraki tura yükselirken, kaybeden yarışma dışı kalır ve tüm maçlarını kazanan yarışmacı, turnuvanın galibi olur. Bu makale boyunca turnuva dendiğinde kastedilen, tek

eleme turnuvasıdır.

Eleme turnuvası şemaları doğal olarak bir ikili ağaç ile temsil edilebilirler. Yarışmacı sayısı 2^n 'nin bir tam kuvveti olmak zorunda değilse de, çoğu turnuvada dengeli bir ağaç oluşturulabilmek için yarışmacı sayısı böyle seçilir. Yarışmacılar, uygulamada ve literatürde tipik olarak 1 ile 2^n arasındaki tamsayılarla numaralandırılırlar. Bu numaralandırma da, genellikle, turnuva öncesi bir değerlendirmeye göre belirlenir. Bu değerlendirme, basketbolda NBA final turnuvasına katılacak takımların uzun bir normal sezon sonrası belirlenmesi gibi bir ön turnuva yöntemiyle yapılabileceği gibi, ATP Tenis Dünya Turu gibi bir veya daha fazla sezon boyunca geçmiş turnuva performanslarına göre de yapılabilir. Yarışmacı 1 bu değerlendirmeye göre en üst sıradadır ve en güçlü yarışmacı olarak kabul edilir. Yarışmacılar böylece 2^n 'e kadar, değerlendirme ölçütüne göre güçlüden güçsüze doğru sıralanırlar.

Turnuvanın ilk turundaki maçların hangi yarışmacıların arasında oynanacağı, yalnızca yarışmacıların yarısı bu turda eleneceği için değil, turnuvanın devamındaki olası karşılaşmaların da aslında ilk turda belirleniyor olması sebebiyle turnuva düzenleyiciler için önemli bir karardır. İşte bu ilk tur maçlarının belirlenmesi, yani yarışmacıların turnuva ağacının yapraklarına yerleştirilmesi, seribaşı eşlemesi olarak anılır. Şekil 1'deki turnuva ağacı, muhtemelen en çok kullanılan seribaşı eşlemesini göstermektedir. Bu seribaşı eşlemesinin neden tercih edildiği sorgulanamaz değildir. Muhtemelen, yüksek sıralamaya sahip yarışmacıların daha geç karşılaşmasını sağlıyor olması bu seribaşı eşlemesini popüler yapmaktadır.



Şekil-1: Yaygın kullanılan bir seribaşı eşlemesi

Seribaşı eşlemesi, turnuvanın gidişatını tayin etmesi nedeniyle turnuvanın adaleti, seyirci ve yarışmacı memnuniyeti gibi ana özelliklerini de etkiler. Araştırmacılar, seribaşı eşlemesinin bir turnuvayı nasıl etkilediğini çeşitli yaklaşımlarla ele almışlardır.

Öncül sayılabilecek çalışmalarda, bazı niceliksel koşulları sağlayan seribaşı eşlemeleri aranmış ve az sayıda yarışmacısı olan turnuvalar için bu şartları sağlayan bazı seribaşı eşlemeleri belirlenmiştir [1–4]. Takip eden çalışmalardan bir kısmında seribaşı eşlemelerinin etkileri gerçek turnuva örnekleri üzerinde matematiksel olarak incelenirken [5–9], diğer bir kısmında da bir seribaşı eşlemesinin neden ve nasıl diğerlerinden daha iyi olduğunu belirten aksiyomatik kriterler ortaya konmuş, bu kriterleri sağlayan seribaşı eşlemeleri önerilmiştir [10–17]. Ancak, bildiğimiz kadarıyla, kombinatorik bir problem olan seribaşı eşlemelerinin sayımlanması için bir algoritma sunulmamıştır. Bu makalede önce farklı (biricik) seribaşı eşlemelerinin nasıl sayılacağı tartışılmış, bilinen kapalı sayım formülüne eşit olduğu gösterilerek bir özyinelemeli formül sunulmuştur. Devamında, bu özyinelemeden de yararlanılarak seribaşı eşlemelerini sayımlamak için etkin bir algoritma önerilmiştir.

2. Temel Tanım ve İfadeler

Turnuvadaki tur sayısı k ve yarışmacı sayısı $n = 2^k$ olmak üzere yarışmacılar $C = \{1, 2, 3, \dots, 2^k\}$ kümesi ile ifade edilir. En güçlü yarışmacı 1 olmak üzere daha güçlü yarışmacılar, daha küçük pozitif tamsayılarla temsil edilirler.

Bir seribaşı eşlemesi, i . elemanı turnuva ağacının i . yaprağına yerleştirilmek üzere bir S sıralı kümesi ile ifade edilir. Örneğin, Şekil 1'deki seribaşı eşlemesi $S = (1, 8, 4, 5, 2, 7, 3, 6)$ ile ifade edilir.

Bir maç, iki yarışmacının birbirine eşleştiği ve sonucunda birinin galip diğerinin de mağlup olarak belirleneceği oyun veya oyunlar dizisidir. $c_1, c_2 \in C$ olmak üzere bir maç $m = \{c_1, c_2\}$ ile ifade edilir. Örneğin, Şekil 1'deki ağacın en solundaki iki yaprağın belirttiği maç $m = \{1, 8\}$ ile gösterilir. Bu kümenin sıralı olmamasına karşın, gösterim uyumluluğu ve kolaylığı için küçük numaralı yarışmacı sola yazılır.

Bir maç içerisindeki sıralama önemsiz olduğu gibi, bir maçın bulunduğu tur içerisindeki sıralama da, yalnızca o tur düşünüldüğünde önemsizdir. Ancak bir maçın tur içindeki sırası, sonraki turlardaki muhtemel eşlemeleri tayin etmektedir. Dolayısıyla, $k = 3$ için bir turnuvanın seribaşı eşlemesinin $S = (1, 8, 4, 5, 2, 7, 3, 6)$ olduğunun ifadesi ile ilk turunun $t_1 = (\{1, 8\}, \{4, 5\}, \{2, 7\}, \{3, 6\})$ olduğunun ifadesi birbirine denktir. Öte yandan, $M = \{\{1, 8\}, \{4, 5\}, \{2, 7\}, \{3, 6\}\}$ ile gösterilen maçlar kümesi, bu iki ifadeyle gösterilen turnuvayı temsil

etmez; yalnızca ilk turda hangi maçların oynanacağını gösterir.

3. Farklı Seribaşı Eşlemelerini Sayma

Tur sayısı k olan bir tek eleme turnuvasının farklı seribaşı eşlemelerinin sayısı Groh ve ark. [11] ve Karpov [13] aşağıdaki kapalı formül ile vermişlerdir.

$$NS(k) = \frac{2^k!}{2^{2^k-1}} \quad (1)$$

Bu formülün farklı seribaşı eşlemelerini nasıl saydığına dair bir tartışmaya veya bu konuda bir referansa ise rastlanmamıştır. Formül (1)'in de ima ettiği gibi, yarışmacı numaralarının tüm farklı sıralamalarının farklı bir seribaşı eşlemesine karşılık gelmediği dikkate değerdir.

$k = 1$ için, turnuvada sadece bir maç vardır ve bu maçtaki yarışmacı sırası önemsiz olduğu için $NS(1) = 1$ 'dir.

$k = 2$ için, ilk turda iki maç oynanır. Tek maçın oynanacağı ikinci tur final maçıdır ve ilk turdaki maçların sıralamasından etkilenmez. Örneğin, $(1,2,3,4)$ seribaşı eşlemesi şu iki kümeyle de gösterilebilir: $(\{1,2\}, \{3,4\})$ veya $(\{3,4\}, \{1,2\})$. Sonuçta final, $\{1,2\}$ maçının kazananı ile $\{3,4\}$ maçının kazananı arasında oynanır. Fakat $\{1,2\}, \{3,4\}$ kümesi ile gösterilebilecek ilk tur maçları kümesi, 4 yarışmacı için tüm olası maçları içermez.

n yarışmacı için farklı maç kümelerinin sayısı

$$NM(n) = (n-1)!! = \prod_{i=1}^{\frac{n}{2}} (2i-1) \quad (2)$$

ile hesaplanabilir. Buradaki çift faktöriyel gösterimi, formülün en sağındaki çarpımdan anlaşılacağı üzere $n-1$ 'e kadar olan pozitif tek sayıların çarpımını ifade etmektedir. Aslında n farklı cismin $n/2$ adet çifte kaç farklı şekilde ayrılacağına ilişkin sayısını veren bu formül hakkında, Sloane [18] tarafından oluşturulan çevrimiçi ansiklopedide pek çok ilginç yorum ve uygulama örneği bulunabilir. Sonuçta, 4 yarışmacı için $(\{1,2\}, \{3,4\})$, $(\{1,3\}, \{2,4\})$ ve $(\{1,4\}, \{2,3\})$ şeklinde 3 farklı seribaşı eşlemesi vardır.

$k = 3$ için 8 yarışmacı ve $NM(8) = 105$ tane farklı maç kümesi vardır. Bu kez ilk turdaki maçların sırası, kendi de 4 yarışmacılı bir turnuva olan ikinci turu etkiler. İkinci turda ise durum, $k = 2$ durumu ile aynıdır ve bu tur üç farklı şekilde oynanabilir. Yani, a, b, c ve d ilk tur maçlarının galipeleri ise

$(\{a, b\}, \{c, d\})$, $(\{a, c\}, \{b, d\})$, $(\{a, d\}, \{b, c\})$

turlarından herhangi biri olasıdır. Dolayısıyla, ilk turdaki maçlar nasıl sıralanırsa sıralansın, ancak üç farklı (biricik) turnuva senaryosu üretilebilir. 105 farklı maç kümesinin her biri için 3 farklı ikinci tur senaryosu olduğuna göre, 8 yarışmacı için $105 \cdot 3 = 315$ farklı seribaşı eşlemesi vardır.

Bu gözlem $k \geq 2$ genel durumu için de, tabii ki geçerlidir. Bir turdaki maçların farklı sıralanışlarının sayısı, kendinden sonraki turdaki farklı eşlemelerin sayısına, yani (2) formülü ile verilen farklı maç kümelerinin sayısına eşittir. Bu sıralama her farklı maç kümesi için tekrar ettiğine göre, k turu olan bir eleme turnuvası için farklı seribaşı eşlemeleri aşağıdaki özylenelele ile sayılabilir.

$$NS(k) = \begin{cases} 1, & k = 1 \\ NM(2^k) \cdot NS(k-1), & k \geq 2 \end{cases} \quad (3)$$

Bu özylenelemenin kapalı formülü aşağıda verilmiştir.

$$NS(k) = \prod_{i=1}^k NM(2^i) = \prod_{i=1}^k (2^i - 1)!! \quad (4)$$

Son olarak, (4) formülü ile literatürde verilen (1) formülünün eşit oldukları aşağıda gösterilmiştir.

$$\begin{aligned} \prod_{i=1}^k (2^i - 1)!! &= \prod_{i=1}^{k-1} (2^i - 1)!! \cdot \prod_{i=1}^{2^{k-1}} (2i - 1) \\ &= \prod_{i=1}^{k-1} (2^i - 1)!! \cdot \frac{2^k!}{\prod_{i=1}^{2^{k-1}} (2i)} \\ &= \prod_{i=1}^{k-1} (2^i - 1)!! \cdot \frac{2^k!}{2^{k-1}! \cdot 2^{2^{k-1}}} \\ &= \prod_{i=1}^{k-2} (2^i - 1)!! \cdot \frac{2^k!}{2^{k-1}! \cdot 2^{2^{k-1}}} \cdot \frac{2^{k-1}!}{2^{k-2}! \cdot 2^{2^{k-2}}} \\ &= \prod_{i=1}^{k-3} (2^i - 1)!! \cdot \frac{2^k!}{2^{k-1}! \cdot 2^{2^{k-1}}} \cdot \frac{2^{k-1}!}{2^{k-2}! \cdot 2^{2^{k-2}}} \\ &\quad \cdot \frac{2^{k-1}!}{2^{k-3}! \cdot 2^{2^{k-3}}} \\ &= \dots \\ &= \prod_{i=1}^1 (2^i - 1)!! \cdot \frac{2^k!}{2^1! \cdot \prod_{j=1}^{k-1} (2^j)} = \frac{2^k!}{2^{\sum_{j=0}^{k-1} (2^j)}} \\ &= \frac{2^k!}{2^{2^k-1}} \blacksquare \end{aligned}$$

Farklı seribaşı eşlemeleri, yalnız bir formül olan (1) ile doğrudan kolayca sayılabilir olsa da, bu eşlemeleri sayımlamak görece zahmetli ve incelikli bir iştir. (3) ile vermiş olduğumuz özylenelele, bir

sonraki bölümde önerilen sayımlama algoritmasına temel oluşturma niteliğindedir.

4. Seribaşı Eşlemelerini Sayımlama

Turnuvarın ilk turu için seçilebilecek her farklı maç kümesi, maçların turnuva ağacındaki yerlerinden, yani maç kümesindeki sıralamalarından bağımsız olarak farklı bir turnuvayı temsil eder. Ne de olsa iki turnuva arasındaki tek bir maç bile farklı olsa, onların farklı şekilde gerçekleşmiş turnuvalar olduğu söylenebilir. Bu nedenle, seribaşı eşlemelerini sayımlayabilmek için önce ilk turdaki tüm farklı maç kümelerinin oluşturulması gerekir.

4.1 Tüm Farklı Maç Kümelerini Üreten Algoritma

Tüm farklı maç kümelerini üretmek için Yarışmacı 1'i bir maça atayarak başlayalım. Yarışmacı 1'in yer alabileceği

$n - 1$ farklı maçın her biri için, n maça Yarışmacı 1'in rakibi olmamış en küçük numaralı yarışmacıyı sonraki maça atayalım. Bu yeni maça atanan yarışmacı da $n - 3$ farklı yarışmacıdan biriyle karşılaşabilir. Buna göre ilk iki maç $(n - 1) \cdot (n - 3)$ farklı şekilde oluşturulabilir. Bu örüntü takip edilerek tüm maçlar tamamlandığında (2) numaralı formüle ulaşılır. Aşağıda kaba kodu verilen Algoritma 1, bu prosedürü takip ederek 1'den n 'e kadar numaralandırılmış yarışmacılar için tüm farklı maç kümelerinin bir koleksiyonunu üretmektedir.

Algoritma 1

Girdi: Yarışmacı sayısı n

1. $M' \leftarrow \emptyset$; // boş bir maç kümeleri kümesi (koleksiyonu)
2. **call** Maçlar(\emptyset);
3. **output** M' ; // Maçlar prosedürü tarafından dolduruldu
4. **end.**

procedure Maçlar (M)

- p1. **if** $|M|=n/2$ **then** // mevcut maç kümesi tamamlandıysa
- p2. $M' \leftarrow M' \cup \{M\}$; // maç kümeleri koleksiyonuna ekle
- else**
- p3. Yeni bir maç oluştur $m: \{m.sol, m.sağ\}$;
- p4. $m.sol \leftarrow \min\{i \in \mathbb{Z}^+ | \forall m \in M, i \notin m\}$;

p5. **for** $i \leftarrow m.sol + 1$ **to** n **do**

p6. **if** $\forall m \in M, i \notin m$ **then**

p7. $m.sağ \leftarrow i$;

p8. Yeni bir maç kümesi oluştur
 $M_1 \leftarrow M \cup \{m\}$;

p9. **call** Maçlar(M_1);

end if

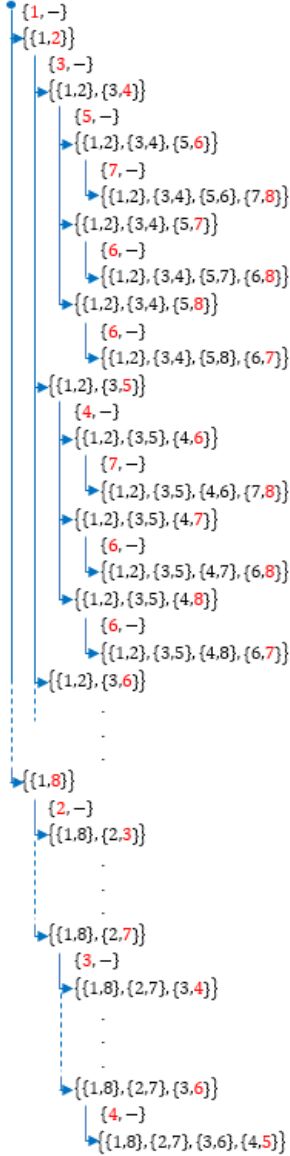
end for

end if

end procedure

Maçlar prosedürüne yapılan her bir çağrıda, bir maçlar kümesine yeni bir maç (yarışmacı çifti) eklenir. Bir maç kümesinde $\frac{n}{2}$ maç olduğuna göre, tek bir maç kümesi tamamlanmaya kadar $\frac{n}{2}$ çağrı yapılır. Farklı maçların sayısı $(n - 1)!!$ olduğuna göre, toplam yapılan prosedür çağrısı sayısı $\left(\frac{n}{2}\right) (n - 1)!!$ 'dir. Tek bir çağrı ise, p4 adımındaki $\min\{i \in \mathbb{Z}^+ | \forall m \in M, i \notin m\}$ ifadesi ve p8 adımındaki $M_1 \leftarrow M \cup \{m\}$ ifadesinin her ikisinde de yarışmacıların en fazla birer kez işleniyor olması sebebiyle $O(n)$ zaman alır. İlk bakışta çağrı içerisindeki "for" döngüsü bu karmaşıklık sınıfına ters görünüyorsa da bu döngü içerisinde yapılan çağrılar, yukarıda toplam çağrı sayısı belirlenirken hesaba katılmıştır. Sonuçta, Algoritma 1'in toplam zaman karmaşıklığı $O(n^2 \cdot (n - 1)!!) \subseteq O(n!)$ 'dir.

Algoritma 1'in $n = 8$ için bir çalışma örneği Şekil 2'de görselleştirilmiştir. Şekildeki oklar özyineleme çağrılarını, girintiler ise özyinelemenin derinliğini göstermektedir.



Şekil-2: Algoritma 1'in çalışma örneği.

4.2 Seribaşı Eşlemelerini Sayımlama Algoritması

İlk turdaki maçların yerleşimi, daha önce de bahsedildiği gibi, sonraki turlardaki eşlemeleri doğrudan etkiler. Tur sayısı k olan bir turnuvanın ilk turunda oynanacak maçlar belirlenmiş ve $M = \{m_1, m_2, \dots, m_{n/2}\}$ kümesiyle temsil ediliyor olsun. Başlangıçta, maçların yerleşebileceği $\frac{n}{2} = 2^{k-1}$ adet boş pozisyonun –yani, bir çift boş yaprağın– olduğu bir turnuva ağacına M kümesinin elemanlarının tek tek yerleştirildiğini düşünelim. İlk maç m_1 herhangi bir pozisyona yerleştirilebilir. Ağacın en soluna yerleştirilelim. Bir m_i maçının galibi g_i olsun. O halde m_2 'nin yeri, g_1 ve g_2 yarışmacılarının karşılaşma ihtimali olan turu tayin eder. Eğer m_2 ağacın sağ yarısına, yani soldan $(2^{n-2} + 1)$. pozisyon ile (2^{n-1}) . pozisyon arasında, bu sınır pozisyonları da dahil olmak üzere yerleştirilirse, $\{g_1, g_2\}$ maçı finalden önce oynanamaz. Benzer şekilde, eğer m_2 ağacın sol yarısında bir pozisyona yerleştirilirse, $\{g_1, g_2\}$ maçı ancak finalden önceki bir turda oynanabilir. Ağacın bir yarısı da $k - 1$ turdan oluşan bir alt turnuvadır. m_2 'nin ağacın sol yarısına yerleştirildiği durumda g_1 ve g_2 yarışmacılarının hangi turda karşılaşacağını da m_2 'nin bu alt turnuva ağacındaki pozisyonu tayin eder. Bu durumda da yine, iki olası farklı yerleştirme yapılabilir: $\{g_1, g_2\}$ maçının $(k - 1)$. turda, yani yarı finalde oynanacağı şekilde ağacın sağ yarısına veya daha önceki turlardan birinde oynanacağı şekilde sol yarısına. Bu şekilde, $1 < k' < k$ için, k' tura sahip her alt turnuva için mevcut alt ağacın sol veya sağ yarısına olmak üzere m_2 yerleştirilmeye devam edilebilir ve her farklı durumda farklı bir seribaşı eşlemesi ortaya çıkar. Çünkü, her farklı durumda oynanacak turnuva da değişecektir.

m_2 'nin seribaşı eşlemesindeki yeri (veya sırası) belirlenirken, her adımda ele alınan turnuva ağacının sağ yarısındaki tüm pozisyonların boş olduğuna dikkat edilmelidir. Bu nedenle, maç ele alınan alt ağacın yalnızca soluna yerleşirken yeniden ikili ihtimaller doğmakta ve sağa doğru bir yinelemeye gerek kalmamaktadır. Genel durumda da, eğer eldeki alt turnuva ağacı tamamen boş ise, m_i maçı bu alt ağaçtaki herhangi bir pozisyona yerleştirilebilir. Aksi halde, m_i 'nin alt ağacın sağ yarısına ve sol yarısına yerleştirildiği iki durum birbirinden farklıdır: m_i sağ yarıya yerleştirilirse, sol yarıda yer alan yarışmacılardan biriyle ancak alt ağacın temsil

ettiği alt turnuvanın son turunda karşılaşılabirirken, sağ yarıdaki bir yarışmacıyla ancak son turdan önce karşılaşılabir. Ele alınan alt turnuvada boş bir pozisyon yoksa, m_i 'nin o pozisyonlardan birinde yer aldığı farklı bir seribaşı eşlemesi yok demektir. Sonuçta, M kümesindeki maçlar mümkün tüm pozisyonlara yerleştirildiğinde, farklı turnuvalar oluşturacak şekilde M 'nin elemanlarının tüm farklı dizilişleri bulunmuş olur. Algoritma 1'deki üretilen M' maç kümeleri koleksiyonundaki tüm M maç kümeleri için bu yerleştirme işlemi yapıldığında, seribaşı eşlemeleri de sayımlanmış olur.

Algoritma 2, yukarıda anlatılan süreci takip ederek k turlu bir eleme turnuvası için seribaşı eşlemelerini sayımlar. Algoritmada bir alt turnuva ağacı, seribaşı diye isimlendirilmiş bir maçlar dizisi, ve en sol ve en sağ pozisyonların indisleriyle temsil edilmiştir. Şimdiye kadarki gösterimle uyumlu şekilde düşük numaralı yarışmacılar daha önce işlenir ve uygun pozisyonların en soluna yerleştirilir. Bu şekilde yerleştirmenin bir faydası da bir alt turnuva ağacının tamamen boş olup olmadığını belirlemek için yalnızca en sol pozisyonun kontrol edilmesini yeterli hale getirmesidir.

Algoritma 2

Girdi: Tur sayısı k

1. $n = 2^k$ için Algoritma 1'i çalıştırarak M' kümesini üret;

2. $S' \leftarrow \emptyset$; // boş bir seribaşı eşlemeleri kümesi (koleksiyonu)

3. **call** Maçlar (\emptyset);

// $S[1..2^{n-1}]$: bir seribaşı eşlemesini temsil eden boş bir dizi

4. $\forall M \in M'$: **call** Seribaşları($S, M, 1, 2^{n-1}$);

5. **output** S' ; // Seribaşları prosedürü tarafından dolduruldu

6. **end**.

procedure Seribaşları (S, M, l, r)

p1. **if** $M = \emptyset$ **then** //mevcut maçların hepsi yerleştirildiyse

p2. $S' \leftarrow S' \cup \{S\}$; //seribaşı kümesini koleksiyona ekle

else

p3. **if** $S[l] = \text{boş}$ **then**

p4. Bir maç seç $m: \{m.\text{sol}, m.\text{sağ}\} \in M$,

öyle ki $\forall m_i \in M, m.\text{sol} \leq m_i.\text{sol}$;

p5. $S[l] \leftarrow m$;

p6. $M \leftarrow M \setminus \{m\}$;

p7. **call** Seribaşları($S, M, 1, 2^{n-1}$);

else

p8. **if** $r - l > 0$ **then**

p9. **call** Seribaşları ($S, M, l, l + \lfloor (r - l)/2 \rfloor$);

p10. **call** Seribaşları ($S, M, l + \lfloor (r - l)/2 \rfloor + 1, r$);

end if

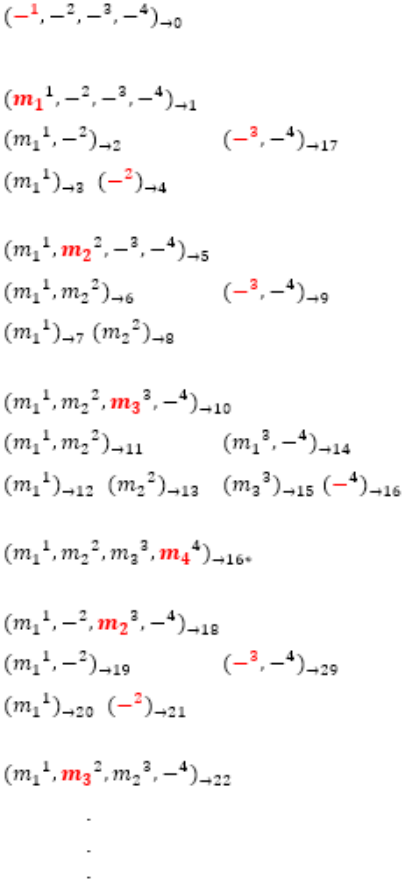
end if

end if

end procedure

Seribaşları prosedürü, başlangıçta boş olan seribaşı dizisine özyinelemeli çağrılarla $\frac{n}{2}$ adet maç ekleyerek bir tam seribaşı eşlemesi (sıralı maçlar kümesi) oluşturur. Tüm dizi boşken bir maçı yerleştirmek için tek bir prosedür çağrısı yeterli olur. En kötü durumda ise, ağacın en sağındaki (dizide en büyük indisli) maç hariç diğer maçlar diziyeye halihazırda eklenmiştir. Her çağrıda dizi boyutu yarılandığı, diğer bir deyişle ağacın bir alt dalına inildiği için, bu en kötü durumda tüm ağaç dolaşılarak en sağdaki pozisyona erişilinceye kadar maç diziyeye eklenemeyecektir. O halde, tüm turnuva ağacının dolaşılması için en çok $2^k - 1 = n - 1$ çağrı yapılması gerekir. Toplamda, tek bir farklı seribaşı eşlemesinin oluşturulması için de en fazla $\left(\frac{n}{2}\right)(n - 1) \in O(2^{2k})$ çağrı yapılması gerekir. Bir çağrıda seribaşı dizisine eklenecek maçın p4 adımındaki $\forall m_i \in M, m.\text{sol} \leq m_i.\text{sol}$ koşulunu sağlayacak şekilde seçilmesi işlemi, Maçlar prosedürünün maçları küçük numaralı yarışmacılar solda olacak şekilde üretmesinden faydalanılarak, son eklenen maçın pozisyonunu kaydetmek suretiyle sabit zaman alacak şekilde uygulanabilir. Sonuçta, toplam $NS(k)$ tane farklı seribaşı eşlemesi olduğu ve Seribaşları prosedürünün ilk çağrısından önce $O(n!)$ zaman alan Algoritma 1'in çalıştırılması gerektiği için, Algoritma 2'nin zaman karmaşıklığı $O(n! + 2^{2k} \cdot NS(k)) \equiv O(2^{k! + 2^{2k}} \cdot \frac{2^{k!}}{2^{2^{k-1}}}) \equiv O(2^{k!}) \equiv O(n!)$ sınıfındadır.

Algoritma 2’de verilen Seribaşları prosedürünün boş bir S ve $M = \{m_1, m_2, m_3, m_4\}$ için bir Seribaşları($S, M, 1, 4$) çağrısı, Şekil 3’te görselleştirilmiştir. Parantez içerisinde gösterilen, çağrıdaki seribaşı dizilerinin sağ altındaki sayı özyineleme çağrısının sırasını göstermektedir. Yerleşen maçların ve boş elemanların sağ üstündeki sayı ise, seribaşı dizisindeki pozisyonu belirtmektedir.



Şekil-3: Algoritma 2’de verilen Seribaşları prosedürüne yapılan bir çağrının çalışma örneği

5. Algoritmaların Uygulanması

Önceki bölümde önerilen algoritmaların sunumları, uygulama geliştirmeye kolaylık sağlamalarından ziyade yalın bir bilimsel dile sahip ve anlaşılabilir olmaları öncelenecek yapılmıştır. Gerçekten bir eleme turnuvası için seribaşlarını sayımlayacak bir

bilgisayar programı geliştirmek için, verilen algoritmalar üzerinde bazı mütevazı değişiklikler yapmak faydalı olacaktır. Diğer bir önemli nokta ise, yarışmacı sayısı ve dolayısıyla tur sayısı arttıkça, farklı seribaşı eşlemelerinin sayısının da aşırı derecede büyüyor olmasıdır. Pratikte, günümüzde ticari olarak satılan sıradan bir bilgisayarla 1 saat içinde ancak 16 yarışmacıya kadar turnuvalar için seribaşı eşmeleri sayımlanabilir. Yine de, daha çok yarışmacıya sahip turnuvalar için de sayımlama kullanışlı olabilir. Aşağıda, algoritmaların uygulamasına yönelik bazı küçük değişiklik önerileri ve kullanım senaryoları tartışılmaktadır.

Algoritma 2’de, olası farklı maç kümelerinin koleksiyonu M' ilk adımda Algoritma 1 kullanılarak elde edilmekte ve Seribaşları prosedürünün son çağrısına kadar sürekli olarak bu koleksiyon kullanılmaktadır. Bu haliyle uygulandığında, koleksiyonun program boyunca bellekte tutulması gerekir, ki bu da, formül (2)’de verilen farklı maç kümelerinin sayısı olan $\Theta((n-1)!!)$ alan karmaşıklığını işaret eder. Algoritma 1’in sonuna kadar çalışarak tüm koleksiyonu vermesini beklemek yerine, koleksiyona yeni bir maç kümesi eklendiğinde Seribaşları prosedürü o küme için çağrılabilir. Bu durumda, iki algoritmadaki özyinelemelerin aynı anda yığın belleği (ing. stack) kullanması dışında ilave bir bellek gereksinimi olmayacak, S dizisinin yanında yalnızca n yarışmacı içeren bir maç kümesi için bellekte yer ayırmak yeterli olacaktır. Hem Maçlar, hem de Seribaşları prosedürleri derinlik öncelikli çalıştıklarından, özyineleme derinliği turnuva ağacının derinliği ile sıkı olarak sınırlıdır. Yani, özyineleme için kullanılacak alanın karmaşıklığı $\Theta(\log N)$ sınıfındadır.

Algoritma 2 paralel programlamaya da oldukça uygundur. Seribaşları prosedürüne farklı M kümeleri için yapılan çağrılardaki hesaplamalar tamamen birbirlerinden bağımsızdır. Çağrıya gönderilen parametreler ve üretilecek seribaşı eşlemelerinin yazılacağı yer için farklı bellek bölgeleri ayrıldığı sürece, birden çok işlemci üzerinde doğrudan paralel olarak çalıştırılabilirler.

Bir başka uygulama yaklaşımı, Seribaşları prosedürünün üretilen tüm farklı maç kümeleriyle çağrılması yerine, yalnızca bir defa “maç temsilcileriyle” çağrılması olabilir. Bu temsilciler m_i gibi sembolik olarak seçilirse, maçların ilk tur

için tüm farklı dizilişlerini gösteren bir soyut şema elde edilecektir. Daha sonra belirlenecek herhangi bir maç kümesi, bu şema üzerine yerleştirilerek olası tüm seribaşı eşlemeleri görülebilir. Maç temsilcisi olarak, değerleri daha sonra doldurulmak üzere ayrılmış bellek adreslerini gösteren işaretçiler de seçilebilir.

Yarışmacı sayısı arttıkça farklı seribaşı eşlemelerinin sayısı üzerinde çalışılmayacak kadar büyüdüğü için, 16'dan daha fazla yarışmacısı olan turnuvaların düzenleyicileri, ancak yarışmacıların bazı alt kümelerini seribaşı seçebilir veya turnuvayı her biri daha küçük alt turnuvalar içeren birkaç aşama şeklinde düzenleyebilirler. Örneğin, 64 yarışmacısı olan bir turnuva, yalnızca 16 yarışmacıyı seribaşı seçerek diğerlerini rastgele olarak turnuva yapraklarına yerleştirebilir. Seribaşı seçilen 16 yarışmacı, birbirleriyle ilk iki turda karşılaşmayacak şekilde yerleştirilirse ve bu turlarda kazanacakları kabul edilirse, bu iki turun ardından oynanacak turnuvanın seribaşı eşlemeleri sayımlanabilir boyuttadır.

Diğer bir kullanım seçeneği turnuvayı iki aşamada tasarlamak olabilir. İlk aşamada, yarışmacılar belirli kriterlere göre, 8 yarışmacılı 8 farklı gruba ayrılır ve her biri ayrı bir alt turnuva oynar. Devamında, alt turnuvanın kazananları ikinci aşamada yine bir 8 yarışmacılı turnuva oynarlar.

Son olarak, önceki seribaşı çalışmalarına daya uygun bir kullanım senaryosu da sayımlama uzayının belirli kriterleri sağlayacak şekilde daraltılarak makul sürelerde kısmi sayımlama yapılmasıdır.

6. Sonuç

Bu makalede, bir tek eleme turnuvasında farklı seribaşı eşlemelerinin nasıl sayılacağı üzerine bir tartışma verilmiş ve bu seribaşı eşlemelerinin sayımlanması problemi ele alınmıştır. Sayma üzerine verilen tartışma, farklı seribaşı eşlemelerinin sayısını veren bir özinelemeli formülü, bu formülün kapalı formunu ve bu özinelemenin literatürde bahsedilmiş olan sayma formülüne denk olduğunun ispatını içermektedir. Seribaşı eşlemelerinin sayımlanması problemi 2 alt probleme bölünmüştür: tüm farklı ilk tur maçlarının kümelerinin üretilmesi ve her bir kümedeki maçların başlangıçta boş olan bir turnuva ağacına her biri farklı bir turnuvaya karşılık gelecek dizilişlerle yerleştirilmesi. Her iki alt problem için birer algoritma önerilmiştir. Bu

algoritmaların teorik performans analizleri, uygulama için önerileri ve kullanım senaryoları verilmiştir.

Yapılan tartışmalar, verilen formüller ve önerilen algoritmaların, gelecekte eleme turnuvaları ve seribaşı eşlemeleri üzerine yapılacak araştırmaları kolaylaştıracağına inanıyoruz. Sunulan bilgiler, araştırmacıların yanında turnuva düzenleyicilerinin de bu konuda strateji ve uygulama geliştirmelerine doğrudan veya dolaylı fayda sağlayabilirler. Son olarak, okuyucuların turnuvaların hesaplamalı ve kombinasyonel bakış açısıyla tasarlanması konusunda ilgisinin tetikleneceğini umuyoruz.

7. Bilgilendirme

Bu makale, 5. Uluslararası Bilgisayar Bilimleri ve Mühendisliği Konferansında (UBMK 2020) sunulan [19] numaları çalışmanın genişletilmiş ve geliştirilmiş halidir.

Kaynakça

- [1] F. K. Hwang, New concepts in seeding knockout tournaments, *The American Mathematical Monthly*, vol. 89, no. 4, pp. 235–239, 1982.
- [2] J. Horen, R. Riezman, 1985, Comparing draws for single elimination tournaments, *Operations Research*, vol. 33, no. 2, pp. 249–262, 1985.
- [3] D. R. Appleton, May the best man win?, *The Statistician*, vol. 44, no. 4, pp. 529–538, 1995.
- [4] A. J. Schwenk, What is the correct way to seed a knockout tournament?, *The American Mathematical Monthly*, vol. 107, no. 2, pp. 140–150, 2000.
- [5] J. Phillips, S. B. Caudill, F. G. Mixon Jr., Tournament Seeding Efficiency and Home Court Advantage: College Basketball's National Invitation Tournament, *International Journal of Statistics and Probability*, vol. 4, no. 3, 2015.
- [6] D. Dagaev, R. Vladimir, Seeding the UEFA Champions League Participants: Evaluation of the Reform, *Higher School of Economics Research Paper No. WP BRP 129/EC/2016*, 2016.
- [7] L. Csató, The UEFA Champions League seeding is not strategy-proof since the 2015/16 season, *Annals of Operations Research* vol. 292, pp. 161–169, 2020.
- [8] L. Csató, Optimal Tournament Design: Lessons From the Men's Handball Champions League, *Journal of Sports Economics*, doi: 10.1177/1527002520944442, 2020.
- [9] F. Della Groce, G. Dragotto, R. Scatamacchia, On fairness and diversification in WTA and ATP tennis tournaments generation, *Annals of Operations Research*, doi: 10.1007/s10479-020-03517-8, 2020.

- [10] T. Vu, Y. Shoham, "Fair seeding in knockout tournaments", vol. 3, no. 1, Article no. 9, 2011.
- [11] C. Groh, B. Moldovanu, A. Sela, U. Sunde, Optimal seedings in elimination tournaments, *Economic Theory*, vol. 49, pp. 59–80, 2012.
- [12] M. Kräkel, Optimal seedings in elimination tournaments revisited, *Economic Theory Bulletin*, vol. 2, no. 1, pp. 77–91, 2014.
- [13] A. Karpov, A new knockout tournament seeding method and its axiomatic justification, *Operations Research Letters*, vol. 44, no. 6, pp. 706–711, 2016.
- [14] D. Dagaev, A. Suzdaltsev, Competitive intensity and quality maximizing seedings in knock-out tournaments, *Journal of Combinatorial Optimization*, 35, pp. 170–188, 2018.
- [15] H. Aziz, S. Gaspers, S. Mackenzie, N. Mattei, P. Stursberg, T. Walsh, Fixing balanced knockout and double elimination tournaments, *Artificial Intelligence*, vol. 262, pp. 1–14, 2018.
- [16] R. Arlegi, D. Dimitrov, Fair elimination-type competitions, *European Journal of Operational Research*, vol. 287, no. 2, pp. 528–535, 2020.
- [17] R. Arlegi, How can an elimination tournament favor a weaker player?, *International Transactions in Operational Research*, doi: 10.1111/itor.12955, 2021.
- [18] N. J. A. Sloane, Sequence: A001147, The on-line encyclopedia of integer sequences, <http://oeis.org>, son erişim: 09.11.2021.
- [19] C. Atılğan, M. E. Dalkılıç, Enumerating Possible Seedings in a Knockout Tournament, 2020 5th International Conference on Computer Science and Engineering (UBMK), Diyarbakir, Turkey, Sept. 2020, doi: 10.1109/UBMK50275.2020.9219510.