



AN EFFICIENT SUDOKU SOLVER APPLICATION BASED ON GRAPH THEORY

Mustafa Batar*¹ 

¹Burdur Mehmet Akif Ersoy University, Faculty of Engineering and Architecture, Department of Computer Engineering, 15030 Burdur, Turkey

Abstract

Original scientific paper

This article has explained in detail what the puzzle of Sudoku is – the meaning of Sudoku –, where it comes from – the origin of Sudoku – and how it can be solved – the solution way of Sudoku. Especially, this paper has analyzed the solution of the problem – Sudoku can be taken as a problem – based on the graph theory. This theory consists of several algorithms, methods, rules and principles which are about the graph in general. In addition, this study has focused on the Welsh-Powell algorithm – greedy coloring algorithm – and Karger’s algorithm – contraction algorithm – among those graph algorithms by trying to give explanations about them (these two methods). Also, based on the rules and principles of those two algorithms, “Sudoku Solver Application” has been designed and developed in order to solve the puzzle of Sudoku. Moreover, the paper has presented a specific solution way of the puzzle Sudoku, and has showed its efficiency and usability by showing complexity and run time of the application among several available solutions based on the graph theory. So, the study has offered “Sudoku Solver Application” to the game world, science world, and education world as well for its use.

Keywords: Graph theory, sudoku, Welsh-Powell algorithm, Karger’s algorithm.

GRAFİK TEORİSİNE DAYALI ETKİN BİR SUDOKU ÇÖZÜCÜ UYGULAMASI

Özet

Orijinal bilimsel makale

Bu makale, Sudoku bulmacasının ne olduğunu (anlamını), nereden geldiğini (kökenini) ve nasıl çözülebileceğini (çözüm yolunu) açıklamaktadır. Özellikle, problemin çözümünü – Sudoku bulmacası bir problem olarak ele alınabilir – grafik teorisine dayalı olarak analiz etmektedir. Bu teori, genel olarak grafiklerle ilgili çeşitli algoritmalar, yöntemler, kurallar ve ilkelerden oluşmaktadır. Ayrıca, grafik algoritmalarından Welsh-Powell (açgözlü renklendirme algoritması) ve Karger (daraltma algoritması) algoritmaları üzerinde durularak bu iki yöntem hakkında bu çalışmada detaylı bir bilgilendirme yapılmıştır. Bununla birlikte, bu iki algoritmanın kural ve prensipleri dikkate alınarak, bu makalede, “Sudoku Çözücü Uygulaması” tasarlanmış ve geliştirilmiştir. Ayrıca, uygulamanın çalışma süresi hesaplanıp etkinliği ve kullanılabilirliği ortaya konmuştur. Buna ek olarak, bu çalışma, Sudoku bulmacasının belirli bir çözüm yolunu grafik teorisine dayalı algoritmalar yardımıyla bulup, kullanımı için hem oyun dünyasına, hem bilim dünyasına, hem de eğitim dünyasına sunmuştur.

Anahtar Kelimeler: Grafik teorisi, sudoku, Welsh-Powell algoritması, Karger algoritması.

1 Giriş

Sudoku bulmacası, gazetelerin ve bulmaca web sitelerinin çoğunda bulunabilen dünya çapında popüler bir oyundur. Ayrıca, Google Play ve Apple Store’daki çok sayıda Sudoku uygulamasında açıkça görülen mobil uygulamalar dünyasına da yayılmıştır. Bu nedenle, günümüzde milyonlarca web sitesinin Sudoku sunduğu doğrudur. Bir Sudoku bulmacasını çözenin genellikle bilinmezler için çok tatmin edici olmasının tipik bir sonucu olarak çeşitli ücretsiz oluşturucu yazılımlar mevcuttur [1]. Ayrıca, Avrupa ve ABD’de parasal ödül kazanma olasılığı olan sosyal turnuvalar da vardır [2].

Sudoku popülaritesinin birçok nedeni vardır; en önemlisi harf veya kelime kullanan diğer bulmacalara

karşı sayıları kullanmasıdır ve böylece, bir bulmacanın yapılması ve onun farklı ülkelerdeki farklı dillere sahip insanlar arasında paylaşılması oldukça kolay bir hale gelmektedir. Diğer önemli bir neden ise, Sudoku bulmacasının kolay kuralları olması ve her yaşta kişinin öğrenim geçmişine bakılmaksızın hızlı bir şekilde oyunu öğrenebilmesidir [3].

Sudoku bulmacası, son yıllarda dünyanın her yerinden birçok insanın tutkusu haline gelmiştir. Aslında, Sudoku ile ilgili ilginç gerçek, çözülmesi gereken önemsiz bir bulmaca olmasıdır. Çözümünün önemsiz olmasının nedeni, Sudoku çözümleri için bir algoritmanın mevcut olmasıdır. Bu algoritma, bir çözüm yolu bulana ve bulunana kadar bir ağaçta geriye doğru izlemeye dayanan ağaç tabanlı bir arama algoritmasıdır [4]. Bir kişinin

* Corresponding author.

E-mail address: mbatar@mehmetakif.edu.tr (M. Batar)

Received 14 August 2021; Received in revised form 22 November 2021; Accepted 09 December 2021

2587-1943 | © 2021 IJIEA. All rights reserved.

Doi: <https://doi.org/10.46460/ijiea.982908>

sadece tek yapması gereken şey, kişisel bilgisayarının karşısında oturup, bulmacada verilen sayıları tuşlamak ve ardından bir bilgisayar programının çözümü hesaplamasını ve bulmasını beklemektir. Fakat insanlar bunu yapmak yerine, Sudoku bulmacalarını çözmek için uğraşmaktadırlar. Bunun nedeninin, insanların Sudoku çözümlerini bulmak için kalem ve kâğıtla uğraşmaktan hoşlanmaları olduğu belirtilmektedir. Bu sayede, bulmacayı yapmaya çalışan biri için ciddi bir zihinsel meydan okuma yerine gelmekte ve bu, çözücüye ısrar etmesi için gerekli cesaret vermektedir [5].

Öte yandan, araştırmacıların bakış açısından, Sudoku bulmacalarını çözerken akıllara birçok soru gelmektedir: Benim bulmacamın bir çözümü var mı? Eğer öyleyse, benim bulmacam için tek mi? Daha az ipucu olsaydı bulmaca daha da zorlaşır mı? Benzersiz bir çözüm sağlamak için minimum ipucu sayısı nedir? En zorunu için bile, farklı Sudoku türlerini çözmek için etkili algoritmalar var mı? Bu ve benzeri sorular, araştırmacılara bu soruların yanıtlarını keşfetmelerini ve bu bulmacaları oluşturup çözmek için algoritmaları uygulamaları için çekici fırsatlar sunmaktadır [6]. Bunun doğal bir sonucu olarak, bu probleme yaklaşıldığı zaman, birçok sayıda strateji ve çözüm yolu olduğu ortaya çıkmaktadır.

2 Sudoku

Sudoku bulmacası 9x9'lük bir tahtada oynanmaktadır. Tahtada, örtüşmeyen dokuz adet 3x3'lük alt panoya bölünmüş seksen bir hücre bulunmaktadır. Bu alt panolar kutu olarak adlandırılabilir ve aşağıda Şekil 1'de görüldüğü gibi panonun sol üst köşesinden başlayarak klasik daktilo sırasına göre 1'den 9'a kadar numaralandırılmaktadır [7]. Tahtadaki belirli bir hücreye atıfta bulunma gösterimi, satır numarasının ardından sütun numarasını vermektir. Örneğin, C(6, 7) gösterimi; 6. satır ve 7. sütundaki C (cell-hücre) hücresini belirtmektedir.

	1	2	3	4	5	6	7	8	9
1									
2		1			2			3	
3									
4									
5		4			5			6	
6									
7									
8		7			8			9	
9									

Şekil 1. Sudoku tahtası.

Sudoku, mantık tabanlı, öğrenmesi kolay bir sayı yerleştirme bulmacasıdır. Sudoku kelimesi, "sayılar tek olmalı" anlamına gelen "Su-ji wa dokushin ni kagiru"nun kısaltmasıdır. Sudoku bulmacasının kökleri İsviçre'ye dayanmaktadır. Leonhard Euler, 18. yüzyılda Sudoku

bulmacasına benzeyen ancak tek tek bölgelerin içeriği üzerinde ek kısıtlamalar olmaksızın "carré latin"i yaratmıştır. Ancak, ilk gerçek Sudoku 1979'da yayınlanmış ve Amerikalı bir mimar olan Howard Garns tarafından icat edilmiştir. Dünya çapındaki gerçek popülerlik ise, 1986 yılında Japonya'da yayımlandıktan ve "Nikoli" tarafından Sudoku adı verildikten sonra başlamış ve ortaya çıkmıştır [8].

Bir Sudoku bulmacası dokuz sütuna, satıra ve bölgeye bölünmüş 81 hücreden oluşmaktadır. Görev, 1'den 9'a kadar olan sayıları boş hücrelere, her satır, sütun ve 3x3'lük bölgede, her sayı sadece bir kez görünecek şekilde yerleştirmektir. Bir Sudoku'da en az 17 tane verilen sayı vardır, ancak normalde 22'den 30'a kadar sayı verilmektedir. Sudoku, matematik tabanlı bir bulmaca değil, mantık tabanlıdır. Harfler ve hatta bazı semboller ile çözülmemiş bir Sudoku bulmacası yapmak oldukça mümkündür. Biraz ilginç bir nokta ise, olası Sudoku bulmacalarının "6.670.903.752.021.072.936.960" (altı seksilyon altı yüz yetmiş kentilyon dokuz yüz üç katrilyon yedi yüz elli iki trilyon yirmi bir milyar yetmiş iki milyon dokuz yüz otuz altı bin dokuz yüz altmış) tane olmasıdır [9]. Böylece, insanlar günde sayısız Sudoku bulmacası oynayabilmekte ve yine de yenileri yerine gelebilmektedir.

3 Sudoku Tarihçesi

Sayı bulmacaları, 19. yüzyılın sonlarında, Fransız bulmaca hazırlayıcılarının sayıları sihirli karelerden çıkarmayı denemeye başladıkları birkaç gazete de ortaya çıkmıştı. Paris gazetesi "Le Siècle" 19 Kasım 1892'de kısmen tamamlanmış 9x9'lük sihirli kareyi 3x3'lük alt karelerle yayınlamıştı. Bu bir Sudoku değildi, çünkü çift haneli sayılar içermekteydi ve mantıktan ziyade çözümünde aritmetik işlemler gerektirmekteydi. Ancak yine de temel özellikleri bakımından Sudoku'yu andırmaktaydı [10].

6 Temmuz 1895'te "Le Siècle'in rakibi" – La France – bulmacayı neredeyse modern bir Sudoku olacak şekilde geliştirmişti. 9x9'lük sihirli kare bulmacayı basitleştirdi ve böylece her satır, sütun ve kırık köşegenler yalnızca 1-9 sayılarını içermekteydi, ancak alt kareleri işaretlenmedi. İşaretlenmemiş olmalarına rağmen, her 3x3'lük alt kareler gerçekten 1-9 sayılarını içermekteydi ve kırık köşegenler üzerindeki ek kısıtlamalar yalnızca bir yolla çözülebilmekteydi. Bu haftalık bulmacalar, yaklaşık on yıl boyunca "L'Echo de Paris" gibi Fransız gazetelerinin ortak ve eğlenceli bir özelliği idi, ancak I. Dünya Savaşı sırasında maalesef bu uygulama ortadan kalktı [11].

Modern Sudoku, çok büyük olasılıkla, Connorsville, Indiana'dan 74 yaşındaki emekli bir mimar ve serbest bulmaca kurucusu olan Howard Garns tarafından anonim olarak tasarlandı ve ilk olarak 1979'da "Dell Magazines" tarafından "Number Place" olarak yayınlandı (bilinen en eski modern Sudoku örnekleri). Garns'ın adı, "Dell Kalem Yapbozları" ve "Sayı Yeri"ni içeren "Kelime Oyunları" sayılarında katkıda bulunanlar listesinde her zaman yer almaktaydı. Yaratılışını dünya çapında bir fenomen olarak görme şansını bulmadan 1989 yılında vefat etti. Garns'ın yukarıda listelenen Fransız gazetelerinden herhangi birine tanıdık olup olmadığı bilgisi ise hala belirsizliğini korumaktadır [12].

Bulmaca Japonya’da “Nikoli” tarafından Nisan 1984’te “Monthly Nikolist” gazetesinde “Sūji wa dokushin ni kagiru (数字は独身に限る)” olarak tanıtıldı, bu aynı zamanda “rakamlar tek olmalıdır” veya “rakamlar tek bir olayla sınırlıdır” (Japonca’da dokushin “evli olmayan kişi” anlamına gelmektedir) anlamlarına gelmektedir. Daha sonraki bir tarihte, isim Maki Kaji tarafından Sudoku (数独) olarak kısaltıldı ve daha kısa bir sürüm oluşturmak için bileşik kelimelerin ilk bölümünü kendi üzerine aldı [13].

Sudoku bulmacası Japonya’da tescilli bir ticari markadır ve bulmaca genellikle “Numara Yeri” veya daha gayri resmi olarak iki kelimenin bir portmantosu - “Num(ber) Pla(ce)” olarak adlandırılmaktadır. 1986’da “Nikoli” iki yenilik getirmişti: verilenlerin sayısı 32 ile sınırlanmış ve bulmacalar simetrik hale getirilmişti; yani verilenler rotasyonel simetrik hücrelere dağıtılmıştı. Böylece, “Asahi Shimbun” gibi ana akım Japon dergilerinde Sudoku bulması yayınlanmıştı [14].

“Times of London”, Wayne Gould’un çabalarından yola çıkarak yerel bir ABD gazetesinde başarılı bir görünümünden sonra 2004 yılının sonlarında Sudoku’ya yer vermeye başlamış ve düzenli olarak bu durum, hızla diğer gazetelere yansımış ve yayılmıştı. Gould, benzersiz bulmacaları hızla üretmek için bir bilgisayar programı tasarlayıp geliştirmişti [15].

Sudoku bulmacası, günümüzde hala popülerliğini korumakta ve 7’den 70’e herkes tarafından oynanmaya devam etmektedir.

4 Sudoku Çözümü

Sudoku bulmacasının çözüm yolu tek değildir ve birden fazla yol/yöntem mevcuttur [16]. Ancak, çözüme giden yolu göstermek için bir rehber hazırlamak mümkündür. Bu amaç doğrultusunda, aşağıdaki mevcut adımlar izlenerek Sudoku bulmacası çözülebilmektedir. Bununla birlikte, Sudoku çözümüne yeni başlayanlar için bir yol gösterici niteliği taşımaktadır. Toplam 9 adımdan oluşan bu çözüm stratejisi adım adım açık bir şekilde gösterilmiştir.

Adım 1								
			9			7	2	8
2	7	8			3		1	
	9				6	4		
	5			6		2		
		6				3		
	1			5				
1			7		6		3	4
			5		4			
7	4	9	1			8		5

Şekil 2. Sudoku çözümünün ilk adımı.

1. adımda (Şekil 2), Sudoku bulmacasının bir örneği yer almaktadır. Yeni başlayanlar için, anlaşılır olması bakımından seviyesi kolay bir örnek (Sudoku bulmacası) verilmiştir. Bulmacayı çözüme başlamak için ilk yapılması gereken en sık görünen sayı ve sayıları aramaktır.

Adım 2								
			9			7	2	8
2	7	8			3		1	
	9				6	4		
	5			6		2		
		6				3		
	1			5		4		
1			7		6		3	4
			5		4			
7	4	9	1			8		5

Şekil 3. Sudoku çözümünün ikinci adımı.

1. adımdan sonra 2. adıma (Şekil 3) geçildiğinde, en sık görünen sayı olarak “4” sayısı ele alınmıştır. Bununla birlikte, “4” sayısı olmayan bölgeler aranmış ve eleme işlemi ile “4” sayısının yerleştirilmesine çalışılmıştır. Bu amaçla hangi sütun ve satırların “4” sayısını içerdiği belirlenmiş, bu satır ve sütunlara “4” sayısının konulamayacağı tespit edilmiştir. Bunun sonucunda, hangi bölgede tek olası yer kalmışsa, “4” sayısı oraya yerleştirilmiştir. Şekil 3’de gösterilen iki tane “4” sayısı bu şekilde elde edilmiştir. Ortaya konan ve uygulanan bu strateji, çapraz tarama olarak isimlendirilmektedir.

Adım 3								
			9			7	2	8
2	7	8			3	5	1	
	9				6	4		
	5			6		2		
		6				3	5	
	1			5		4		
1		5	7		6		3	4
			5		4			
7	4	9	1			8		5

Şekil 4. Sudoku çözümünün üçüncü adımı.

2. adımdan sonra 3. adım (Şekil 4) için de, çapraz tarama işlemi “5” sayısı ile tekrarlanmıştır. Bu işlem neticesinde, 2. adımdaki “4” sayısından daha etkin bir sonuç ortaya çıkmış ve üç tane “5” sayısı bulmacaya yerleştirilmiştir.

Adım 4								
			9			7	2	8
2	7	8	6	4	3	5	1	9
	9					6	4	3
	5			6		2		
		6				3	5	
	1			5		4		
1		5	7		6	9	3	4
			5		4	1		
7	4	9	1			8		5

Şekil 5. Sudoku çözümünün dördüncü adımı.

4. adımda (Şekil 5) satırlar, sütunlar ve bölgeler sadece iki boş hücre ile tamamlanabilmiştir. Bu durumu ortaya çıkarabilmek oldukça kolaydır, çünkü her iki hücre grubundan bir hücre için yalnızca bir olası sayı geride kalmıştır. Bu uygulamanın anlaşılmasını kolaylaştırmak için ilgili sayılar Şekil 5’de verildiği gibi renklerle – “1” sayısı için kırmızı yuvarlak renk, “6” sayısı için mor yuvarlak renk ve “9” sayısı için yeşil yuvarlak renk – vurgulanmış ve yuvarlak içine alınmıştır. Bu işlemin sonucunda, sonraki adımlar için çapraz tarama ve sayma işlemlerini yapmak oldukça kolaylaşmıştır.

Adım 5								
			9			7	2	8
2	7	8	6	4	3	5	1	9
	9					6	4	3
	5			6		2		
		6				3	5	
	1			5		4		
1		5	7		6	9	3	4
			5		4	1	7	2
7	4	9	1			8	6	5

Şekil 6. Sudoku çözümünün beşinci adımı.

4. adımın ardından 5. adımda (Şekil 6) ise, sol alt bölgede, “6” sayısının tam konumu belirlenememiş, ancak hangi sıraya konması gerektiği – Şekil 6’da sol altta, üç kırmızı hücre ile gösterilen yer – ortaya çıkarılmıştır. Bunun neticesi olarak, “6” sayısı sağ alt bölgeye yerleştirilmiştir. Ardından ise, 4. adımda izlenen ve uygulanan boş hücre yolu ile “7” ve “2” sayıları elde edilmiştir.

Adım 6								
			9			7	2	8
2	7	8	6	4	3	5	1	9
	9					6	4	3
	5	7		6		2		
		6				3	5	
	1	2		5		4		6
1	2	5	7	8	6	9	3	4
			5	9	4	1	7	2
7	4	9	1			8	6	5

Şekil 7. Sudoku çözümünün altıncı adımı.

6. adımda (Şekil 7) ise, çapraz tarama yöntemi ile sayma işlemleri denenmiştir. Ayrıca, bir hücre için eksik sayıları belirlemek için bölgeler, satırlar ve sütunlar sayılmıştır. Sayıdan sonra tek bir olası sayı ile bir hücre bulunup, sayı hücreye yerleştirilmiştir. Sayım bulgularını ezberlemeye yardımcı olması bakımından ilgili hücrelere küçük sayılar yazılmıştır. Sayma sonucunda, bu adımda, Şekil 7 de görüldüğü gibi altı sayı – “2”, “2”, “6”, “7”, “8” ve “9” sayıları – elde edilmiştir.

Adım 7								
			9			7	2	8
2	7	8	6	4	3	5	1	9
	9					6	4	3
	5	7		6		2		1
		6				3	5	7
	1	2		5		4		6
1	2	5	7	8	6	9	3	4
			5	9	4	1	7	2
7	4	9	1	3	2	8	6	5

Şekil 8. Sudoku çözümünün yedinci adımı.

6. adımdan sonra 7. adımda (Şekil 8), 4. adımdaki boş hücre arama yolu ile Sudoku bulmacasının bir satırı ve bir sütunu daha tamamlanmıştır.

Adım 8								
			9	1		7	2	8
2	7	8	6	4	3	5	1	9
	5	9	1			6	4	3
	5	7		6		2		1
	8	6				3	5	7
	1	2		5		4		6
1	2	5	7	8	6	9	3	4
		3	5	9	4	1	7	2
7	4	9	1	3	2	8	6	5

Şekil 9. Sudoku çözümünün sekizinci adımı.

8. adımda (Şekil 9), 6. adımdaki gibi sayma yoluna gidilmiştir ve bu işlemin sonucunda, 5 yeni sayı – “1”, “1”, “3”, “5” ve “8” sayıları – daha bulmacada yerini almıştır.

Adım 9								
6	3	4	9	1	5	7	2	8
2	7	8	6	4	3	5	1	9
5	9	1	2	7	8	6	4	3
4	5	7	3	6	9	2	8	1
9	8	6	4	2	1	3	5	7
3	1	2	8	5	7	4	9	6
1	2	5	7	8	6	9	3	4
8	6	3	5	9	4	1	7	2
7	4	9	1	3	2	8	6	5

Şekil 10. Sudoku çözümünün son adımı.

Son olarak 9. adımda (Şekil 10), Sudoku bulmacasının geriye kalan tüm sayıları eldeki çapraz tarama, sayma ve boş hücre yollarından yararlanılarak doldurulmuş, işlenmiş ve çözüm yolu ve stratejisi tam olarak ortaya çıkmıştır. Bu geliştirilen rehber sayesinde 9 adımda Sudoku bulmacası çözülebilmekte ve yeni başlayanlar için bir başucu bilgisi oluşturmaktadır.

5 Yöntem

Sudoku bulmacasının çözüm yolunu (çözüm yaklaşımını) veren “Sudoku Çözücü Uygulaması” tasarlanırken ve geliştirilirken grafik teorisine dayalı iki özel algoritma kullanılmıştır. Bunlar, Welsh-Powell algoritması ve Karger algoritmasıdır.

Welsh-Powell metodu, grafik renklendirme problemi için bir algoritma uygulamaya ve çözüm bulmaya yöneliktir [17]. Statik bir grafik üzerinde çalışan ağgözlü bir algoritma sağlar ve bu yinelemeli bir ağgözlü algoritmadır [18]. Bu algoritmanın işleyiş, çözüm ve strateji adımları aşağıdaki gibidir.

Adım 1: Tüm köşeler, derecelerinin azalan değerine göre “V” (vertices-köşeler) listesinde sıralanır.

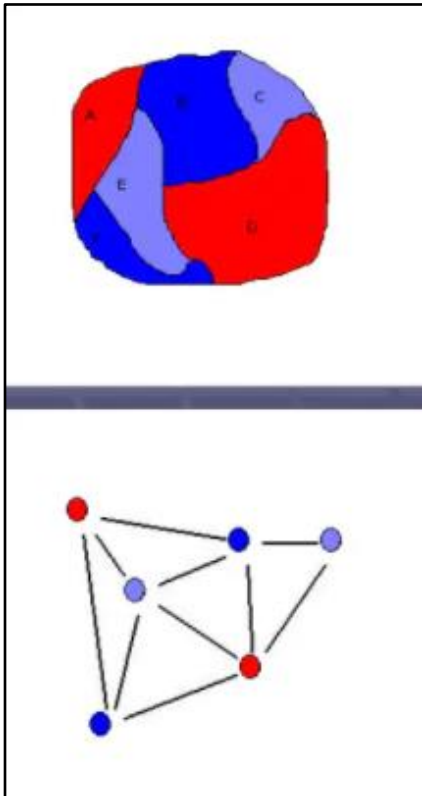
Adım 2: Renkler “C” (colors-renler) listesinde sıralanır.

Adım 3: “V”deki ilk renksiz “v” köşesi, “C”deki mevcut ilk renkle renklendirilir. Mevcut renk, algoritma tarafından daha önce kullanılmayan bir renk anlamına gelmektedir.

Adım 4: Sıralı “V” listesinin kalan kısmına geçilir ve aynı renk, komşu tepe noktalarının aynı renge sahip olmadığı her tepe noktasına atanır.

Adım 5: Tüm köşeler renklendirilene kadar 3. ve 4. adımlar tekrarlanarak uygulanır.

Ortaya konan bu adımların neticesinde, Şekil 11’de Welsh-Powell algoritmasının temelini oluşturan, işleyiş tarzını yansıtan, çalışma prensibini ortaya koyan ve daha iyi anlaşılmasını sağlayan grafik renklendirme ile ilgili yapılan çalışma gösterilmiştir.



Şekil 11. Welsh-Powell yöntemi ile grafik renklendirme.

Karger metodu en küçük kesimi bulmaya çalışır [19]. Bu kesim, verilen yönsüz ve ağırlıksız bir grafiğe dayalı olarak grafiği iki bileşene ayıran en küçük kenar sayısını

ifade etmektedir. Ayrıca, giriş grafiği paralel kenarlara sahip olabilmektedir. Çözüm olarak, minimum kesimi bulmak için “Max-Flow tabanlı s-t kesim algoritması” [20] kullanılır. Her köşe çifti kaynak “s” ve “t” havuzu olarak kabul edilmektedir ve “s-t” kesimini bulmak için minimum “s-t” kesme algoritması olarak adlandırılmaktadır. Ardından tüm “s-t” kesimlerinin minimumunu döndürmektedir. Bu algoritmanın mümkün olan en iyi zaman karmaşıklığı, bir grafik için $O(V^5)$ ’dir – V:vertices-köşeler. Aşağıda bu amaç için Karger yöntemi basit bir şekilde adım adım açıklanmıştır [21]. Bu algoritma $O(E) = O(V^2)$ – E:edges-kenarlar – zamanında uygulanabilmektedir [22].

Adım 1: “CG” (cell to graph) grafiğini orijinal grafiğin kopyası olarak başlatın.

Adım 2: 2’den daha fazla köşe varken;

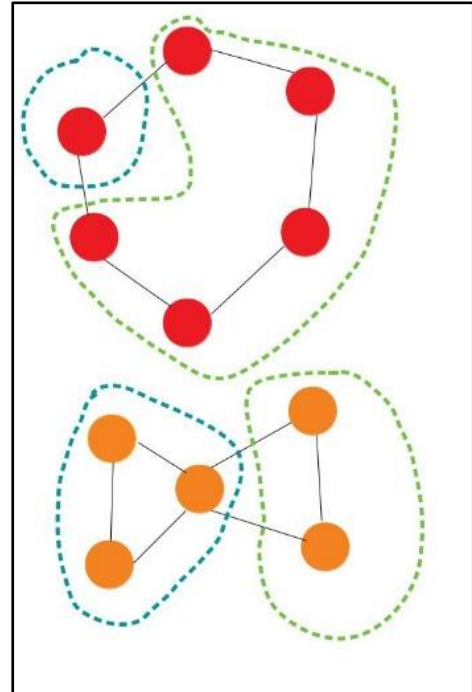
Büzülme grafiğinde rastgele bir kenar (u, v) seçin.

“u” ve “v” yi tek bir köşede birleştirin ve mevcut grafiği güncelleyin.

Kendi kendine döngüleri ortadan kaldırın.

Adım 3: İki köşe ile temsil edilen kısımlar kesilene kadar 2. adımı tekrar edin.

Ortaya konan bu adımların neticesinde, aşağıdaki Şekil 12’de Karger algoritmasının temelini oluşturan, işleyiş tarzını yansıtan, çalışma prensibini ortaya koyan ve daha iyi anlaşılmasını sağlayan kısaltma/daraltma ile ilgili yapılan çalışma gösterilmiştir.



Şekil 12. Karger algoritması ile kısaltma/daraltma.

6 “Sudoku Çözücü Uygulaması”

“Sudoku Çözücü Uygulaması” tasarlanıp geliştirilirken Yöntem bölümünde de açıklandığı gibi Welsh-Powell ve Karger algoritmaları kullanılmıştır. Uygulama ortaya çıkarılırken işin önemli noktası renklendirilecek tepe ve tepe noktalarını seçmektir [23]. Önerilen algoritmada, doymuş derece sıralaması (saturated degree order) [24] ve derece sıralamasının bir kombinasyonunu [25] kullanılmaktadır. Bir tepe

noktasının doygunluk derecesi, bitişik farklı renkteki köşelerinin sayısı ile ölçülmektedir. Köşeler doygunluk derecelerine göre sıralanmakta ve eğer iki köşe aynı dereceye sahipse, dereceleri karşılaştırılarak bağlar bozulup çözümüleme işlemi yapılmaktadır.

```

int colorNumber = 1; //number of used colors
int numberOfColoredNodes = 0;

while (numberOfColoredNodes < graph.Count)
{
    var max = -1;
    var index = -1;

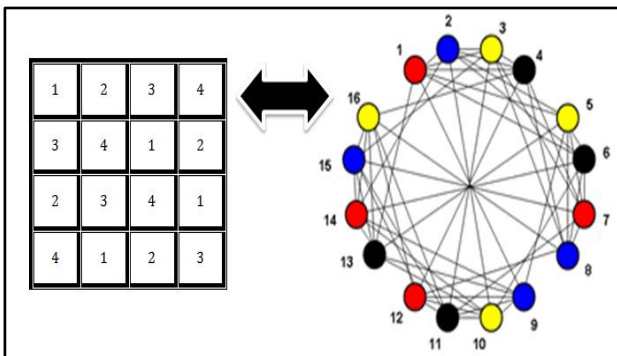
    for (int i = 0; i < graph.Count; i++)
    {
        if (!Colored(graph.Nodes[i], nodeSet))
        {
            var d = SaturatedDegree(graph.Nodes[i], nodeSet);
            if (d > max)
            {
                max = d;
                index = i;
            }
            else if (d == max)
            {
                if (Degree(graph.Nodes[i]) > Degree(graph.Nodes[index]))
                {
                    index = i;
                }
            }
        }
    }
    AssignColor(graph.Nodes[index], nodeSet, ref colorNumber);
    numberOfColoredNodes++;
}

```

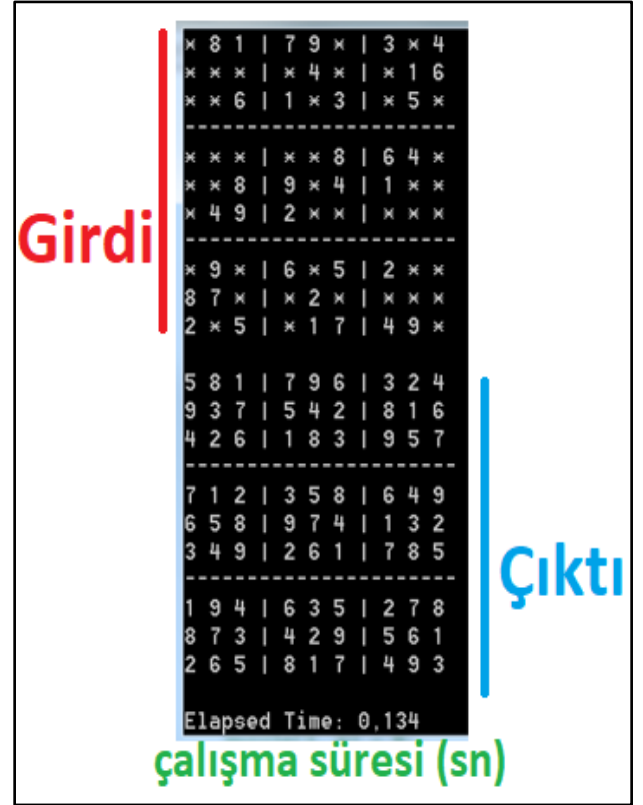
Şekil 13. Uygulamada kullanılan hibrit yöntemin kodu.

Şekil 13’de Welsh-Powell ve Karger algoritmalarına bağlı olarak geliştirilen etkin “Sudoku Çözücü Uygulaması”na ait hibrit yöntemin kaynak kodu verilmiştir.

Bir Sudoku bulmacası (9x9), her hücre için bir tane olmak üzere 81 köşeli bir grafik olarak düşünülebilir ve eğer aynı değere atanamıyorlarsa iki köşe bir kenarla bağlanmaktadır [26]. Örneğin, aynı satır, sütun veya bloktaki tüm hücreler, karşılık gelen köşeleri arasında kenarlara sahip olmaktadır. Buna ek olarak, verilen bir Sudoku bulmacası ile ilgili bir grafik oluşturulabilmektedir. Bulmacada verilen sayı, grafiğe ek kenarlar eklemek için kullanılabilir; daha sonra bu grafiğin 9 rengini bulmak için grafik boyamada söz sahibi olmaktadır (renkler 1-9). Bu renklendirme yöntemine bağlı olarak ortaya çıkartılan uygulama Şekil 14’de gösterilmiş ve “Sudoku Çözücü Uygulaması”na ait girdi ve çıktılar da “Şekil 15”te çalışma süresi ile birlikte ortaya konmuştur.



Şekil 14. Uygulamada kullanılan renklendirme işlemi.



Şekil 15. “Sudoku Çözücü Uygulamasının” etkin sonucu.

Ayrıca “Sudoku Çözücü Uygulaması”nın zorluk seviyeleri ile ilgili çıktılarına bakıldığı zaman oldukça iyi bir performans sergilediği görülmektedir. Zorluk derecesi sudoku bulmacasında verilen ipucu sayısına göre değişmektedir [27]. Eğer bulmacanın başında verilen ipucu sayısı 19 ile 26 arasında ise zor, 27 ile 36 arasında ise orta, 36’dan fazla ise kolay olarak nitelendirilmektedir [28]. Aşağıda Tablo 1’de, geliştirilen “Sudoku Çözücü Uygulaması”nın bulmacanın zorluk seviyelerine göre verdiği cevapların çalışma süreleri listelenmiştir.

Tablo 1. Uygulamanın çalışma süresi

Zorluk Seviyesi	Çalışma Süresi (sn)
Düşük	~ 0,16 sn
Orta	~ 0,39 sn
Yüksek	~ 0,74 sn

7 Sonuç

Bu çalışma, tüm dünyada çok popüler bir oyun olan ve 7’den 70’e herkes tarafından oynanan Sudoku bulmacasını, anlamını, nereden geldiğini, gelişimini örnekleriyle beraber açıklamıştır. Bununla birlikte, Sudoku’nun çözümü (çözüm yolu) ile grafik teorisinin yaklaşımı ve içeriği arasındaki ilişkiyi açıkça göstermiştir. Ayrıca grafik yöntemleri içinde yer alan; grafiği renklendirmeyi amaçlayan Welsh-Powell algoritması ve grafiği daraltmayı amaçlayan Karger algoritması ile Sudoku bulmacası için yeni ve etkin bir çözüm yolu geliştirilmiş ve ortaya çıkarılmıştır. Bu etkin “Sudoku Çözücü Uygulaması” sayesinde, “Sudoku” ve “Grafik”in birbirine çok iyi uyduğu ve birbirini çok iyi tamamladığı bu çalışma ile açıkça gösterilmiştir. Oyun sektörüne, bilişime, bilime ve eğitime hizmet eden bir çalışma ortaya konmuştur.

Bilgilendirme

Bu çalışmada etik kurul onay belgesine ihtiyaç bulunmamaktadır ve yazar herhangi bir çıkar çatışması olmadığını beyan etmektedir.

Kaynaklar

- [1] Maji, A., Roy, S., & Pal, R. (2013). A novel algorithmic approach for solving Sudoku puzzle in guessed free manner. *European Academic Research*, 1(6), 1126-1154.
- [2] Semeniuk, I. (2005). Stuck on you. *New Scientist*, 31, 45-47.
- [3] Mandal, S., & Sadhu, S. (2013). Solution and level identification of Sudoku using harmony search. *International Journal of Modern Education and Computer Science*, 5(3), 49-55.
- [4] Mandal, S., & Sadhu, S. (2011). An efficient approach to solve Sudoku problem by harmony search algorithm. *An International Journal of Engineering Sciences*, 4, 312-323.
- [5] Crook, J. F. (2009). A pencil-and-paper algorithm for solving Sudoku puzzles. *Notices of the American Mathematical Society*, 56(4), 460-468.
- [6] Majumder, A., Kumar, A., Das, N., & Chakraborty, N. (2010). The game of Sudoku-advanced backtrack approach. *International Journal of Computer Science and Network Security*, 10(8), 22-33.
- [7] Herzberg, A. M., & Murty, M. R. (2007). Sudoku squares and chromatic polynomials. *Notices of the AMS*, 54(6), 708-718.
- [8] Deng, X., Li, J., & Li, G. (2013). Research on Sudoku puzzles based on metaheuristics algorithm. *Journal of Modern Mathematics Frontier (JMMF)*, 2(1), 25-32.
- [9] Crawford, B., Castro, C., & Monfroy, E. (2009). Solving sudoku with constraint programming. In *International Conference on Multiple Criteria Decision Making* (pp. 345-348). Springer, Berlin, Heidelberg.
- [10] Yongda, L., & Xiuqin, D. (2011). Solving Sudoku puzzles based on improved genetic algorithm. *Computer Applications and Software*.
- [11] Lewis, R. (2007). Metaheuristics can solve Sudoku puzzles. *Journal of Heuristics*, 13(4), 387-401.
- [12] Pillay, N. (2012). Finding solutions to Sudoku puzzles using human intuitive heuristics. *South African Computer Journal*, 49, 25-34.
- [13] Soto, R., Crawford, B., Galleguillos, C., Monfroy, E., & Paredes, F. (2013). A hybrid AC3-tabu search algorithm for solving Sudoku puzzles. *Expert Systems with Applications*, 40(15), 5817-5821.
- [14] Boryczka, U., & Juszczak, P. (2012). Solving the Sudoku with the differential evolution. *Zeszyty Naukowe Politechniki Białostockiej. Informatyka*, 9, 5-16.
- [15] Deng, X., & Li, Y. (2013). A novel hybrid genetic algorithm for solving Sudoku puzzles. *Optimization Letters*, 7(2), 241-257.
- [16] Rosenhouse, J., & Taalman, L. (2011). Taking sudoku seriously: *The math behind the world's most popular pencil puzzle*. OUP USA.
- [17] Olariu, S., & Randall, J. (1989). Welsh-Powell opposition graphs. *Information Processing Letters*, 31(1), 43-46.
- [18] Zhou, S. (1999). A sequential coloring algorithm for finite sets. *Discrete Mathematics*, 199(1-3), 291-297.
- [19] Karger, D. R., Klein, P. N., & Tarjan, R. E. (1995). A randomized linear-time algorithm to find minimum spanning trees. *Journal of the ACM*, 42(2), 321-328.
- [20] Neumayera, S., Efrat, A., & Modiano, E. (2015). Geographic max-flow and min-cut under a circular disk failure model. *Computer Networks*, 77, 117-127.
- [21] Karger, D. R., & Motwani, R. (1997). An NC algorithm for minimum cuts. *SIAM Journal on Computing*, 26(1), 255-272.
- [22] Karger, D. R. (1999). Random sampling in cut, flow, and network design problems. *Mathematics of Operations Research*, 24(2), 383-413.
- [23] Doumont, J., & Vandebroek, P. (2002). Choosing the right graph. *IEEE Transactions on Professional Communication*, 45(1), 1-6.
- [24] Sharieh, A., & Sabri, K. E. (2008). Parallel graph colouring based on saturated degree ordering. *Basic Sci. & Eng.*, 17(2), 489-503.
- [25] Yamamoto, T., Brewster, R., & Safran, S. A. (2010). Chain ordering of hybrid lipids can stabilize domains in saturated/hybrid/cholesterol lipid membranes. *EPL (Europhysics Letters)*, 91(2).
- [26] Omari, H., & Sabri, K. E. (2006). New graph coloring algorithms. *American Journal of Mathematics and Statistics*, 2(4), 439-441.
- [27] Edgington, J. (2006). Solving sudoku puzzles: Nifty course assignments. *Journal of Computing Sciences in Colleges* 21(4), 90-91.
- [28] Najafian, M., Tadayon, M. H., & Esmaeili, M. (2020). Construction of strongly mutually distinct sudoku tables and solid sudoku cubes by cyclotomic cosets. *IEEE Transactions on Games*, 12(1), 54-62.