

T.C.
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MALATYA MERKEZİLİK TABANLI GRAF RENKLENDİRME YÖNTEMİNİN SUDOKU VE 8 VEZİR KISIT TAT-
MİN PROBLEMLERİNE UYGULANMASI

Erkan KARAGÖZ

Yazılım Mühendisliği Ana Bilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı

Doç. Dr. Selman YAKUT

Tez Jüri Üyeleri

Doç. Dr. Furkan ÖZTEMİZ

Doç. Dr. Selman YAKUT

Doç. Dr. Fatih TOPALOĞLU

MALATYA

2026

T.C.
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜNE

ETİK ve YAPAY ZEKÂ KULLANIM BEYANI

02 / 07 / 2026

İnönü Üniversitesi Lisansüstü Tez Yazım Kurallarına uygun olarak, “Doç. Dr. Selman Yakut” danışmanlığında hazırlayıp sunduğum “Malatya Merkezilik Tabanlı Graf Renklendirme Yönteminin Suda ve 8 Vezir” başlıklı Yüksek Lisans tezimde:

1. ETİK ve BİLİMSEL SORUMLULUKLARIM:

- Tezimde yer alan verileri, bilgileri ve belgeleri akademik ve etik kurallar çerçevesinde elde ettiğimi, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Çalışmada yararlandığım tüm kaynaklara bilimsel kurallara uygun şekilde atıfta bulunduğumu ve kaynakça gösteriminde APA 7. baskı kurallarına uyduğumu,
- Tezimin özgün olduğunu, çalışma ve yazım sürecinde patent ve telif haklarını ihlal edici hiçbir davranışta bulunmadığımı,
- Aksi bir durumda doğabilecek tüm hukuki ve akademik sonuçları kabul ettiğimi,

2. YAPAY ZEKÂ KULLANIMINA İLİŞKİN BEYANIM:

() Tez yazım sürecinde yapay zekâ tabanlı araçlardan **faydalanmadım**.

(X) Tez yazım sürecinde yapay zekâ tabanlı araçlardan **faydalandım**.

Bu durumda:

- Yapay zekâ araçlarını yalnızca bilimsel üretim sürecini desteklemek amacıyla kullandığımı,
 - Kullanımın hipotez geliştirme, teorik tartışma ve yorumlama gibi üst düzey akademik becerileri içermediğini,
 - Kullanımın tezin özgünlüğüne, bilimsel etik kurallarına ve ilgili mevzuata aykırı olmadığını,
 - Usulsüz kullanımlara (kaynak göstermeksizin metin üretimi, yanıltıcı kaynak gösterimi, otomatik içeriklerin özgün eser gibi sunulması vb.) kesinlikle başvurmadığımı,
 - Kullanılan yapay zekâ araçlarının ve işlevlerinin, danışmanımın bilgisi dâhilinde, ekte verilen tabloda şeffaf biçimde belirtildiğini,
 - Yükseköğretim Kurumları Bilimsel Araştırma ve Yayın Faaliyetlerinde Üretken Yapay Zekâ Kullanımına Dair Etik Rehber’de belirtilen ilkelere riayet ettiğimi,
- kabul ve taahhüt ederim.

Erkan Karagöz

İmza

EK: Yapay Zekâ Kullanım Tablosu (varsa)

EK: Yapay Zekâ Kullanım Tablosu

Bu tablo, tez çalışmanızda kullanılan üretken yapay zekâ araçlarının şeffaf biçimde beyan edilmesi amacıyla hazırlanmıştır. Her satırda kullanılan aracın adı, kullanım amacı, hangi bölümde kullanıldığı ve kullanım şekli belirtilmelidir. Tablo, danışman öğretim üyesinin bilgisi dâhilinde doldurulmalı ve gerektiğinde ek belgelerle desteklenmelidir.

No	Üretken Yapay Zekâ Aracı	Kullanım Amacı	Kullanıldığı Bölüm	Kullanım Şekli
1	ChatGPT	Literatür özeti	İlgili Çalışmalar	Literatürdeki görüşleri özetlemek için yönlendirici sorular sorulmuştur. Otomatik üretilen metin aynen kullanılmamıştır.
2	DeepL	Dil kontrolü	Giriş, İlgili Çalışmalar	Dil ve imla kontrolü
3				
4				
5				

İÇİNDEKİLER

SİMGELER VE KISALTMA DİZİNİ	vii
ŞEKİLLER DİZİNİ	viii
TABLolar DİZİNİ	ix
ÖZET	x
ABSTRACT	xi
1.GİRİŞ	1
1.1 Problemin Tanımı ve Önemi.....	1
1.2 Araştırmanın Amacı ve Kapsamı.....	2
1.3 Araştırmanın Özgün Değeri ve Katkıları	3
1.4 Tezin Organizasyonu.....	3
2. İLGİLİ ÇALIŞMALAR	5
3. ÖNERİLEN YÖNTEM ve PROBLEMLER	10
3.1 Malatya Merkezilik (MC) Algoritması.....	10
3.2 Sudoku Çözümü.....	11
3.2.1 Shidoku Çözümü.....	13
3.2.2 Algoritma ve Akış Diyagramı.....	16
3.2.3 Veri Seti	18
3.3 Sekiz-Vezir Çözümü	19
4. DENEYSEL SONUÇLAR	23
4.1 Sudoku Problemine Ait Sonuçlar	23
4.1.1 İlgili Çalışmalar ile Karşılaştırma	23
4.1.2 Graf Renklendirme Algoritmaları İle Karşılaştırma.....	25
4.1.3 Karmaşıklık Analizi	25
4.2 Sekiz-Vezir Problemine Ait Sonuçlar	30
5. SONUÇ ve ÖNERİLER	33
KAYNAKLAR	35

TEŐEKKÖR

Bu tez alıőmasının hazırlanması sürecinde bilgi, deneyim ve deęerli yönlendirmeleriyle bana her zaman destek olan danışman hocam Do. Dr. Selman YAKUT'a içten teşekkürlerimi sunarım.

Ayrıca eęitim hayatım boyunca desteklerini esirgemeyen ve katkılarıyla yanımda olan eşime teşekkür ederim.



SİMGELER VE KISALTMA DİZİNİ

$\psi(uk)$	: D�ğ�m�n�n merkeziliđini ifade eden fonksiyon
C	: Renk ataması yapılan d�ğ�m k�mesi
$d(uk)$	: uk d�ğ�m�n�n derecesi
$d(um)$	: uk d�ğ�m�n�n komřusu olan um d�ğ�m�n�n derecesi
E	: Kenar
G	: Graf
MC	: Malatya Merkezilik Algoritması
$N(uk)$	: İncelenen d�ğ�m�n�n komřuları
$umax$	: En y�ksek merkezilik deđerine sahip d�ğ�m
RLF	: Recursive Largest First Algoritması
V	: D�ğ�m

ŞEKİLLER DİZİNİ

Şekil No	Sayfa No
Şekil 1. Önerilen Metodolojinin Grafikselsel Özeti	13
Şekil 2. Bir Shidoku Bulmacasının Çözüm Adımları	15
Şekil 3. Önerilen Yöntemin Akış Şeması	18
Şekil 4. Kolay Sudoku	19
Şekil 5. Orta Sudoku	19
Şekil 6. Zor Sudoku	19
Şekil 7. 27. Düğüme Ait Mevcut Kenar Bağlantıları	20
Şekil 8. Oluşturulan Graf Yapısı	21
Şekil 9. MC Değerleri Hesaplanmış Graf Yapısı	21
Şekil 10. MC Değerlerine Göre Renklendirilmiş (Çözülmüş) Graf Yapısı	22
Şekil 11. MC Değerlerine Göre Renklendirilmiş Satranç Tahtası	22
Şekil 12. 40.000 Sudoku Veri Kümesi Ortalama Çözüm Süreleri	26
Şekil 13. 200.000 Sudoku Veri Kümesi Ortalama Çözüm Süreleri.....	27
Şekil 14. 400.000 Sudoku Veri Kümesi Ortalama Çözüm Süreleri.....	27
Şekil 15. Kolay Seviye Veri Kümesi Ortalama Çözüm Süreleri.....	28
Şekil 16. Orta Seviye Veri Kümesi Ortalama Çözüm Süreleri	29
Şekil 17. Zor Seviye Veri Kümesi Ortalama Çözüm Süreleri	29
Şekil 18. 64 Vezir Probleminin Grafiği ve Yerleşimi	30
Şekil 19. 8 Vezir Problemi: 12 Benzersiz Çözümün Grafiği ve Yerleştirme Örnekleri.....	31
Şekil 20. 64 Vezir Yerleştirme Probleminin Toplam Çözüm Süreleri.....	32
Şekil 21. Sekiz-Vezir Benzersiz Çözümlerin Toplam Çözüm Süreleri	32

TABLÖLAR DİZİNİ

Tablo No	Sayfa No
Tablo 1. MC Algoritmasının Matematiksel İfadesi	11
Tablo 2. Shidoku Bulmacasının Çözümünün Her Adımında Hesaplanan MC Değerleri	16
Tablo 3. Sudoku Çözümünde Önerilen Yöntemin Sözde Kodu	16
Tablo 5. Farklı Zorluklardaki Sudoku Bulmacalarının Algoritma Başarı Oranları Karşılaştırılması	24
Tablo 6. Zaman Karmaşıklığı Analizi	26



ÖZET

Malatya Merkezilik Tabanlı Graf Renklendirme Yönteminin Sudoku ve 8 Vezir Kısıt Tatmin Problemlerine Uygulanması

Kısıt tatmin problemleri, bir takım kısıtlar altında değişkenlere uygun olan değerlerin atanmasının amaçlandığı ve çözüm uzaylarının karmaşıklığı ve büyüklüğünden dolayı ciddi hesaplama zorlukları barındıran problemlerdir. Sudoku ve Sekiz-Vezir problemleri, içerdikleri kısıtlar ve kombinatoryal yapı nedeniyle bu problem sınıfının en bilinen ve çok incelenen önemli örnekleri arasında yer almaktadır. Bu problemlerin çözüm çalışmalarında geri izleme, kısıt yayılımı, sezgisel arama ve optimizasyon temelli oldukça fazla yöntem tasarlanmış ve geliştirilmiş olmasına karşın, özellikle problemin boyutundaki büyümeler ile çözüm adımlarının etkili bir biçimde yönetilmesi güncelliğini koruyan bir araştırma alanı olarak önemini sürdürmektedir.

Bu tezde, Önemli Kısıt tatmin problemlerinden olan Sudoku ve Sekiz-Vezir problemlerinin çözümünde Malatya Merkezilik Algoritması ile graf teorisine dayalı ve kontrollü graf renklendirme içeren bir yöntem geliştirilmiştir. Önerilen yöntemde problemler birer graf yapısı olarak modellenmiş, problemde yer alan kutucuklar birer düğüm ve bu düğümler arasında yer alan kısıtlar gözetilerek kenar bağlantıları tanımlanmıştır. Üretilen graf yapılarının üzerinde Malatya Merkezilik Algoritması uygulanarak düğümlerin merkezilik değerleri hesaplanmış ve bu değerler düğümlerin önceliklerinin belirlenmesinde kullanılmıştır. Ardından düğümler elde edilen merkezilik değerleri gözetilerek sıralanmış ve kontrollü bir biçimde renklendirme süreci yürütülmüş problem çözümleri elde edilmiştir.

Önerilen yaklaşım, 1.000.000 adet Sudoku bulmacasına kadar ve farklı zorluk seviyelerinde diğer graf renklendirme algoritmaları ile karşılaştırılmış, diğer bir çalışmada ise yaklaşım 64x64 boyutunda kadar büyütülen Sekiz-Vezir problemleri ele alınarak güncel graf renklendirme algoritmaları ile karşılaştırılmış ve incelenmiştir. Sonuçlar göz önüne alındığında, Malatya Merkezilik algoritması düğüm merkezilik hesabına dayalı önceliklendirme mekanizmasının graf renklendirme sürecinin yönlendirilmesinde etkili sonuçlar ürettiğini ve kısıt tatmin problemlerinin çözümüne yönelik uygulanabilir etkili bir yaklaşım sunduğunu göstermektedir.

Anahtar Kelimeler: Grafik renklendirme, Grafik teorisi, Kısıt tatmin problemleri, Malatya Merkezilik Algoritması, Sekiz-vezir problemi, Sudoku

ABSTRACT

Application of the Malatya Centre-Based Graph Colouring Method to Sudoku and the 8-Queen Constraint Satisfaction Problems

Constraint satisfaction problems involve assigning appropriate values to variables under a set of constraints and pose significant computational challenges due to their large solution spaces. Sudoku and the Eight-Queens problem are among the most widely studied examples of this problem class because of their combinatorial structures and constraint-based nature. Although numerous approaches based on backtracking, constraint propagation, heuristic search, and optimization techniques have been developed, effectively managing the solution process, particularly as problem size increases, remains an active research topic.

This thesis proposes a graph-based method that combines the Malatya Centrality Algorithm with a controlled graph coloring approach for solving Sudoku and Eight -Queens problems. In the proposed method, each problem is transformed into a graph structure in which cells are represented as nodes and constraint relationships are represented as edges. The centrality values of the nodes are calculated using the Malatya Centrality Algorithm and are subsequently employed to determine node priorities. The nodes are then ordered according to their centrality values, and solutions are obtained through a controlled graph coloring process.

The proposed approach was evaluated on Sudoku problems with different difficulty levels and on Eight-Queens problems of various sizes. In addition, its performance was compared with several contemporary graph coloring algorithms. The experimental findings demonstrate that the centrality-based node prioritization mechanism effectively guides the graph coloring process and provides a practical approach for solving constraint satisfaction problems.

Keywords : Constraint satisfaction problems, Malatya Centrality Algorithm, graph coloring, Sudoku, Eight-Queens problem, graph theory.

1.GİRİŞ

Bilgisayar bilimleri alanlarında karşı karşıya kalınan çoğu gerçek dünyaya ait problem, bilinen bazı kısıtlar altında olası çözümlerin elde edilmesine dayanmaktadır. Rota planlama, kaynak tahsisi, ağ optimizasyonu, yerleştirme problemleri ve çizelgeleme gibi problemler birçok alanda var olan, kısıt tatmin problemleri olarak tanımlanmaktadır. İlgili problemlerdeki kısıtların oluşturduğu karmaşık yapı ve çözüm uzayının büyümesi, çözümün elde edilmesini zorlaştırmakta ve etkili algoritmalar geliştirilmesini gerektirmektedir. Bu sebeple kısıt tatmin içeren problemlerin çözümünün amaçlandığı farklı ve yeni yöntemlerin uygulandığı çalışmalar yoğunluktadır.

Karmaşık ilişkilerin modellenmesi ve analizinde önemli bir matematiksel araç olarak graf teorisi kullanılmaktadır. Graf renklendirme yaklaşımı özellikle grafta yer alan düğümlerin ve kısıtların (düğümler arası kenar bağlantıları) tanımlanarak problemin çözümünün elde edilmesinde oldukça önemlidir ve etkili sonuçlar üretmektedir. Öte yandan, çözüm aşamasında düğümün öncelik sıralamasını belirlenmesi graf tabanlı yöntemin başarısı için çok fazla önem arz etmektedir. Bu tezde, Malatya Merkezilik (MC) algoritması kullanılarak grafta yer alan düğümlerin derecelerinin (önemlerinin) belirlenmesi ile çözüm sürecinde önemli bir yere sahiptir. Önerilen yaklaşımın uygulanabilirliği, kısıt tatmin problemlerinin iki önemli örneği olan Sudoku ve Sekiz-Vezir problemleri üzerinde incelenmiştir.

1.1 Problemin Tanımı ve Önemi

Kısıt tatmin problemleri, tanımlanmış kısıtlar altında değişkenlere doğru değerlerin atanmasını gerektirdiği ve bilgisayar bilimleri, yapay zekâ ile optimizasyon alanlarında genel olarak ele alınan problem sınıflarından biridir [1]. Bu problemlerde, tanımlanmış tüm kısıtları karşılayan geçerli bir çözümün elde edilmesi asıl amaçtır. Çözüm uzayındaki karmaşık ilişkiler ve buna bağlı oluşan büyüklük nedeniyle çoğu kısıt tatmin problemi yüksek hesaplama maliyetlerine sahiptir. Bu sebeple söz konusu problemlerin verimli ve etkin bir biçimde çözülmesi yıllardır araştırmacıların ilgi odağı olmuştur.

Sudoku ve Sekiz-Vezir problemleri, kısıt tatmin problemlerinin en bilinen örnekleri arasındadır. Sudoku problemi, belirli kısıtlar gözetilerek hücrelerin uygun sayılar ile doldurulmasını gerektiren kombinatorial bir problem olarak tanımlanırken NP-Tam problem sınıfında kabul edilmektedir[2]. Bu problem, en yaygın haliyle 9x9 boyutlarında bir tablo üzerinde oynanır[3]. Bu tablo, her biri 3x3 boyutunda olan dokuz alt bölgeye ayrılmıştır. Sudoku tahtası üzerindeki her bir hücre, hem ait olduğu satır ve sütunla hem de bulunduğu 3x3'lük alt matris ile ilişkilidir[4]. Sekiz-Vezir problemi ise bir satranç tahtası üzerinde sekiz vezirin birbirlerini tehdit etmeyecek şekilde yerleştirmesine dayanan klasik bir kombi-

natoryal optimizasyon problemidir[5]. 1848 yılında Max Bezzel tarafından üretilmiştir[6]. Sonraki çalışmalarda problem güncellenerek N-Vezir problemi şeklinde ele alınmış ve farklı boyutlardaki tahtalar üzerinde incelenmiştir[7].

İlgili çalışmalarda bu problemlerin çözümüne yönelik geri izleme ve kaba kuvvet gibi algoritmalar kullanıldı[8]. Ardından genetik algoritmalar, kısıt programlama teknikleri, sürü zekâsı tabanlı yöntemler ve çeşitli sezgisel yaklaşımlar önerilmiştir[9]. Özellikle problemin boyutunun büyümesi gibi durumlarda çözüm uzayının aynı oranda genişlemesi denenen yöntemlerin performansını doğrudan etkileyebilmektedir. Bu sebeple çözüm sürecinin geliştirilmesi için kullanılabilecek farklı düğüm önceliklendirme mekanizmalarının geliştirilmesi önem taşımaktadır.

Graf teorisi ve graf renklendirme yaklaşımları, kısıt tatmin problemlerinin modellenmesinde ve çözümünde etkili matematiksel araçlar sunmaktadır[10]. Problemlerin graf yapısında olarak temsil ile değişkenler arasındaki ilişkilerin sistematik biçimde analiz edilmesine olanak sağlar, merkezilik ölçüleri graf üzerindeki düğümlerin göreceli önem düzeylerinin belirlenmesini mümkün kılmaktadır[11]. Bu bağlamda MC algoritması, düğümlerin yapısal önemlerinin belirlenmesine yönelik bir merkezilik yaklaşımı olarak öne çıkmakta ve kısıt memnuniyet problemlerinin çözümünde kullanılabilecek potansiyel bir araç sunmaktadır[12].

1.2 Araştırmanın Amacı ve Kapsamı

Bu tezin temel amacı, kısıt tatmin problemlerinin çözümünde MC algoritması ve buna dayalı kontrollü graf renklendirme yaklaşımını temel alan graf tabanlı bir çözüm yöntemi geliştirmek ve yöntemin farklı problem türleri üzerindeki etkinliğini incelemesidir. Bu amaç için Sudoku ve Sekiz-Vezir problemleri çalışma alanı olarak seçilmiştir.

Her iki problem de graf teorisi kullanılarak modellenmiştir. Problemlere ait kutucuklar düğüm, kısıtları ise kenar bağlantıları şeklinde temsil edilmiş ve graf yapıları oluşturulmuştur. Oluşturulan bu graf yapıları üzerinde MC algoritması düğümlerin merkezilik değerleri hesaplanması için kullanılmış ve hesaplanan değerler düğümlerin işleme alınma önceliklerinin belirlenmesinde kullanılmıştır. Ardından kontrollü graf renklendirme yaklaşımı kullanılarak problem çözümleri elde edilmiştir[13].

Tez kapsamında geliştirilen yöntem, Sudoku problemleri üzerinde farklı zorluk seviyelerine sahip ve farklı adetlerdeki veri kümeleri kullanılarak değerlendirilmiştir. Ayrıca yöntemin genellenebilirliğini ortaya koymak amacıyla Sekiz-Vezir probleminin farklı boyutları üzerine deneysel çalışmalar gerçekleştirilmiştir. Elde edilen sonuçlar, literatürde yaygın olarak kullanılan çeşitli graf renklendirme algoritmaları ile karşılaştırmalı olarak analiz edilmiştir.

Bu çalışma, MC algoritmasının kısıt tatmin problemlerinin çözümündeki etkinliğini göstermeyi ve graf renklendirme tabanlı çözüm yaklaşımlarına alternatif ve etkili bir yöntem sunmayı amaçlamaktadır[14].

1.3 Araştırmanın Özgün Değeri ve Katkıları

Bu çalışmanın özgün değeri, MC algoritması ile kontrollü graf renklendirme yaklaşımının kısıt tatmin problemlerinin çözümünde tümleşik bir yapı içerisinde kullanılması ve bu yaklaşımın farklı problem türleri üzerinde uygulanabilirliğinin gösterilmesidir. Bu zamana kadar aynı alanda yapılan ilgili çalışmalarda Sudoku ve Sekiz-Vezir problemleri çoğunlukla birbirinden bağımsız olarak ele alınmış ve farklı çözüm yöntemleri ile incelenmiştir. Bu tezde ise her iki problem ortak bir graf modeli altında değerlendirilmiş ve merkezilik temelli bir düğüm önceliklendirme mekanizması kullanılarak çözülmüştür.

Tez kapsamında sağlanan başlıca katkılar aşağıda sunulmaktadır:

- Sudoku ve Sekiz-Vezir problemleri graf teorisi temelinde modellenmiş ve ortak bir çözüm çerçevesi oluşturulmuştur.
- MC algoritmasının kısıt tatmin problemlerinde düğüm önceliklendirme amacıyla kullanılabilirliği ve çözüm üretebildiği ortaya konulmuştur.
- Merkezilik değerlerini temel alan kontrollü graf renklendirme yaklaşımı geliştirilmiştir.
- Önerilen yaklaşımın farklı kısıt tatmin problemlerine uygulanabileceği gösterilmiştir.
- Geliştirilen yöntemin performansı güncel graf renklendirme algoritmaları ile karşılaştırmalı olarak değerlendirilmiştir.
- Merkezilik analizi ve graf renklendirmenin birlikte kullanılmasına dayalı tümleşik bir çözüm modeli önerilmiştir.

Bu yönleriyle çalışma, MC algoritmasının kullanım alanlarını genişletmekte ve kısıt memnuniyet problemlerinin çözümüne yönelik graf tabanlı yaklaşımlara katkı sağlamaktadır.

1.4 Tezin Organizasyonu

Bu tez beş ana bölümden oluşmaktadır. Birinci bölümde çalışmanın genel çerçevesi ile problemin tanımı ve önemi açıklanmış, araştırmanın amacı, kapsamı ve özgün katkıları sunulmuştur.

İkinci bölümde üzerinde çalışılan problemlere yönelik yapılmış geri izleme, kaba kuvvet, sezgisel yaklaşımlar ve yapay zeka optimizasyonları incelenmiş ve araştırmalar değerlendirilmiştir.

Üçüncü bölümde önerilen yöntem tez kapsamında detayları ile ele farklı problemler üzerinde ele alınmıştır. Bu bölümde MC algoritmasının temel yapısı ile çalışma yöntemi ele alınmıştır. Ardından

Sudoku, Shidoku ve Sekiz-Vezir problemlerinin algoritma yapısı, graf olarak modellenmesi ve kontrollü olarak renklendirmesi adımları hakkında bilgi verilmiştir.

Dördüncü bölümde önerilen yöntemin deneysel sonuçları sunulmuştur. Bu bölümde Sudoku ile Sekiz-Vezir problemleri farklı boyut, adet ve zorluk seviyeleri ile ilgili çalışmalardaki yöntemlerle ve bilinen graf renklendirme algoritmaları ile karşılaştırılmıştır.

Beşinci bölümde ise sonuçlar tez kapsamında değerlendirilerek, yapılan çalışmanın literatüre katkıları üzerinde durulmuş ve gelecekte yapılabilecek araştırmalara yönelik öneriler sunulmuştur.



2. İLGİLİ ÇALIŞMALAR

Sekiz-Vezir problemi, kuralları açısından basit görünmesine rağmen çözüm süreci bakımından karmaşık bir yapıya sahip olduğundan günümüze kadar farklı yöntemlerle ele alınmıştır. Problem aynı zamanda klasik bir kombinatoriyal optimizasyon problemi olarak kabul edilmektedir. Bu nedenle etkin algoritmaların geliştirilmesi ve kısıt tatmin uygulamaları açısından önemli bir araştırma alanı oluşturmaktadır.

Sudoku problemi, karmaşık yapısı ve NP-Tam problem sınıfında yer alması nedeniyle çözümüne yönelik çok sayıda yöntem geliştirilmiştir. Sudoku probleminin çözümü yalnızca teorik karmaşıklık açısından değil, aynı zamanda algoritma verimliliği ve kısıt yayılımı gibi uygulamalı araştırmalar açısından da önem taşımaktadır. Kaba kuvvet, geri izleme ve kısıt tabanlı programlama gibi klasik yöntemler küçük boyutlu problemler için etkili sonuçlar üretebilse de büyük ve karmaşık Sudoku bulmacalarında çözüm sürelerinin uzamasına veya çözüm elde edilememesine neden olabilmektedir[2], [15]. Bu sınırlılıklar, Genetik Algoritmalar ve Harmony Search gibi sezgisel ve meta-sezgisel yöntemlerin kullanılmasını teşvik etmiştir. Bununla birlikte, PSO ve tamsayılı doğrusal programlama gibi yöntemlerin özellikle zor Sudoku problemlerinde yetersiz kaldığı belirtilmiştir[9]. Derin öğrenme ve makine öğrenmesi gibi yapay zekâ tabanlı yaklaşımlar ile Douglas-Rachford algoritması gibi graf teorisi temelli yöntemler de her durumda tutarlı çözümler üretememektedir[10]. İlgili çalışmalarda mantıksal çıkarım ve modern optimizasyon tekniklerine dayalı çok sayıda çalışma bulunmakta olup Sudoku, eğitim ve araştırma açısından disiplinler arası bir çalışma alanı oluşturmaktadır[16]. Sudoku problemleri, başlangıçta doldurulmuş hücre sayısına bağlı olarak kolay, orta ve zor seviyelerde sınıflandırılmaktadır. Bu zorluk seviyelerinin belirlenmesine yönelik olarak entropi ve donma geçişi kavramlarını kullanan yapay zekâ ve makine öğrenmesi tabanlı çalışmalar gerçekleştirilmiştir[17], [18]. Ayrıca makine öğrenmesi algoritmaları kullanılarak Sudoku bulmacalarının zorluk seviyelerinin tahmin edilmesi üzerine araştırmalar yapılmıştır. Genetik algoritma gibi sezgisel yöntemlerle istenilen zorluk seviyesinde Sudoku üretimine yönelik çalışmalar da literatürde yer almaktadır[19]. Bunun yanında geri izleme algoritmasını kullanan yapay zekâ destekli mekatronik sistemlerin Sudoku çözüm performansları incelenmiştir[20]. Kaba kuvvet ve geri izleme algoritmalarının karşılaştırıldığı çalışmalarda geri izleme yönteminin daha başarılı sonuçlar verdiği gösterilmiştir[21].

Önceki çalışmalarda Sudoku probleminin çözümüne yönelik çeşitli sezgisel yöntemler önerilmiştir. Bu çalışmalardan bazılarında Tanner grafi ve Belief Propagation algoritmaları kullanılarak Sudoku çözümü araştırılmıştır[22]. Seyrek optimizasyon tekniklerine dayalı yöntemlerin klasik sezgisel ve kural tabanlı yaklaşımlara göre daha etkili sonuçlar ürettiği gösterilmiştir[23], [24]. Görüntüler üzerin-

den otomatik Sudoku çözümüne yönelik çalışmalarda geri izleme, benzetimli tavlama ve genetik algoritmalar gibi yöntemler ile görüntü işleme tekniklerinin performansları karşılaştırılmıştır[4], [15]. Sudoku renklendirme problemlerinin çözümünde genetik algoritma ve tabu arama gibi yöntemlerden yararlanılmıştır[25]. Ayrıca genetik algoritmalar ile çok kollu bandit yöntemlerini birleştiren yeni hibrit yaklaşımlar geliştirilmiştir[26]. Düğüm tabanlı Coincidence (COIN) algoritması ve NB-COIN yönteminin genetik algoritmalara göre daha başarılı sonuçlar ürettiği gösterilmiştir[27].

Mevcut çalışmalarda Sudoku probleminin yapısının değiştirilerek farklı biçimlerde modellenmesine yönelik çalışmalar da bulunmaktadır. Purge-and-Merge algoritması kullanılarak problem ağaç benzeri yapılara dönüştürülmüş ve çözüm araştırılmıştır[28]. Sudoku problemi üç boyutlu fraktal yapılara dönüştürülerek renklendirme yaklaşımı incelenmiştir[29]. Ayrıca Sudoku problemi 3-SAT problemine dönüştürülmüş ve SP2G23SAT algoritması kullanılarak 3-SAT formülasyonu ile graf renklendirme yaklaşımı birlikte değerlendirilmiştir[30], [31]. Bunun yanında tamsayılı programlama, polinom hesaplamaları ve graf teorisi tabanlı farklı modelleme yaklaşımları da önerilmiştir[32].

Graf teorisi ve graf renklendirme yöntemleri, Sudoku probleminin çözümünde yaygın olarak kullanılan yaklaşımlar arasında yer almaktadır. Sudoku tahtasının graf olarak modellenmesiyle ve graf renklendirme problemine dönüştürülmesi ve geri izleme yöntemleriyle çözülmesi araştırılmıştır[33], [34]. K-Renklendirme yaklaşımları ve tümevarımsal transfer öğrenme yöntemleri kullanılarak çözümler elde edilmiştir[35], [36]. Kısmen renklendirilmiş graflar olarak ele alınan Sudoku bulmacalarının renklendirilmesine ilişkin çeşitli matematiksel teoremler ortaya konulmuştur[37]. Douglas-Rachford algoritması, teorik olarak güçlü ve etkili bir yöntem olarak Sudoku çözümünde kullanılmıştır[2], [38]. DSatur, Greedy ve RLF algoritmalarının karşılaştırıldığı çalışmalarda, yüksek maliyetine rağmen RLF algoritmasının daha etkili sonuçlar verdiği gösterilmiştir[39]. Ayrıca DSatur algoritmasının başarılı bir çözüm yöntemi olduğu vurgulanmıştır[40]. Welsh-Powell algoritması kullanılarak komşuluk matrisi tabanlı çözümler geliştirilmiş, Karger ve Welsh-Powell algoritmalarının birlikte kullanıldığı uygulamalar da önerilmiştir [16], [35].

Literatürde Sudoku probleminin çözümüne yönelik geri izleme, kaba kuvvet, genetik algoritmalar, Harmony Search, derin öğrenme ve makine öğrenmesi gibi çok sayıda yöntem önerilmiştir. Bununla birlikte graf teorisi ve graf renklendirme yaklaşımları Sudoku problemlerinin modellenmesi ve çözümünde önemli bir araştırma alanı oluşturmaktadır. K-Renklendirme, DSatur ve Welsh-Powell gibi algoritmalar, kısıt yayılımı ve kombinatoriyal optimizasyon açısından başarılı sonuçlar ortaya koymuştur[16], [35], [40]. Douglas-Rachford algoritması ve SP2G23SAT gibi hibrit yöntemler teorik açıdan güçlü performans sergilemiştir[2], [31]. RLF algoritması kesin çözümler üretirken, Greedy ve Karger ta-

banlı yöntemler pratik uygulamalarda kullanılmıştır[16], [39]. Bununla birlikte yöntemlerin başarı düzeyleri Sudoku'nun zorluk seviyesi ve başlangıç yerleşimine bağlı olarak değişebilmektedir[17], [41]. Tanner grafi ve Belief Propagation gibi seyrek optimizasyon teknikleri, klasik yöntemlere göre daha esnek çözümler sunmuştur[22]. Sonuç olarak graf tabanlı yaklaşımlar Sudoku probleminin çözümünde önemli teorik ve pratik avantajlar sağlamakla birlikte, genel ve etkili çözümler için algoritmaların problem özelliklerine göre uyarlanması gerekmektedir[29], [30].

Çok kameralı sistemlerde düğümler arasındaki ilişkiler, 8×8 sihirli kare düzeni kullanılarak özellik sıralama aracı şeklinde yeniden tanımlanmıştır. İki farklı kameradan elde edilen özellikler ağırlıklandırılarak sıralanmış ve optimum kamera yerleşimi belirlenmiştir. Böylece yüksek tanıma başarımı elde edilmiştir[42]. Veri noktaları arasındaki komşuluk ilişkilerinin klasik yöntemler yerine zaman tabanlı ölçütlerle yeniden tanımlandığı çalışmalarda ise sınıflandırma ve kümeleme performanslarında iyileşmeler sağlanmıştır. Düğümler arasındaki ilişkilerin alternatif ölçütlerle modellenmesi çözüm süreçlerine farklı bakış açıları kazandırmıştır[43].

Ağ analizlerinde yaygın olarak kullanılan merkezilik ölçüleri, bir düğümün ağ içerisindeki etkinliğini ve önemini belirlemeyi amaçlamaktadır. Derece, yakınlık, arasındalık ve PageRank gibi merkezilik ölçülerinin tanıtıldığı çalışmalarda, ağ yapısına uygun merkezilik ölçüsünün seçilmesinin önemi vurgulanmıştır[44]. Grup merkeziliği üzerine yapılan çalışmalarda ise ağlarda kolektif etkinin önemi ele alınmış, grup merkezilik ölçüleri tanımlanmış ve bu ölçülere yönelik optimizasyon yöntemleri incelenmiştir[45].

MC algoritması kullanılarak minimum baskın küme problemlerinin çözümüne yönelik çalışmalarda, düğümlerin baskınlık değerlerinin derece bilgileri yardımıyla hesaplanabileceği gösterilmiştir[46]. Ayrıca köşe örtüsü, bağımsız küme, baskın küme ve graf renklendirme gibi NP-Zor problemlerde MC değerlerinin etkili sonuçlar ürettiği ortaya konulmuştur[47]. Yakut tarafından önerilen graf renklendirme algoritmasının ise Welsh-Powell, Greedy, DSatur, RLF ve MSISCA gibi algoritmalarla kıyasla belirli graf türlerinde daha başarılı performans gösterdiği ve bazı graf problemleri için optimal sonuçlar verdiğinin matematiksel olarak ispatlandığı belirtilmiştir[48].

Serkan Güldal, Veronica Baugh ve Saleh Allehaibi, klasik geri izleme algoritmasına ek olarak satranç tahtasına boş karelerin yerleştirildiği dinamik küme tabanlı bir yöntem önermiştir[49]. Ancak çalışmalarında yalnızca beş vezir problemi üzerinde deneyler gerçekleştirilmiş ve üstel karmaşıklık azaltılamamıştır. Alfredo Arteaga ve arkadaşları kaba kuvvet algoritmasını doğrusal programlama ve geri izleme yöntemleri ile karşılaştırmış, geri izleme ve kısıt tabanlı algoritmaların kaba kuvvet yaklaşımına göre daha başarılı sonuçlar ürettiğini göstermiştir[8]. Kaba kuvvet yöntemi, vezir sayısının artmasıyla birlikte pratik çözümler üretememektedir.

U. Sarkar ve S. Nag, problemin çözümünde genetik algoritma yaklaşımını kullanmıştır[50]. Rastgele genetik varyasyonlar sayesinde çözüm uzayı hızlı bir şekilde araştırılmıştır. Ancak yöntemin çatışmasız bir çözümü garanti edemediği ve her çalışmada farklı sonuçlar üretebildiği belirtilmiştir. Vinod J. ve Shankar P., genetik algoritmaya ağırlıklı mutasyon yaklaşımını entegre ederek klasik genetik algoritmaya kıyasla daha hızlı ve verimli bir yöntem geliştirmiştir[51]. Bununla birlikte algoritmanın başarısı uygun parametre seçimine bağlıdır.

S. Jolfaei ve S. Abadi, meta-sezgisel bir yöntem olan Brado algoritmasını önermiştir[52]. Bu algoritmanın parçacık sürü optimizasyonu ve genetik algoritmalarla göre daha az çakışma ürettiği belirtilmiş, ancak parametre ayarlama sürecinde çeşitli zorluklar yaşandığı ifade edilmiştir. O. Moghimi ve A. Amini, çatışmaları dinamik olarak tespit eden ve sıralı olmayan çözüm yaklaşımı sunan NSCR algoritmasını geliştirmiştir[53]. Algoritmanın zaman karmaşıklığının bazı durumlarda $O(n^3)$ seviyesine ulaşabildiği belirtilmiştir.

B. Karabulut ve arkadaşları, doğrusal zamanda çalışan kural tabanlı bir algoritma geliştirmiştir[54]. Yöntem önceden tanımlanmış desenlere dayandığından arama işlemi gerçekleştirmemekte ancak yalnızca tek bir çözüm üretebilmektedir. C. Zhang ve J. Ma, Monte Carlo benzetimleri kullanarak vezir probleminin geçerli çözüm sayısını hesaplamayı amaçlamıştır[55]. Yöntem yalnızca yaklaşık sonuçlar üretebilmektedir. Aravind K. ve arkadaşları, çaprazlama ve mutasyon işlemlerini içeren genetik algoritma varyasyonları kullanarak problemi çözmeye çalışmıştır[56]. Ancak yöntemin performansı parametre seçimine bağlıdır.

Yuxin D. ve arkadaşları, yapay sinir ağları tabanlı yüksek dereceli sinir ağı yaklaşımını kullanmıştır[57]. Nöronlar arasındaki bağlantılar ve uygunluk kontrolü yardımıyla vezir yerleşimleri gerçekleştirilmiştir. Ancak kesin çözüm garantisi bulunmamakta ve hesaplama yükü yüksek olmaktadır. R. Moghaddas ve arkadaşları, her değişkenin bir ajan tarafından kontrol edildiği çok etmenli bir sistem geliştirmiştir[58]. Sistem hızlı ve ölçeklenebilir çözümler üretebilmesine rağmen rastgele yapısı nedeniyle her çalışmada farklı sonuçlar verebilmektedir.

Santhosh G. S. ve arkadaşları, klasik geri izleme ve kaba kuvvet yöntemlerini kuantum algoritmalarıyla birleştiren yeni bir yaklaşım önermiştir[59]. Bu yöntemle çok sayıda çözüm hızlı biçimde denenebilmekte ve uygun olmayan yerleşimler otomatik olarak elenebilmektedir. Nagesha Shivappa, yapay zekâ tabanlı kısıt tatmin yaklaşımını klasik vezir problemine uyarlamış ve arama alanını daraltarak daha hızlı çözümler elde etmiştir[60]. S. Guha ve arkadaşları, karınca kolonisi optimizasyonundan esinlenen doğa temelli bir yöntem kullanarak problemi çözmüştür[61]. Bu yöntemle geleneksel yaklaşımlara kıyasla zaman karmaşıklığında önemli iyileştirmeler sağlanmıştır.

Z. Bouneb, paralel ve dağıtık hesaplama yöntemleri kapsamında problemi Kneser graf yapısına dönüştürmüş ve düğümleri ikili bit dizileri ile temsil etmiştir[62]. Böylece sıralı algoritmalarından daha hızlı çözümler elde edilmiştir. R. Sasic ve Jun Gu, problemi çözmek için yerel, olasılıksal ve hızlı arama algoritmaları önermiştir[63]. Vezir sayısı oldukça büyük olduğunda dahi çözüme ulaşabilmişlerdir. Ancak farklı başlangıç dizilimleri farklı sonuçlar üretebilmektedir. A. Dehghani ve arkadaşları, Henry V adı verilen doğrulanabilir deterministik bir algoritma geliştirmiştir[64]. Bu yöntem, beş farklı desen kullanarak herhangi bir boyuttaki vezir problemini arama yapmadan çözebilmektedir. J. Heris ve M. Oskoei, genetik algoritma üzerinde çeşitli iyileştirmeler gerçekleştirerek daha hızlı ve kararlı sonuçlar elde etmeyi amaçlamıştır[65]. Çalışmada yerel arama yöntemleriyle daha başarılı sonuçlar elde edilebileceği belirtilmiştir.

O. Kashif ve arkadaşları, genetik algoritmayı genişlik öncelikli arama (BFS) ve derinlik öncelikli arama (DFS) yöntemleriyle karşılaştırmıştır[66]. Sonuçlar, vezir sayısının sekiz veya daha az olduğu durumlarda klasik yöntemlerin daha hızlı çalıştığını göstermiştir. B. Al-Khateeb ve W. Tareq, genetik algoritma ile benzetimli tavlama yöntemlerini karşılaştırmıştır[66]. Benzetimli tavlama yönteminin çözüme yaklaşık yarı sayıda yineleme ile ulaştığı görülmüştür. A. Farhan ve arkadaşları, genetik algoritma kullanarak arama uzayını daraltmayı ve daha hızlı çözümler üretmeyi amaçlamıştır[67]. Çalışmada yalnızca tek bir çözüm değil, mümkün olan tüm çözümler elde edilmeye çalışılmıştır.

3. ÖNERİLEN YÖNTEM ve PROBLEMLER

3.1 Malatya Merkezilik (MC) Algoritması

Graf teorisinde merkezilik ölçüleri, bir düğümün ağ içerisindeki önemini ve etkisini belirlemek amacıyla kullanılmaktadır. Derece merkeziliği, yakınlık merkeziliği, arasındalık merkeziliği ve özvektör merkeziliği gibi klasik merkezilik ölçütleri, düğümlerin ağ içerisindeki konumlarını farklı bakış açılarıyla değerlendirmektedir. Ancak özellikle kısıt yoğunluğu yüksek problemlerde, düğümlerin yalnızca kendi bağlantı sayılarının değil, aynı zamanda komşularının yapısal özelliklerinin de dikkate alınması önem taşımaktadır.

MC algoritması, bir düğümün önem derecesini belirlerken düğüm ile komşuları arasındaki ilişkiyi birlikte değerlendiren merkezilik temelli bir yaklaşımdır. Algoritma, her düğüm için komşu düğümlerin derece bilgilerini kullanarak bir merkezilik değeri hesaplamakta ve böylece ağ içerisindeki önemini belirlemektedir. Elde edilen merkezilik değerleri, düğümlerin çözüm sürecindeki öncelik sıralamasının oluşturulmasında kullanılmaktadır.

Bu tez kapsamında MC algoritması, Sudoku ve Sekiz-Vezir problemlerinin graf yapısı olarak modellenmesinden sonra düğümlerin önem düzeylerini belirlemek amacıyla kullanılmıştır. Hesaplanan merkezilik değerleri yardımıyla düğümler büyükten küçüğe doğru sıralanmış ve graf renklendirme işlemi bu sıralamaya göre gerçekleştirilmiştir. Böylece renklendirme sürecinin daha sistematik bir şekilde yürütülmesi ve çözüm sırasında oluşabilecek kısıt ihlallerinin azaltılması amaçlanmıştır.

Önerilen yaklaşımda algoritması, yalnızca bir düğüm sıralama mekanizması olarak değil, aynı zamanda graf renklendirme sürecini yönlendiren temel karar mekanizması olarak görev yapmaktadır. Bu yönüyle algoritma, kısıt memnuniyet problemlerinin çözümünde düğüm önceliklendirmesine dayalı alternatif bir yaklaşım sunmaktadır. Bu bölümün devamında MC algoritmasının matematiksel tanımı verilmekte ve algoritmanın çalışma adımları ayrıntılı olarak açıklanmaktadır.

MC değerinin hesaplanma yaklaşımı Denklem (1)'de verildi. Denklemde graftaki her düğüm için MC değeri tek tek hesaplanmaktadır. Denklemde $\psi(uk)$, uk düğümünün merkeziliğini ifade eden fonksiyon, uk incelenen düğüm, $N(uk)$ incelenen düğümün komşuları, $d(uk)$, uk düğümün derecesini, $d(um)$, uk düğümünün komşusu olan um düğümünün derecesini ifade etmektedir.

$$\psi(uk) = \sum_{\forall um \in N(uk)} \frac{d(um)}{d(uk)} \quad (1)$$

Tablo1 MC algoritmasının matematiksel ifadesini içermektedir.

Girdi : $G = (V, E)$ // V düğümlü ve E kenarlı G grafiği

Çıktı : G 'deki her bir düğüm için renk ataması

C : Renk ataması yapılan düğüm kümesi

$umax$: En yüksek merkezilik değerine sahip düğüm

Tablo 1. MC Algoritmasının Matematiksel İfadesi

```
1.  $C \leftarrow \emptyset$ 
2. While  $V \neq \emptyset$  do
3.   For each  $uk \in V$  do
4.      $\psi(uk) = \sum_{um \in N(uk)} \frac{d(um)}{d(uk)}$ 
5.   End for
6.    $umax \leftarrow \arg \max (\psi(uk))$ 
7.   Assign a color to  $umax$ 
8.    $C \leftarrow C \cup \{ \text{Assign a color to } umax \}$ 
9.    $V \leftarrow V - \{ umax \}$ 
10.   $E \leftarrow E - \{ (umax, um) \mid um \in N(umax) \}$ 
11. End While
12. Output  $G$ 
```

3.2 Sudoku Çözümü

Bu çalışmada, Sudoku probleminin çözümüne yönelik olarak geri izleme (backtracking) ve kaba kuvvet (brute-force) gibi geleneksel yöntemlerden farklı şekilde, graf teorisi, merkezilik ölçüleri ve graf renklendirme yaklaşımlarını bütünleştiren yenilikçi bir yöntem önerilmektedir. Önerilen yöntemde ilk olarak Sudoku tahtası graf yapısına dönüştürülmekte ve her bir hücre bir düğüm olarak temsil edilmektedir. Düğümler arasındaki bağlantılar ise Sudoku kurallarına dayalı olarak oluşturulmaktadır[32]. Oluşturulan graf yapısında her hücre; bulunduğu satır, sütun ve alt matris içerisindeki diğer hücrelerle komşu olacak şekilde tanımlanmıştır. Daha sonra graf üzerindeki düğümlerin merkezilik değerleri MC algoritması kullanılarak hesaplanmış ve renklendirme işlemi bu değerlere göre gerçekleştirilmiştir[13].

Önerilen yöntemde kullanılan renk sayısı, Sudoku probleminin kromatik sayısını ifade etmektedir. Standart 9×9 Sudoku problemleri için kromatik sayı 9 olarak belirlenmiştir. Bu değer, komşu düğümler arasında renk tekrarı olmaksızın kullanılabilecek minimum renk sayısını göstermektedir[68]. Graf üzerindeki tüm düğümler uygun şekilde renklendirildikten sonra her renge karşılık gelen sayısal değer atanarak Sudoku çözümü elde edilmektedir. Eğer problemde başlangıçta atanmış sayılar mevcut ise, bu değerler farklı sayıların farklı renklerle temsil edilmesini sağlayacak şekilde önceden renklendirilmektedir.

Sudoku tahtasının graf yapısına dönüştürülmesi sırasında aşağıdaki kurallar uygulanmıştır:

- Sudoku tahtasında bulunan her hücre bir düğüm olarak modellenmiştir.

- Satır, sütun ve alt matris kısıtları düğümler arasındaki kenar bağlantıları ile temsil edilmiştir.
- MC algoritması ile hesaplanan merkezilik değerleri, graf renklendirme sürecindeki düğüm önceliklerinin belirlenmesinde kullanılmıştır.

Önerilen yöntemin genel yapısı ve çalışma prensibi Şekil 1’de sunulmaktadır. Yöntem genel olarak altı temel aşamadan oluşmaktadır. Bu aşamalar aşağıda sırasıyla açıklanmıştır.

Aşama 1: Algoritmaya giriş olarak verilen Sudoku tahtası oluşturulur ve çözüm sürecinde kullanılacak satır, sütun ve alt matris ilişkileri belirlenir.

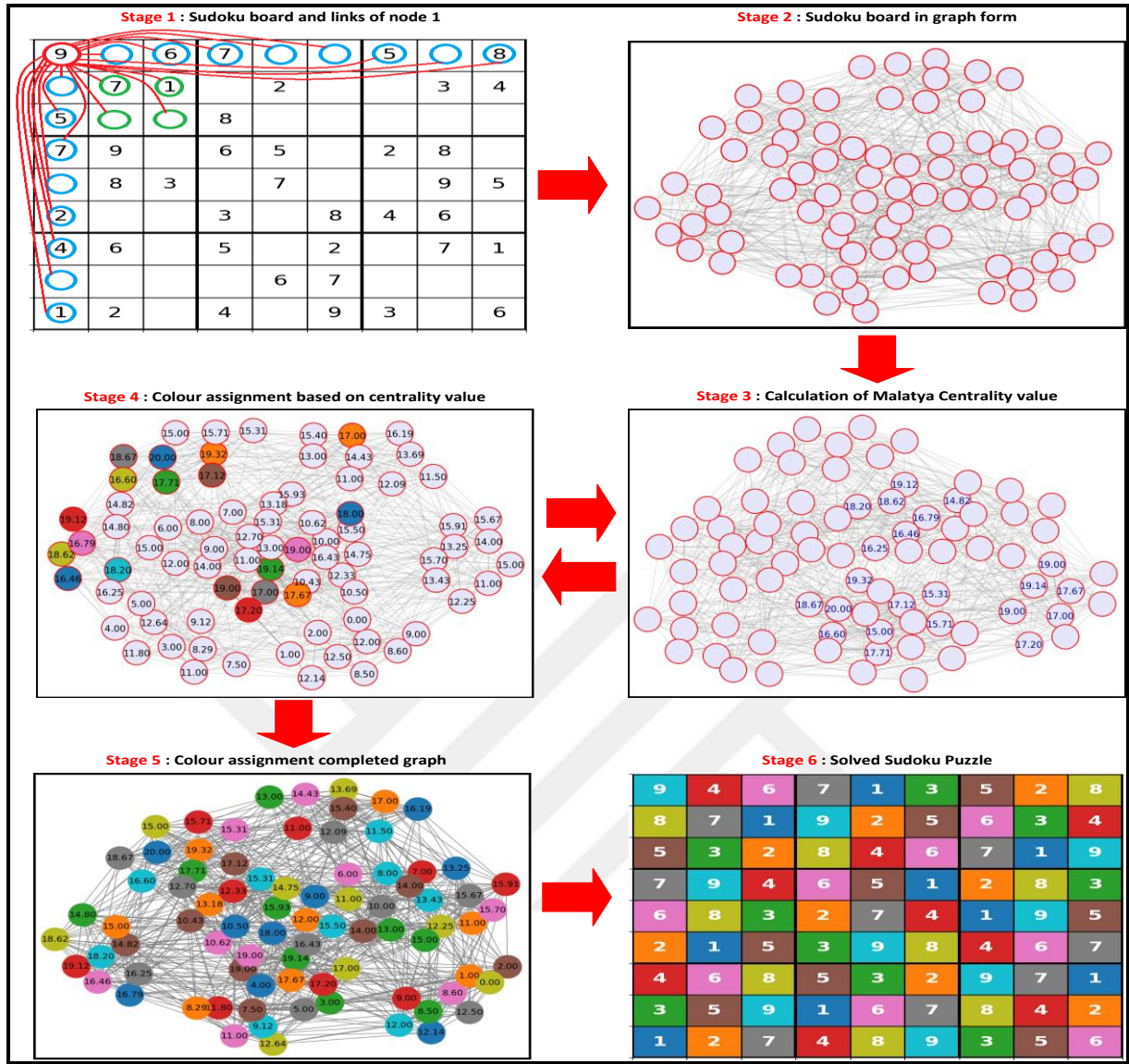
Aşama 2: Sudoku tahtasını temsil eden graf yapısı oluşturulur. Bu aşamada Sudoku hücreleri düğüm olarak tanımlanırken, satır, sütun ve alt matris kısıtları kenar bağlantıları şeklinde modellenir.

Aşama 3: Oluşturulan graf üzerinde MC algoritması uygulanarak tüm düğümlerin merkezilik değerleri hesaplanır.

Aşama 4: En yüksek merkezilik değerine sahip düğümden başlanarak uygun renk ataması gerçekleştirilir. Renklendirilen düğüm ve ilgili bağlantıları graf yapısından kaldırılır. Graf üzerinde işlenmemiş düğüm bulunması durumunda merkezilik değerleri yeniden hesaplanarak işlem tekrarlanır.

Aşama 5: Tüm düğümlerin renklendirilmesi tamamlanır ve Sudoku problemine ait tam renklendirilmiş graf yapısı elde edilir.

Aşama 6: Elde edilen renklendirilmiş graf yapısı tekrar Sudoku tahtasına dönüştürülür. Her renk belirli bir sayısal değeri temsil edecek şekilde eşleştirilir ve böylece Sudoku çözümü elde edilir.



Şekil 1. Önerilen Metodolojinin Grafiksel Özeti

Algoritmanın ilk aşamasında Sudoku problemine ait graf yapısı oluşturulmaktadır. Daha sonra merkezilik hesaplamaları sırasında düğüm ve bağlantılarının sistemden kaldırılması nedeniyle grafın bir kopyası oluşturulmakta ve hesaplamalar bu kopya graf üzerinde gerçekleştirilmektedir. Hesaplanan MC değerlerine göre düğümler büyükten küçüğe sıralanmakta ve son aşamada her düğüm için komşuluk ilişkileri dikkate alınarak kontrollü graf renklendirme işlemi uygulanmaktadır.

3.2.1 Shidoku Çözümü

Shidoku, klasik 9x9 Sudoku probleminin 4x4 boyutundaki özel bir versiyonudur. Toplam 16 hücreden oluşan bu yapıda, klasik Sudoku kuralları geçerliliğini korumaktadır. Problemin çözümünde 1, 2, 3 ve 4 sayıları kullanılmakta ve bu sayıların her satır, sütun ve 2x2 alt bölgede tekrar etmeyecek şekilde yerleştirilmesi gerekmektedir[69]. Shidoku problemi de Sudoku probleminde olduğu gibi MC

algoritması ve kontrollü graf renklendirme yaklaşımı kullanılarak çözülebilmektedir. Bu yapıda kullanılan dört farklı sayı, dört farklı renge karşılık gelmekte ve böylece problem dört renk kullanılarak çözülebilmektedir.

Önerilen yöntemin Shidoku problemi üzerindeki çalışma aşamaları aşağıda verilmiştir:

Aşama 1: Shidoku bulmacası oluşturulur ve örnek olarak 1 numaralı düğümün komşuluk ilişkileri gösterilir.

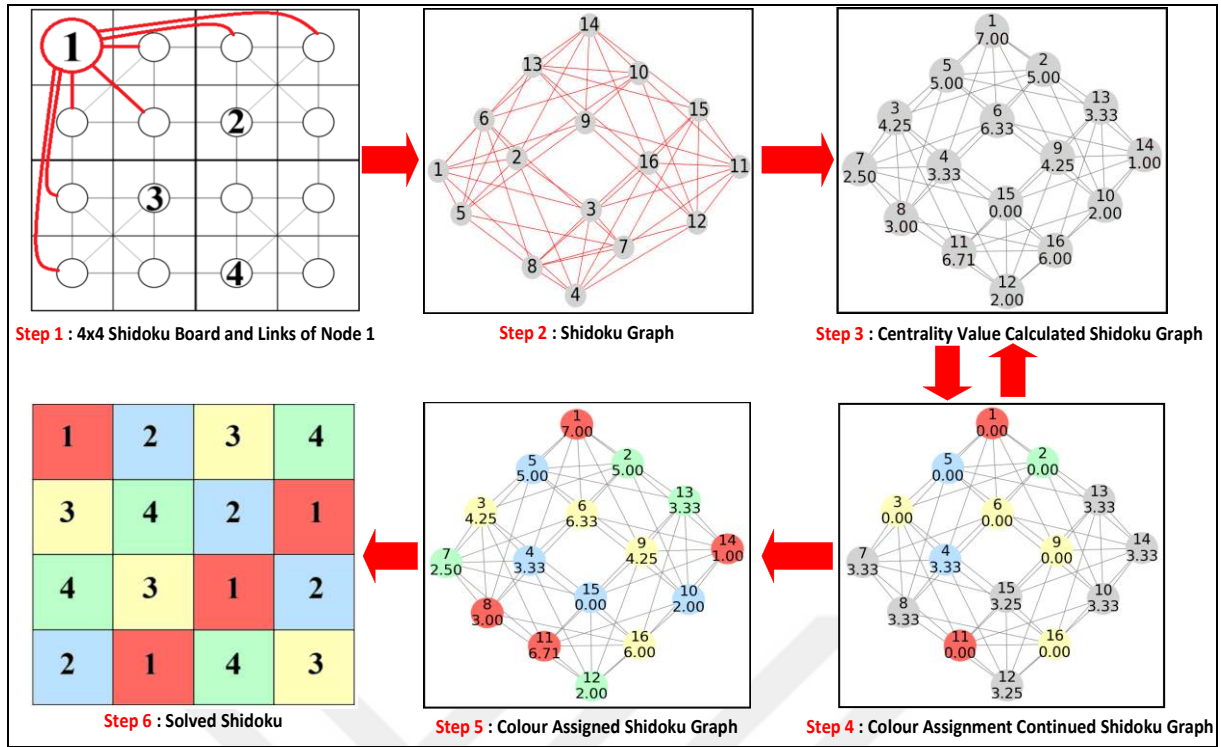
Aşama 2: Belirlenen komşuluk ilişkileri dikkate alınarak Shidoku problemi graf yapısına dönüştürülür. Oluşturulan graf yapısında düğümler arasındaki toplam kenar sayısı 56'dır.

Aşama 3: Graf üzerindeki tüm düğümler için MC değerleri hesaplanarak önem sıralaması elde edilir. Bu aşamada başlangıçta tüm düğümlerin merkezilik değerleri eşit olduğundan hesaplama ilk olarak en küçük indeks değerine sahip düğümden başlatılır. İlgili düğümün merkezilik değeri, komşu düğümlerin derece toplamının düğüm derece sine bölünmesiyle elde edilir. Hesaplama tamamlandıktan sonra düğüm ve ona bağlı kenarlar graf yapısından kaldırılır. Daha sonra aynı işlem sıradaki düğüm için tekrarlanır. Bu süreç tüm düğümlerin merkezilik değerleri hesaplanıncaya kadar devam eder.

Aşama 4: Tüm merkezilik değerleri elde edildikten sonra en yüksek merkezilik değerine sahip düğümden başlanarak kontrollü renklendirme işlemi gerçekleştirilir.

Aşama 5: Renklendirme işleminin tamamlanmasıyla birlikte Shidoku grafi merkezilik değerlerine göre uygun şekilde renklendirilmiş olur.

Aşama 6: Renklendirilmiş graf yapısı tekrar Shidoku bulmacasına dönüştürülerek problemin çözümü elde edilir.



Şekil 2. Bir Shidoku Bulmacasının Çözüm Adımları

Bu örnek, önerilen yöntemin yalnızca standart Sudoku problemleri için değil, daha küçük boyutlu Sudoku türevleri için de uygulanabilir olduğunu göstermektedir. Ayrıca graf teorisi ve graf renklendirme temelli bu yaklaşımın, daha büyük boyutlu ve daha karmaşık kısıt memnuniyet problemlerinin çözümünde de kullanılabileceği değerlendirilmektedir.

Şekil 2’de renklendirilen Shidoku grafına ait adım adım merkezilik hesaplamaları sonucunda elde edilen MC değerleri Tablo 2’de sunulmuştur. Tabloda her iterasyonda seçilen düğüm değeri kalın yazı ile gösterilmiştir. İlgili iterasyonda bu düğümlerin en yüksek MC değerine sahip olduğu, MC değerlerinin eşit olması durumunda ise en küçük indeks değerine sahip düğümün seçildiği görülmektedir. Merkezilik değeri hesaplandıktan sonra graf yapısından kenar bağlantılarıyla birlikte çıkarılan düğümler, sonraki iterasyonlarda “—” işareti ile gösterilmiştir. Tüm merkezilik değerleri hesaplandıktan sonra, en yüksek MC değerine sahip düğümden başlanarak kontrollü graf renklendirme işlemi gerçekleştirilmiş ve böylece Shidoku probleminin çözümü elde edilmiştir.

Tablo 2. Shidoku Bulmacasının Çözümünün Her Adımında Hesaplanan MC Değerleri

Iteration	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1st	7.00	7.00	7.00	7.00	7.00	7.00	7.00	7.00	7.00	7.00	7.00	7.00	7.00	7.00	7.00	7.00
2nd	—	6.33	6.67	6.67	6.33	6.67	6.43	6.43	6.67	6.43	6.71	6.71	6.67	6.43	6.71	6.71
3rd	—	6.00	6.20	6.00	6.00	6.33	6.00	5.86	6.20	6.00	—	6.00	6.00	5.86	6.00	6.33
4th	—	5.40	5.60	5.50	5.60	—	5.60	5.50	5.60	5.60	—	5.67	5.50	5.50	5.67	6.00
5th	—	5.00	5.00	5.00	5.00	—	5.00	5.00	5.00	5.00	—	5.00	5.00	5.00	5.00	—
6th	—	—	4.75	4.75	5.00	—	4.40	4.40	4.40	4.75	—	4.60	4.40	4.75	4.60	—
7th	—	—	4.25	4.25	—	—	4.25	4.25	4.25	4.25	—	4.20	4.25	4.25	4.20	—
8th	—	—	—	3.67	—	—	3.33	3.67	4.25	4.25	—	3.60	4.00	4.00	4.00	—
9th	—	—	—	3.33	—	—	3.33	3.33	—	3.33	—	3.25	3.33	3.33	3.25	—
10th	—	—	—	—	—	—	3.00	2.50	—	3.00	—	3.00	3.33	3.33	2.75	—
11th	—	—	—	—	—	—	2.50	2.50	—	2.50	—	2.33	—	2.50	2.33	—
12th	—	—	—	—	—	—	—	3.00	—	2.50	—	1.67	—	2.00	2.50	—
13th	—	—	—	—	—	—	—	—	—	2.00	—	2.00	—	2.00	2.00	—
14th	—	—	—	—	—	—	—	—	—	—	—	2.00	—	2.00	1.00	—
15th	—	—	—	—	—	—	—	—	—	—	—	—	—	1.00	1.00	—
16th	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0.00	—
Selected iteration	1	5	7	9	6	3	11	12	8	13	2	14	10	15	16	4
Coloured	7.00	5.00	4.25	3.33	5.00	6.33	2.50	3.00	4.25	2.00	6.71	2.00	3.33	1.00	0.00	6.00

3.2.2 Algoritma ve Akış Diyagramı

Önerilen yöntemin sözde kodu Tablo 3'te sunulmuştur. İlk olarak Sudoku problemi graf yapısına dönüştürülmüş ve düğümlerin merkezilik değerleri MC algoritması ile hesaplanmıştır. Hesaplanan merkezilik değerlerine göre düğümler büyükten küçüğe sıralanmış ve renklendirme sırası belirlenmiştir. Son aşamada ise en yüksek merkezilik değerine sahip düğümden başlanarak geri izleme (backtracking) destekli graf renklendirme işlemi uygulanmış ve Sudoku kısıtlarını sağlayan geçerli çözüm elde edilmiştir.

Tablo 3. Sudoku Çözümünde Önerilen Yöntemin Sözde Kodu

```

function SudokuGraphColoring(G)
    input:
        G - An undirected graph representing a Sudoku grid (V nodes, E edges)
    output:
        colored_graph - A properly colored Sudoku graph (9 colors, no adjacent conflicts)
    // Initialize variables
    N ← 9 // Standard Sudoku size (9x9 grid)
    H ← CopyGraph(G) // Working copy of the graph
    centrality_values ← // Stores centrality for each node
    color_map ← // Maps nodes to their assigned colors (1)-9)
    // Step 1: Compute centrality for each node
    for node in G.nodes() do
        centrality_values[node] ← ComputeCentrality(H, node)
    H.remove_node(node) // Remove processed node
    end for
    // Step 2: Sort nodes by centrality (descending)
    sorted_nodes ← SortNodesByCentrality(centrality_values)
    // Step 3: Assign colors using backtracking
    if not BacktrackColoring(G, sorted_nodes, color_map, 0) then
        raise Error("Graph is not 9-colorable (invalid Sudoku)")
    end if
    // Step 4: Apply colors to the original graph
    colored_graph ← ApplyColorMap(G, color_map)
    return colored_graph
end function
function ComputeCentrality(H, node)

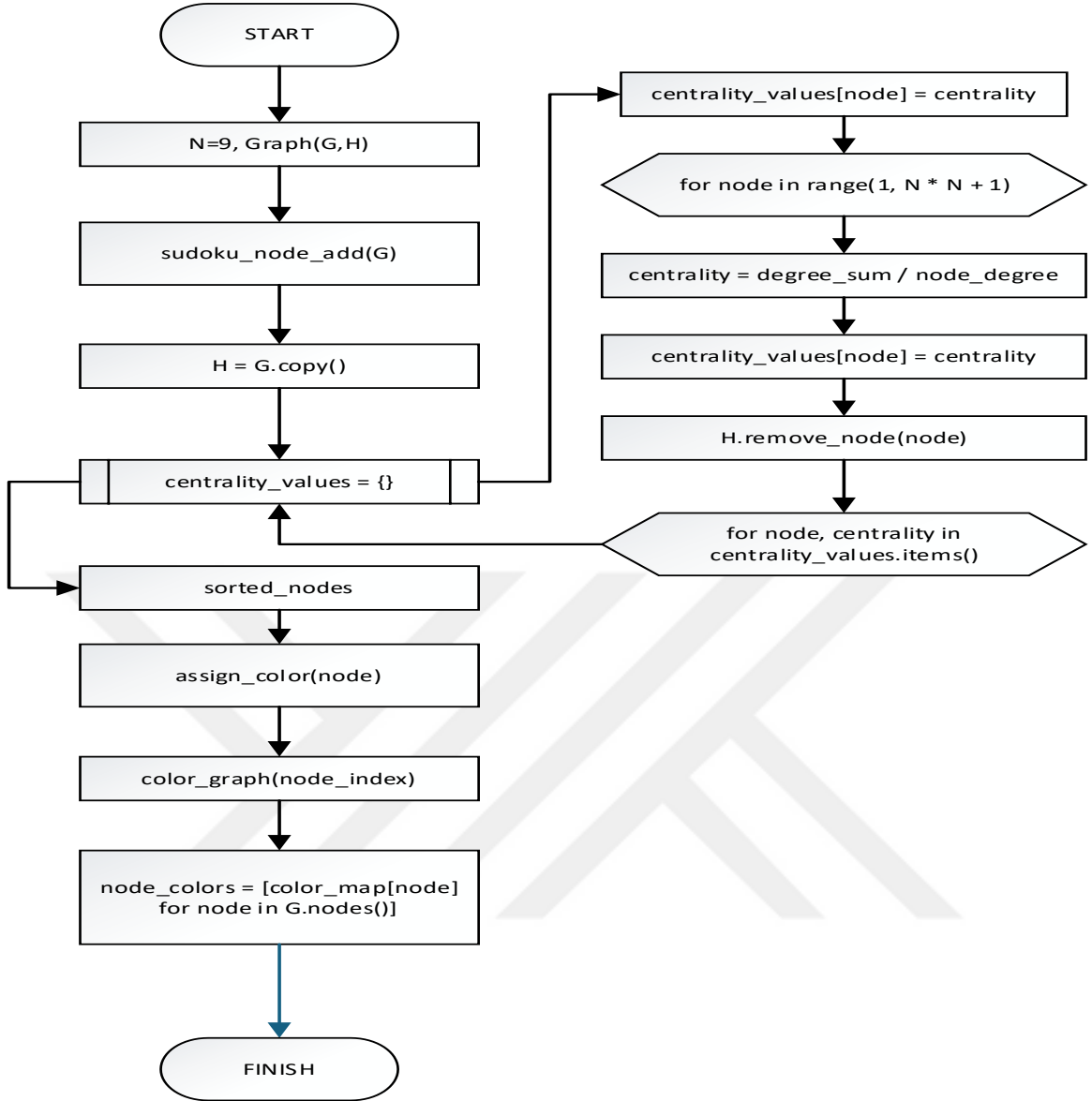
```

```

input:
  H - Subgraph of G
  node - Current node to analyze
output:
  centrality - Computed centrality value
neighbors ← Neighbors(H, node)
degree_sum ← 0
node_degree ← Degree(H, node)
if node_degree > 0 then
  for neighbor in neighbors do
    degree_sum ← degree_sum + Degree(H, neighbor)
  end for
  centrality ← degree_sum / node_degree
else
  centrality ← 0 // Avoid division by zero
end if
return centrality
end function
function SortNodesByCentrality(centrality_values)
input:
  centrality_values - Dictionary node: centrality
output:
  sorted_nodes - Nodes sorted by centrality (highest first)
return SortDescending(centrality_values)
end function
function BacktrackColoring(G, sorted_nodes, color_map, node_index)
input:
  G - Original graph
  sorted_nodes - Nodes in processing order
  color_map - Current color assignments
  node_index - Current node being processed
output:
  True if coloring is successful, False otherwise
// Base case: All nodes colored
if node_index == Length(sorted_nodes) then
  return True
end if
node ← sorted_nodes[node_index]
available_colors ← 1, 2, ..., 9 // Sudoku colors (1)-9)
// Remove colors used by neighbors
for neighbor in Neighbors(G, node) do
  if neighbor in color_map then
    available_colors.remove(color_map[neighbor])
  end if
end for
// Try each available color
for color in available_colors do
  color_map[node] ← color
  if BacktrackColoring(G, sorted_nodes, color_map, node_index + 1) then
    return True // Propagate success
  end if
color_map.remove(node) // Backtrack
end for
return False // No valid coloring found
end function

```

Şekil 3'te önerilen yöntemin akış diyagramı sunulmuştur. İlk aşamada Sudoku problemi graf yapısına dönüştürülmüş, ardından düğümlerin merkezilik değerleri hesaplanarak öncelik sıraları belirlenmiştir. Son olarak düğümler, merkezilik sıralamasına göre geri izleme destekli graf renklendirme algoritması ile renklendirilmiş ve Sudoku kısıtlarını sağlayan çözüm elde edilmiştir.



Şekil 3. Önerilen Yöntemin Akış Şeması

3.2.3 Veri Seti

Bu çalışmada, Bryan Park tarafından derlenen ve Kaggle platformunda yayımlanan 1.000.000 adet Sudoku bulmacasından oluşan veri seti kullanılmıştır[70]. Veri setinde her satırda bir adet çözülmemiş ve buna karşılık gelen bir adet çözülmüş Sudoku (9×9) bulmacası yer almaktadır. Deneysel çalışmalar kapsamında veri seti farklı büyüklüklerde alt kümelere ayrılmış ve 40.000, 200.000, 400.000 ve 1.000.000 bulmacadan oluşan veri grupları üzerinde değerlendirmeler gerçekleştirilmiştir.

Kullanılan veri seti farklı zorluk seviyelerine sahip Sudoku problemlerini içermektedir. Literatürde Sudoku problemlerinin zorluk dereceleri genellikle başlangıçta verilen ipucu (önceden atanmış hücre) sayısına göre belirlenmektedir. Buna göre 36 ve üzeri ipucu içeren bulmacalar kolay, 27–36 arasında ipucu içeren bulmacalar orta ve 17–26 arasında ipucu içeren bulmacalar ise zor seviyede sınıflandırılmaktadır [5]. Bununla birlikte Sudoku problemlerinin zorluk düzeyi yalnızca ipucu sayısına değil, bu

ipuçlarının tahta üzerindeki dağılımına da bağlı olabilmektedir. Ancak hesaplamalı çalışmalarda en yaygın kullanılan ölçüt, verilen ipucu sayısı olduğundan, bu çalışmada zorluk seviyelerinin belirlenmesinde doldurulmuş hücre sayısı esas alınmıştır.

Farklı zorluk seviyelerini temsil eden örnek Sudoku problemleri aşağıda sunulmuştur. Şekil 4'te 37 adet başlangıç değeri içeren kolay seviye bir Sudoku problemi, Şekil 5'te 30 adet başlangıç değeri içeren orta seviye bir Sudoku problemi ve Şekil 6'da ise 25 adet başlangıç değeri içeren zor seviye bir Sudoku problemi gösterilmektedir.

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

Şekil 4. Kolay Sudoku

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

Şekil 5. Orta Sudoku

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

Şekil 6. Zor Sudoku

3.3 Sekiz-Vezir Çözümü

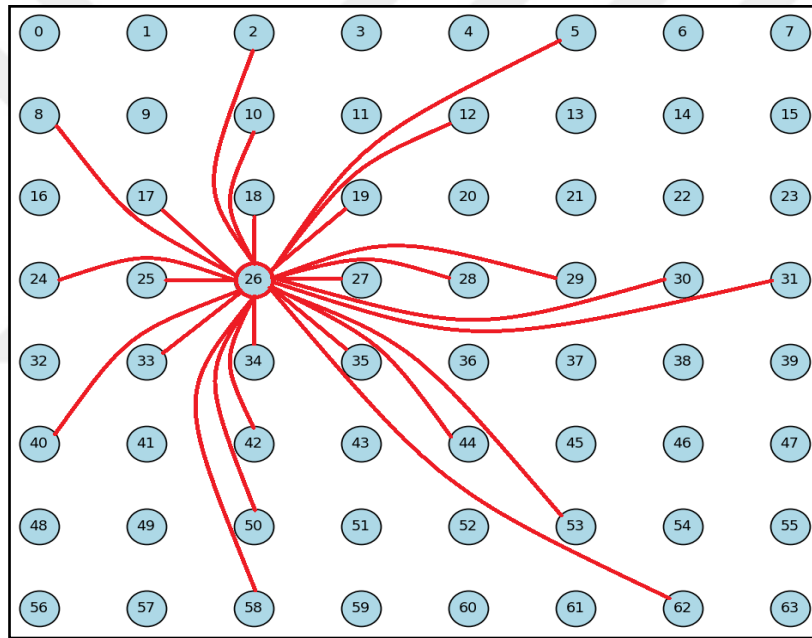
Bu çalışmada, sekiz-vezir probleminin çözümüne yönelik olarak merkezilik değerleri, graf teorisi ve graf renklendirme yaklaşımlarını bütünleştiren, literatürde yer alan geleneksel yöntemlerden farklı bir çözüm yöntemi önerilmektedir. Önerilen yöntemde ilk olarak satranç tahtası graf yapısına dönüştürülmekte ve tahtada bulunan her bir kare bir düğüm olarak temsil edilmektedir. Düğümler arasındaki bağlantılar ise vezirin hareket kuralları dikkate alınarak oluşturulmaktadır[2]. Oluşturulan graf yapısında her düğüm; bulunduğu satır, sütun ve köşegenlerde yer alan diğer düğümlerle bağlantılı olacak şekilde tanımlanmıştır. Daha sonra MC algoritması kullanılarak düğümlerin merkezilik değerleri hesaplanmış ve renklendirme işlemi bu değerlere göre gerçekleştirilmiştir[71].

Önerilen yöntemde kullanılan renk sayısı, satranç tahtasını temsil eden grafın kromatik sayısını ifade etmektedir. 8×8 boyutundaki standart satranç tahtası için kromatik sayı 8 olarak belirlenmiştir[68]. Sekiz-vezir probleminde toplam 92 geçerli çözüm bulunmakta olup, bu çözümler temelde 12 özgün çözümden elde edilmektedir. Diğer çözümler ise bu özgün çözümlerin döndürme ve simetrik yansıma işlemleri sonucunda oluşmaktadır[72]. Önerilen yöntemde çözüm süreci, düğümlerin merkezilik değerlerine göre sıralanması ve bu sıralama doğrultusunda kontrollü renklendirme yapılması esasına dayanmaktadır.

Problemin graf yapısına dönüştürülmesi sırasında aşağıdaki kurallar uygulanmıştır:

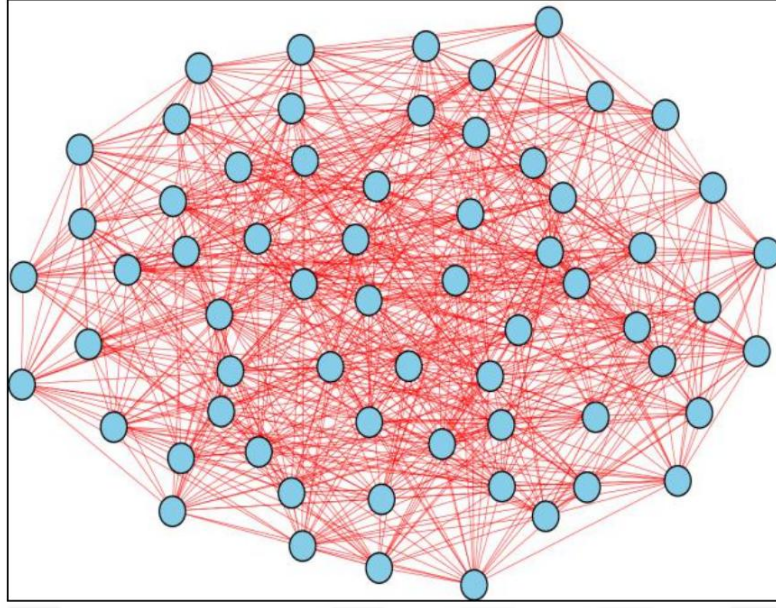
- Satranç tahtasında bulunan her kare bir düğüm olarak modellenmiştir.
- Satır, sütun ve köşegen kısıtları düğümler arasındaki kenar bağlantıları ile ifade edilmiştir.
- Oluşturulan grafın düğümlerine ait merkezilik değerleri MC algoritması ile hesaplanmış ve elde edilen değerler kullanılarak renklendirme öncelik sırası belirlenmiştir.

Önerilen yöntemin genel yapısı ve çalışma prensibini gösteren grafiksel gösterimler ilgili şekillerde sunulmuştur. Yöntem genel olarak dört temel aşamadan oluşmaktadır. Her aşamada gerçekleştirilen işlemler ve yöntemin işleyiş süreci aşağıda sırasıyla açıklanmaktadır. İlk aşamada satranç tahtası üzerindeki düğümler ve bu düğümler arasındaki kenar bağlantıları belirlenmektedir. Bu bağlantıların gösterimine örnek olarak, 26 numaralı düğümün sahip olduğu olası kenar bağlantıları Şekil 7'te sunulmuştur.



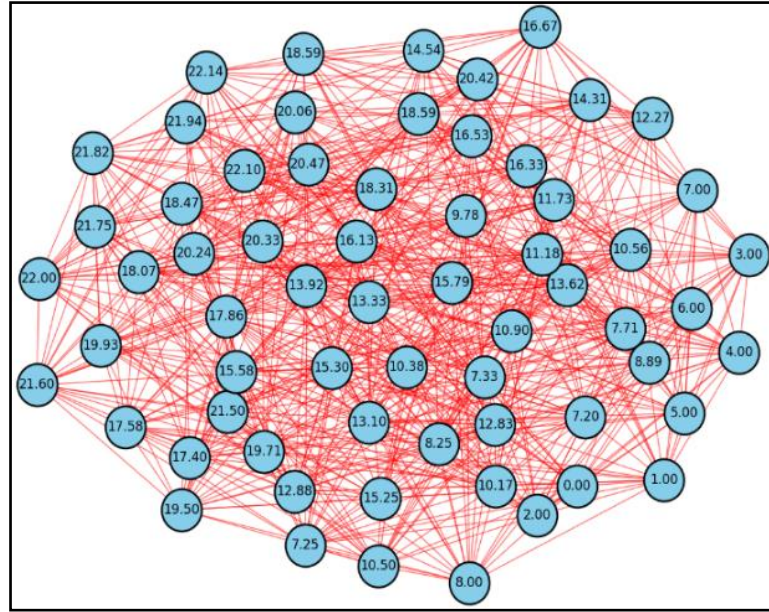
Şekil 7. 26. Düğüme Ait Mevcut Kenar Bağlantıları

Şekil 8'de oluşturulan graf yapısı gösterilmektedir. Bu aşamada satranç tahtasının tamamı, düğümler arasındaki kısıt ilişkileri dikkate alınarak graf yapısına dönüştürülmüştür.



Şekil 8. Oluşturulan Graf Yapısı

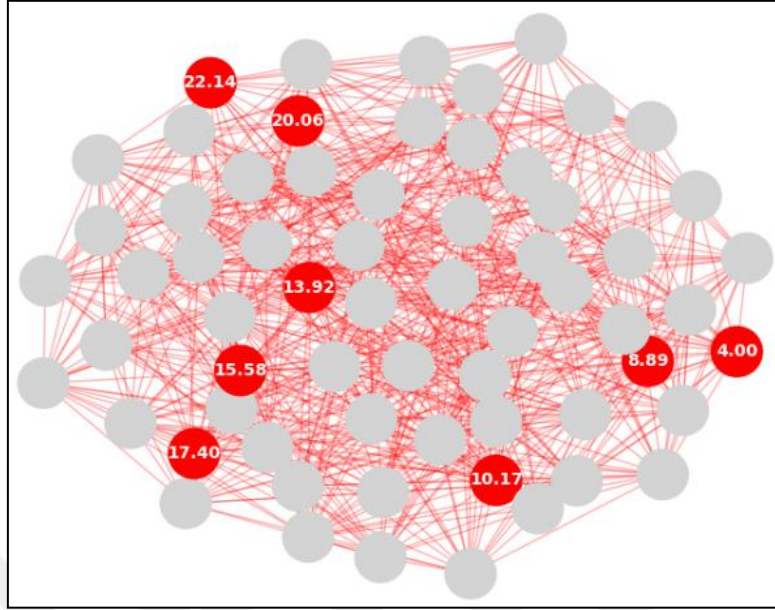
Graf yapısının oluşturulmasının ardından düğümlerin merkezilik hesaplamaları gerçekleştirilmiştir. Hesaplama sürecinde merkezilik değeri belirlenen düğümler ve bu düğümlere ait bağlantılar sistemden kaldırılmış, daha sonra kalan düğümler üzerinde hesaplama işlemi sürdürülmüştür. Bu işlem son düğüme kadar devam ettirilmiş ve tüm düğümlere ait MC değerleri elde edilmiştir. Hesaplanan merkezilik değerleri Şekil 9'da gösterilmektedir.



Şekil 9. MC Değerleri Hesaplanmış Graf Yapısı

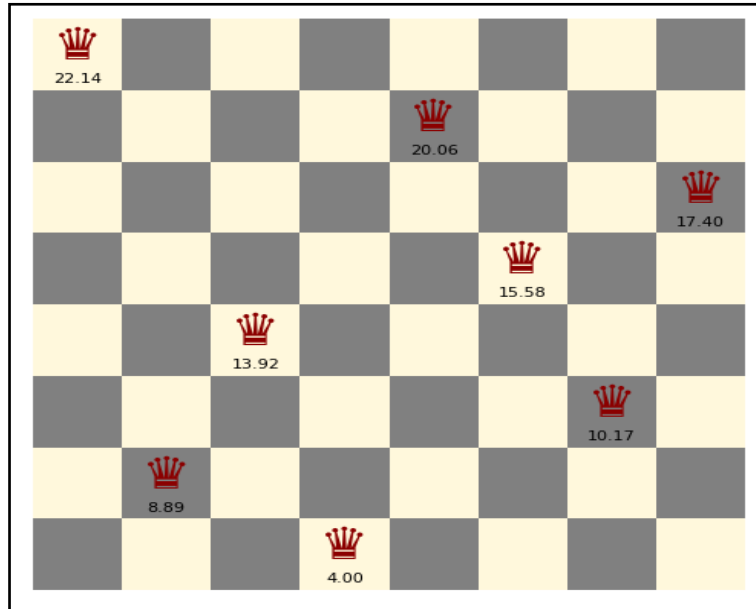
Çözüm aşamasında hesaplanan MC değerleri büyükten küçüğe doğru sıralanmış ve elde edilen önem sıralaması dikkate alınarak vezir yerleştirme işlemi kontrollü bir şekilde gerçekleştirilmiştir. Önerilen yöntemin en önemli özelliklerinden biri, merkezilik hesaplamaları sonucunda elde edilen düğüm

öncelik sıralamasının çözüm sürecini yönlendirmesidir. Bu şekilde elde edilen çözüm Şekil 10'da sunulmaktadır.



Şekil 10. MC Değerlerine Göre Renklendirilmiş (Çözülmüş) Graf Yapısı

Şekil 10'da görüldüğü üzere, vezirlerin yerleşimleri MC değerlerine dayalı olarak belirlenmiştir. Çözümün daha açık ve anlaşılır biçimde gösterilebilmesi amacıyla, aynı çözümün satranç tahtası üzerindeki gösterimi Şekil 11'de verilmiştir.



Şekil 11. MC Değerlerine Göre Renklendirilmiş Satranç Tahtası

4. DENEYSEL SONUÇLAR

Bu bölümde, önerilen MC algoritması tabanlı kontrollü graf renklendirme yaklaşımının performansına ilişkin deneysel sonuçlar sunulmaktadır. Çalışma kapsamında önerilen yöntem, hem Sudoku hem de Sekiz-Vezir problemleri üzerinde uygulanmış ve elde edilen sonuçlar farklı ölçütler açısından değerlendirilmiştir. Öncelikle Sudoku problemlerine ilişkin deneysel bulgular verilmiş, ardından yöntemin literatürde yer alan çalışmalarla ve yaygın olarak kullanılan graf renklendirme algoritmalarıyla karşılaştırmalı performans analizleri gerçekleştirilmiştir. Daha sonra önerilen yaklaşımın Sekiz-Vezir problemindeki başarımı incelenmiş ve elde edilen sonuçlar sunulmuştur. Böylece önerilen yöntemin farklı kısıt memnuniyet problemleri üzerindeki uygulanabilirliği, etkinliği ve performansı kapsamlı olarak değerlendirilmiştir.

4.1 Sudoku Problemine Ait Sonuçlar

Önerilen önceki yöntemlerde bazı optimizasyon problemleriyle karşılaşılmıştır. Veri setinde bulunan farklı zorluk seviyelerindeki Sudoku bulmacaları ya çözülememiş ya da çözümleri MC algoritmasına göre daha geç elde edilmiştir. Bu bilgiler ışığında, önerilen algoritmanın merkezilik değerleri, önem sıralaması ve renklendirme yoluyla çözüm üretmesi açısından özgün bir konumda olduğu görülmektedir. Önerilen yöntem, Sudoku problemini graf yapısına dönüştürerek her bir hücreyi bir düğüm olarak modellemektedir.

Bilinen klasik merkezilik ölçülerinden (özvektör, derece, uzaklık vb.) farklı olarak, yalnızca düğümün derecesi, komşu düğümlerinin derece değerleriyle ilişkili özel bir formül ve hesaplanan düğüm ile kenar bağlantılarına dayanan yeni ve etkili bir yaklaşım sunmaktadır. Bu hesaplama sonucunda elde edilen önem sıralaması, performans açısından etkili sonuçlar ortaya koymaktadır. Literatürde bulunan diğer merkezilik ölçülerine kıyasla daha pratik ve uygulamaya yönelik bir ölçüt sunduğu görülmektedir.

Bu çalışma, Sudoku bulmacasının yapısını graf teorisi çerçevesinde modellemeyi ve bu model üzerinde merkezilik tabanlı bir algoritmanın etkinliğini araştırmayı amaçlamaktadır. MC algoritması ile Sudoku bulmacası düğüm-kenar yapısına sahip bir graf olarak tanımlanmakta ve bu yapı üzerinde merkezilik hesaplamaları gerçekleştirilmektedir. Hesaplamalar sonrasında MC değerleri dikkate alınarak bu sıralamaya göre düğüm renklendirme işlemi uygulanmaktadır. Bu yöntem, bilinen merkezilik ölçülerinden farklı olarak bir önem sıralaması ortaya koymaktadır. Elde edilen bu önem sıralaması, önerilen yöntemin mevcut yöntemlerden ayrışmasını sağlamaktadır. Bu yönüyle çalışma yalnızca bir bulmaca çözüm yöntemi olmayıp, benzer yapısal kısıtlara sahip farklı graf tabanlı problemlerin çözümüne de katkı sağlamaktadır.

4.1.1 İlgili Çalışmalar ile Karşılaştırma

Aşağıdaki tabloda, belirli Sudoku veri kümeleri üzerinde bazı çözüm yöntemlerinin uygulanmasıyla elde edilen sonuçlar gösterilmektedir. Tablo 2, farklı boyut ve zorluk seviyelerindeki Sudoku bulmacalarının çözümüne yönelik geliştirilen veya test edilen çalışmaların başarı oranlarının karşılaştırmasını sunmaktadır. Geleneksel yaklaşımlar gibi geri izleme (backtracking), genetik algoritma ve benzetilmiş tavlama kombinasyonları, kolay seviyedeki bulmacalarda yaklaşık %100 başarı oranına ulaşırken, zor seviyedeki bulmacalarda başarı oranı %90 seviyelerine düşmektedir[15].

Tablo 4. Farklı Zorluklardaki Sudoku Bulmacalarının Algoritma Başarı Oranları Karşılaştırılması

Method(s)	Dataset Size			Success Rate		
	<500	≤40.000	1.00.000	Easy	Medium	Difficult
Backtracking, Genetic Algorithm, Simulated Annealing [15]	✓	—	—	%100	%98	%90
Graph Referencing Algorithm, Genetic Algorithm, Simulated Annealing, Harmony Search, Rough Force [10]	✓	—	—	%95-98	%90-95	%80-90
Sparse Optimisation (L1) [23]	—	✓	—	%99+	%99+	%84.8- %99+
Douglas-Rachford Algorithm, Binary Formulation, Graph Colouring [2]	✓	—	—	%90	%90	%90
Irregular Learning Cellular Automata (ILCA) [73]	✓	—	—	%99.9	%96.7	%58
Sparse Optimisation, Integer Linear Programming, Entropy Minimisation [24]	—	✓	—	%100	%79.5	%25
with Node Based Coincidence Algorithm [27]	✓	—	—	%36	%29.5	%26
General variable neighborhood search for solving Sudoku puzzles: unfiltered and filtered models [74]	✓	—	—	%100	%100	%95
Particle Swarm Optimization incorporated within the framework of cell-like P-systems [9]	✓	—	—	%100	%80-90	%50
Exact Method for Generating Strategy-Solvable Sudoku Clues [75]	—	✓	—	%100	%80-100	%40-50
Constraint Propagation Using Naked Pair [76]	—	—	✓	%100	%95	%92
Blended Glial Cell's Spiking Neural Network (BGSNN) [77]	—	—	✓	%100	%99	%97
Solving Sudoku using Boolean Algebra[78]	✓	—	—	%100	%98-100	%95-98
MC Algoritması [13]	✓	✓	✓	%100	%100	%100

Graf tabanlı referanslama ve optimizasyon temelli yöntemler, özellikle orta seviyedeki bulmacalarda %90-95 başarı oranı gösterirken, zorluk seviyesi arttıkça başarı oranları yaklaşık %80'e kadar düşmektedir[10]. Seyrek optimizasyon (L1) ve Douglas-Rachford algoritması gibi doğrusal programlama tabanlı yöntemler genel olarak yüksek doğruluk oranları üretmekle birlikte, zor seviyedeki bulmacalarda başarı oranları %85'in altına düşebilmektedir[2], [23]. Hücresel otomata ve zar algoritmaları kolay seviyedeki bulmacalarda etkili olmalarına rağmen, zor seviyelerde başarı oranları %30 seviyelerine kadar gerileyebilmektedir[9], [73]. Daha güncel yaklaşımlar arasında yer alan Blended Glial Cell's Spiking Neural Network gibi yapay sinir ağı tabanlı modeller, zor seviyedeki bulmacalarda %97'ye kadar başarı sağlayabilmektedir[77].

Önerilen yöntem olan MC algoritması ise düğüm merkeziliğine dayalı graf renklendirme stratejisi sayesinde tüm zorluk seviyelerinde %100 başarı oranına ulaşmış ve hem küçük hem de büyük ölçekli veri kümelerinde tutarlı sonuçlar sağlamıştır. Bu açıdan önerilen yöntem, zorluk seviyesinden bağımsız olarak sağladığı yüksek doğruluk oranı ve kararlılık sayesinde mevcut yöntemlere kıyasla önemli bir teknik üstünlük göstermektedir.

4.1.2 Graf Renklendirme Algoritmaları İle Karşılaştırma

MC algoritması, büyük Sudoku veri kümeleri üzerinde test edilmiştir. Bu testlerde MC algoritması, daha önce Sudoku çözümünde etkili olduğu kanıtlanmış bazı algoritmalar ile karşılaştırılmıştır. MC algoritması; 40.000, 200.000 ve 400.000 Sudoku bulmacası içeren veri kümelerinin yanı sıra farklı zorluk seviyelerine (kolay, orta ve zor) sahip veri kümeleri üzerinde Greedy, Brute Force, RLF, Douglas-Rachford, DSATUR ve Welsh-Powell algoritmaları ile karşılaştırılmıştır.

Deneyler, 13. nesil Intel® Core™ i5-13500H işlemci (2.60 GHz), NVIDIA GeForce RTX 4050 (8 GB), 16 GB RAM ve 500 GB SSD bileşenlerinden oluşan bir sistem üzerinde gerçekleştirilmiştir. Ayrıca, algoritmaların zaman performanslarını ölçmek amacıyla Python tabanlı kodlama uygulanmış ve ölçümlerde tek çekirdek kullanımı esas alınmıştır.

4.1.3 Karmaşıklık Analizi

Çalışmada karşılaştırılan RLF, Douglas–Rachford, Welsh–Powell, DSATUR, Greedy ve Brute Force algoritmaları, Sudoku problemi için kendi zaman karmaşıklıklarına uygun maliyetler üretmektedir. Önerilen yöntem ise Sudoku grafinin oluşturulması, düğümlerin merkezilik değerlerinin yalnızca bir kez hesaplanması ve düğümlerin bu değerlere göre sıralanması gibi avantajlar sunmaktadır. Sudoku çözüm sürecinde renklendirme işlemi aynı düğüm sıralaması kullanılarak gerçekleştirilmektedir. Böylece merkezilik hesaplama maliyeti tek bir işlemle sınırlandırılmakta ve sonraki problemlerde tekrar ortaya çıkmamaktadır. Bu özellik, yöntemi özellikle büyük veri kümeleri üzerinde avantajlı hâle getirmektedir.

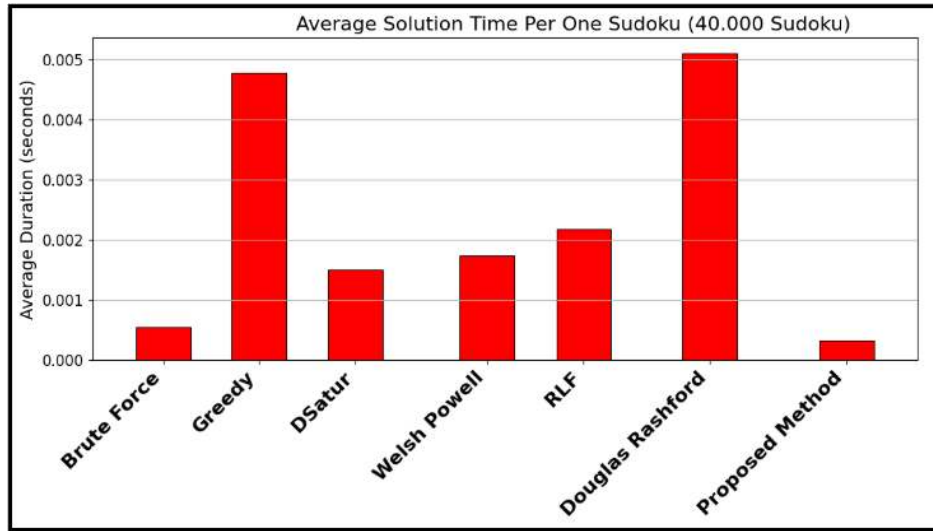
Teorik olarak yöntem, geri izleme (backtracking) temelli yapısı nedeniyle üstel bir karmaşıklığa sahiptir ve $O(2^n)$ karmaşıklık seviyesinde değerlendirilmektedir. Ancak merkezilik sıralamasının yalnızca bir kez gerçekleştirilmesi sayesinde, her problem için yeniden sıralama veya seçim maliyeti oluşturan diğer algoritmalara kıyasla daha verimli bir yapı sunmaktadır. Bu nedenle uygulama açısından değerlendirildiğinde karmaşıklığı $O(n+m)$ seviyesine indirgenmektedir. Tablo 6’ da yöntemlerin karmaşıklık analizi yer almaktadır.

Tablo 5. Zaman Karmaşıklığı Analizi

Algoritmalar	Teorik Zaman Karmaşıklığı
Greedy [34]	$O(n+m)$
Brute Force [16]	$O(k^n)$
RLF [34]	$O(n^3)$
Douglas-Rashford [1]	$O(T.C)$
DSatur [35]	$O(n^2)$
Welsh-Powell [5]	$O(n^2)$
MC [10]	$O(n+m)$

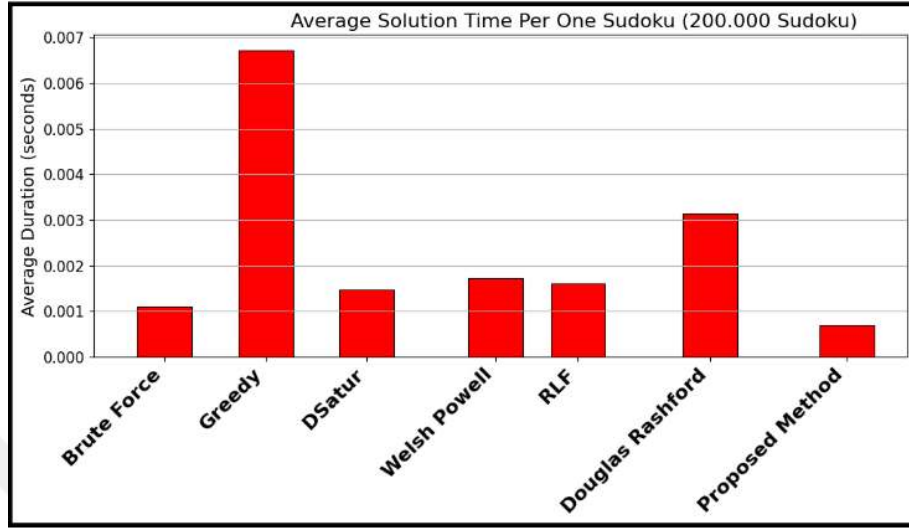
Bu yöntemlerin tamamı tüm bulmacaları başarıyla çözmüştür. Karşılaştırılan algoritmalar arasında MC algoritmasının zaman performansı açısından en iyi sonuçları verdiği tespit edilmiştir. Bu süreçlere ait ayrıntılı sonuç görselleştirmeleri aşağıda sunulmuştur.

Şekil 12’de, 40.000 Sudoku bulmacası içeren veri kümesi üzerinde Brute Force, Greedy, DSatur, Welsh Powell, RLF, Douglas Rashford algoritmaları ve önerilen MC algoritmasının ortalama çözüm performansları verilmiştir. Bir Sudoku bulmacasının ortalama çözüm süresi dikkate alındığında, Douglas Rashford algoritması 0,005 ms’nin üzerindeki değeriyle en kötü sonucu üretirken, 0,004 ms’nin üzerindeki süresiyle Greedy algoritması ikinci en yavaş sonucu elde etmiştir. Bunu, 0,002 ms’nin üzerindeki çözüm süresiyle RLF algoritması takip etmektedir. Welsh Powell ve DSatur algoritmaları 0,001 ms’nin üzerinde birbirine yakın sonuçlar üretirken, MC algoritmasının Brute Force algoritmasına göre daha kısa sürede çözüme ulaştığı ve en hızlı çözüm süresine sahip yöntem olduğu görülmektedir.

**Şekil 12.** 40.000 Sudoku Veri Kümesi Ortalama Çözüm Süreleri

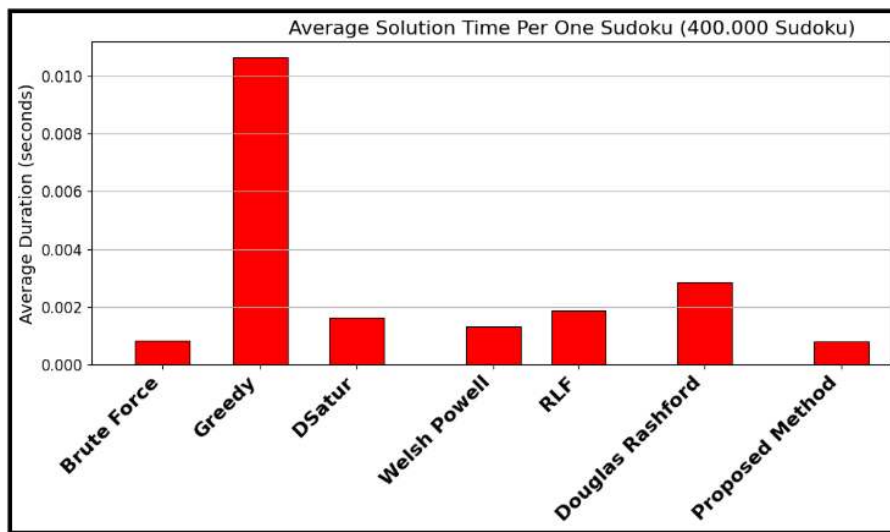
200.000 Sudoku bulmacası içeren veri kümesi üzerinde Brute Force, Greedy, DSatur, Welsh Powell, RLF, Douglas Rashford ve MC algoritmaları (her bir bulmaca için) ortalama çözüm süreleri açısından karşılaştırılmış ve Şekil 13’de gösterilen sonuçlar elde edilmiştir. Greedy algoritması 0,007

ms'nin üzerindeki süresiyle en kötü sonucu üretmiştir. Bunu, 0,003 ms'nin üzerindeki süresiyle Douglas Rashford algoritması takip etmiştir. Diğer tüm yöntemler 0,001 ms ile 0,002 ms aralığında sonuçlar üretirken, MC algoritması 0,001 ms'lik çözüm süresiyle en iyi sonucu elde etmiştir.



Şekil 13. 200.000 Sudoku Veri Kümesi Ortalama Çözüm Süreleri

400.000 Sudoku bulmacası içeren veri kümesi üzerinde Brute Force, Greedy, DSatur, Welsh Powell, RLF, Douglas Rashford algoritmaları ve önerilen MC algoritması (her bir bulmaca için) ortalama çözüm süreleri açısından karşılaştırılmış ve Şekil 14'te gösterilen sonuçlar elde edilmiştir. Greedy algoritması 0,010 ms'nin üzerindeki değeriyle en kötü sonucu üretmiştir. Ardından, Douglas Rashford algoritması 0,002 ms'lik çözüm süresiyle ikinci en düşük performansı göstermiştir. Diğer tüm yöntemler 0,001 ms'nin altında birbirine yakın sonuçlar üretirken, MC algoritması tüm yöntemler arasında en hızlı çözüm süresini elde etmiştir.

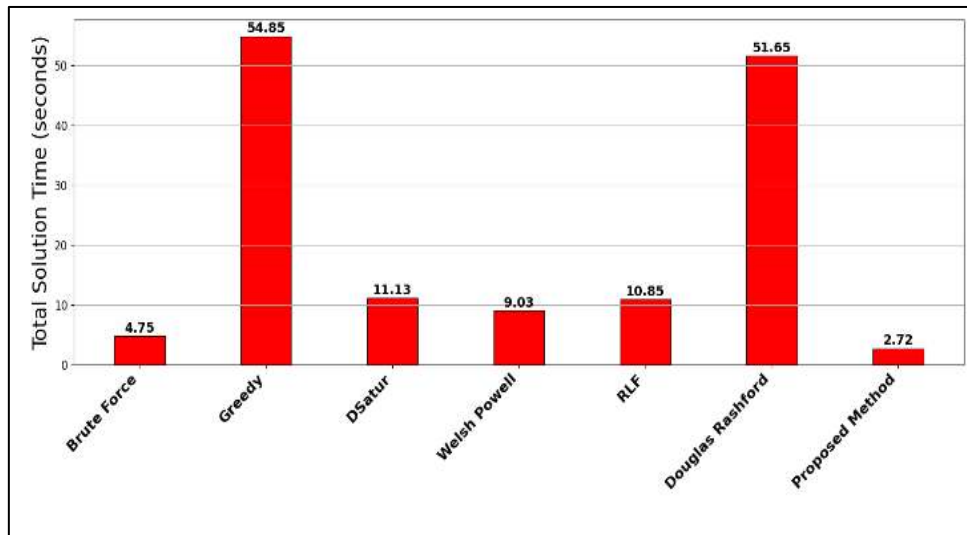


Şekil 14. 400.000 Sudoku Veri Kümesi Ortalama Çözüm Süreleri

Sudoku veri kümesi üzerinde gerçekleştirilen deneysel çalışmalar, genel ortalama performans ve farklı zorluk seviyeleri açısından değerlendirilmiştir. Bu amaçla, 1.000.000 Sudoku örneği sahip oldukları ipucu sayılarına göre üç gruba ayrılmıştır: Kolay (≥ 36 ipucu, 12.719 örnek), Orta (33–35 ipucu, 917.857 örnek) ve Zor (≤ 32 ipucu, 69.425 örnek). Önerilen yöntem, üç zorluk seviyesinin tamamında tutarlı ve başarılı sonuçlar elde etmiş, özellikle zor seviyelerde DSatur, Welsh Powell, Greedy, Brute Force, Douglas Rashford ve RLF gibi yöntemlere kıyasla üstün bir performans sergilemiştir.

Önerilen yöntem, 12.719 adet kolay seviyedeki Sudoku bulmacasından oluşan veri kümesi üzerinde toplam çözüm süreleri bakımından Brute Force, Greedy, DSatur, Welsh Powell, RLF ve Douglas Rashford algoritmaları ile karşılaştırılmıştır. Şekil 15 karşılaştırmalı analiz sonuçlarının görsel sunumunu göstermektedir. Greedy algoritması 54 saniye ile en kötü sonucu elde etmiştir. Douglas Rashford algoritması ise 51 saniye ile ikinci en kötü sonucu göstermiştir. Diğer yöntemler birbirine yakın sonuçlar üretirken, MC algoritması 2 saniyelik çözüm süresiyle en hızlı sonucu elde etmiştir.

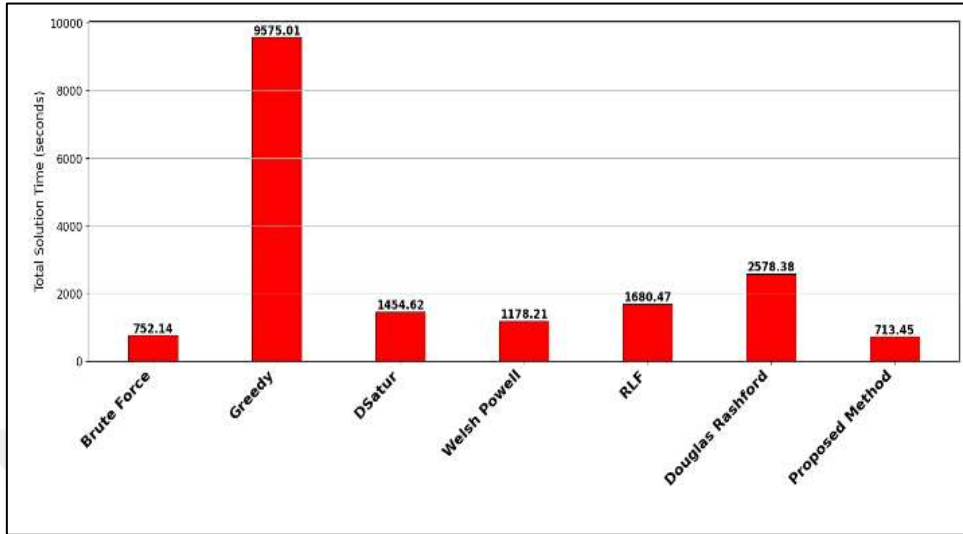
Performans değerlendirmesi, farklı boyutlara ve değişen karmaşıklık seviyelerine sahip veri kümeleri üzerinde gerçekleştirilmiştir. Ortalama çözüm süresi açısından başarı oranını ortaya koyarken, karmaşık veri kümeleri yöntemin gerçek gücünü daha etkili bir şekilde göstermektedir. 1.000.000 Sudoku bulmacası içeren veri kümesi; zor, orta ve kolay olmak üzere farklı zorluk seviyelerine ayrılmış ve performans sonuçları elde edilmiştir. Önerilen yöntem, düğümlerin çözüm sıralamasını optimize etmekte ve DSatur, Welsh-Powell, Greedy, Brute Force, Douglas-Rashford ve RLF gibi klasik karşılaştırma algoritmalarına göre daha etkili bir yapı sergilemektedir.



Şekil 15. Kolay Seviye Veri Kümesi Ortalama Çözüm Süreleri

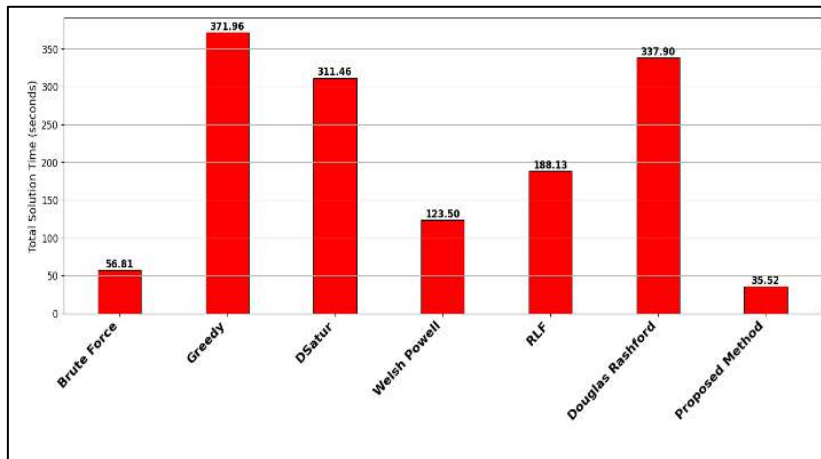
917.857 adet orta seviyedeki Sudoku bulmacası içeren veri seti üzerine Brute Force, Greedy, DSatur, Welsh Powell, RLF, Douglas Rashford algoritmaları ile Önerilen yöntem toplam çözüm süreleri bakımından karşılaştırıldı. Şekil 16 da bu karşılaştırmanın sonuç görseli verildi. Greedy algoritması 9575

saniye ile en kötü sonucu üretti. Ardından Douglas Rashford algoritması 2578 saniye ile ikinci kötü sonucu verdi. Diğer yöntemler benzer sonuçlar üretirken en hızlı sonucu 713 saniye MC algoritması verdi.



Şekil 16. Orta Seviye Veri Kümesi Ortalama Çözüm Süreleri

69.424 adet zor seviyedeki Sudoku bulmacası içeren veri seti üzerine Brute Force, Greedy, DSatur, Welsh Powell, RLF, Douglas Rashford algoritmaları ile Önerilen yöntem toplam çözüm süreleri bakımından karşılaştırıldı. Şekil 17’de bu karşılaştırmanın sonuç görseli verildi. Greedy algoritması 371 saniye ile en kötü sonucu üretti. Ardından Douglas Rashford algoritması 337 saniye ile ikinci kötü sonucu verdi. Diğer yöntemler benzer sonuçlar üretirken en hızlı sonucu 35 saniye MC algoritması verdi.



Şekil 17. Zor Seviye Veri Kümesi Ortalama Çözüm Süreleri

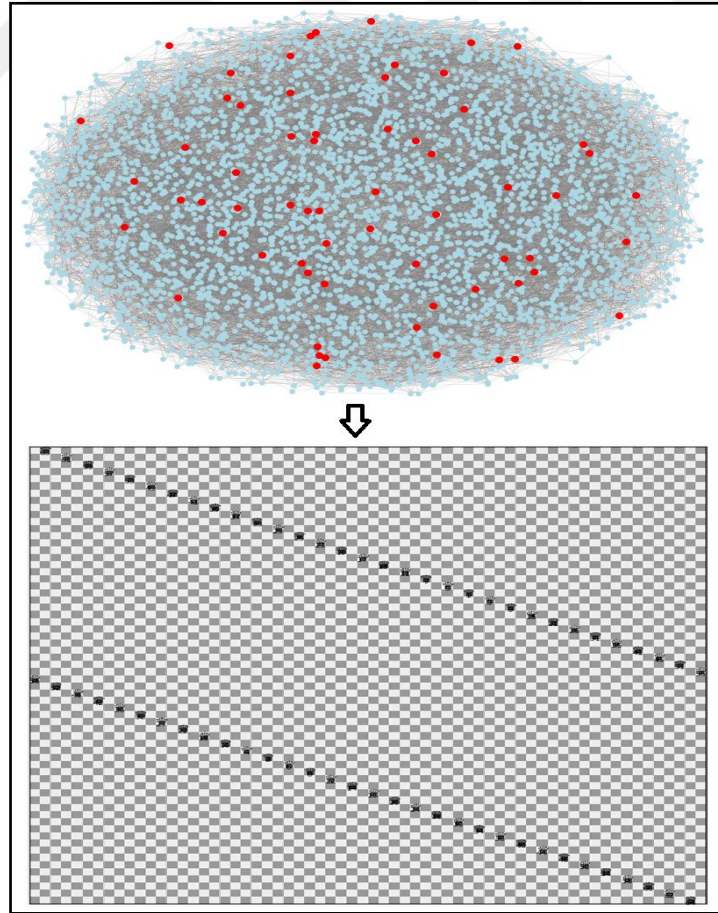
Çalışmaya eklenen grafikler, bu karşılaştırmayı açık biçimde ortaya koymakta ve yöntemin yalnızca genel başarı açısından değil, zorluk derecelerine göre de güçlü bir performans sergilediğini göstermektedir. Sudoku problemini çözmek kullanılan Brute Force, DSatur gibi algoritmalar optimize versiyonları ile uygulandı. Önerilen yöntem bu standart yöntemlerle adil ve güvenilir biçimde karşılaştırıldı.

Bu algoritmaların her biri, Sudoku graf yapısındaki düğümler (hücreler) ve kenarlar (satır, sütun ve kutu kısıtları) dikkate alınarak uyarlandı, böylece önerilen yöntemimiz yalnızca basit implementasyonlarla değil, alanda kabul gören standart yaklaşımlarla karşılaştırıldı. Elde edilen sonuçlar, yöntemin özellikle farklı zorluk seviyelerinde sudoku problemlerinde çözüm süresi ve verimlilik açısından sağladığı üstünlüğü göstermektedir.

4.2 Sekiz-Vezir Problemine Ait Sonuçlar

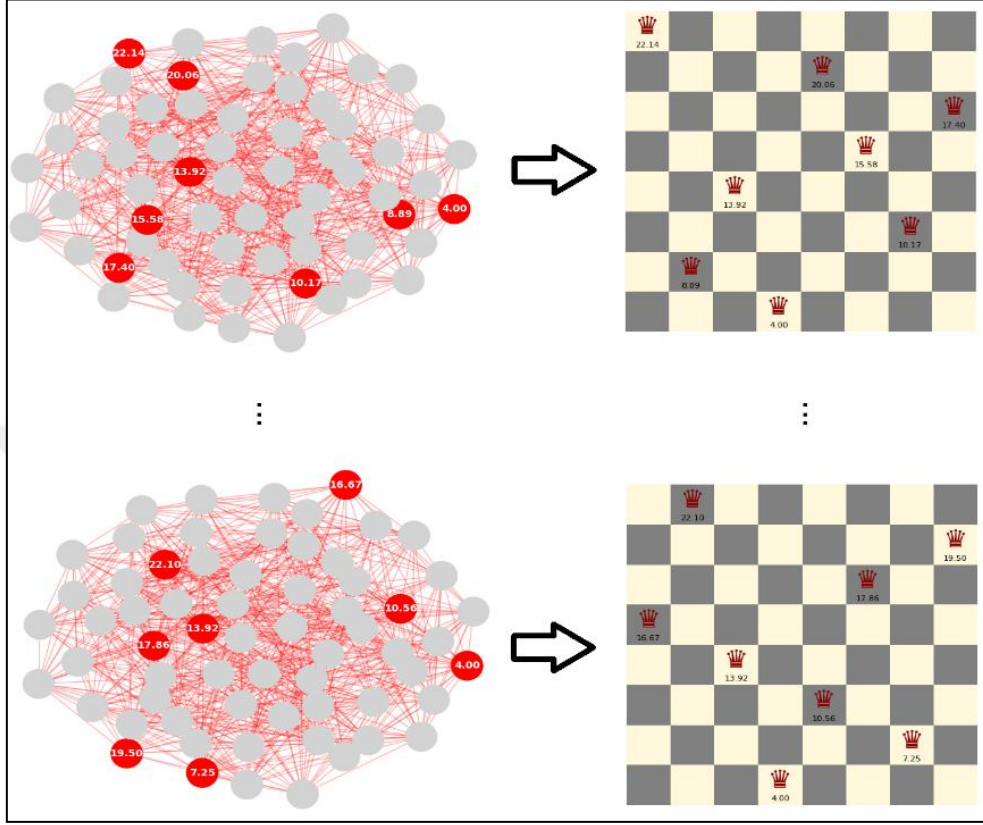
Bu çalışmada, sekiz vezir problemi grafik teorisi, merkezîyet hesaplamaları ve grafik renklendirme yöntemleri kullanılarak ele alınmıştır. Sekiz vezir problemi iki farklı bakış açısıyla incelenmiştir. Bunlar, bugüne kadar bilinen 12 farklı çözümü elde etmek ve problemin üstel olarak genişletilmiş bir versiyonu olan 64 vezir problemidir.

64 vezir problemi için grafik yapısı, 64x64'lük bir yapıya dönüştürüldü. Bu yapı 4.096 düğüm ve 428.736 kenar bağlantısına sahiptir. 64 vezir, birbirlerini tehdit etmeyecek şekilde sisteme yerleştirildi. Vezirlerin hareket kısıtlamalarını dikkate alan bir renklendirme süreci kullanılarak, klasik yöntemlerden farklı bir algoritmik ve görsel yapı oluşturuldu. Grafik ve satranç tahtası görselleştirmesi Şekil 18'de gösterilmiştir.



Şekil 18. 64 Vezir Probleminin Grafiği ve Yerleşimi

8 Vezir problemine 12 farklı çözüm bulunmaktadır. Bu çözümlerin tümü, önerilen yöntem kullanılarak elde edilmiştir. Çözümlerin örnekleri olarak, grafiğin ve satranç tahtası görünümünün daha net anlaşılması için Şekil 7’de ilk ve son çözümler gösterilmiştir.

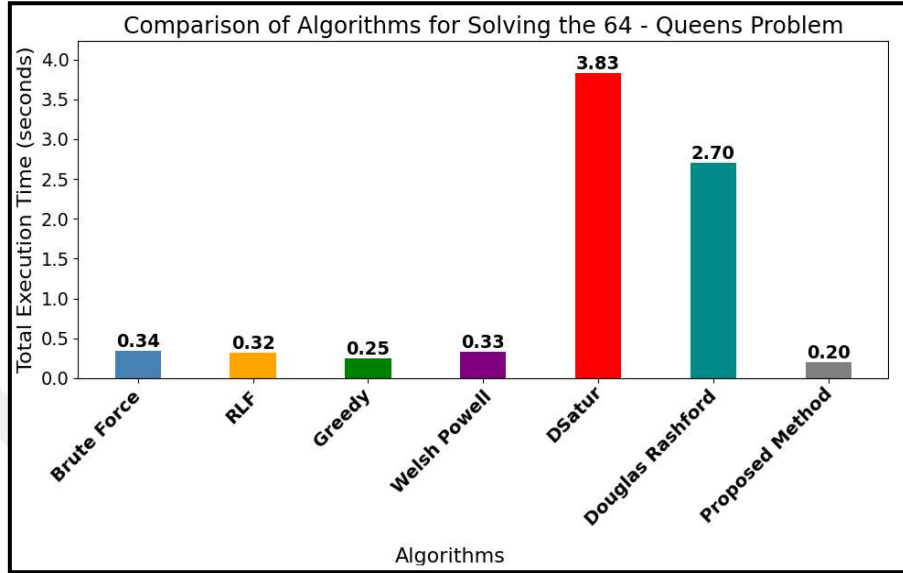


Şekil 19. 8 Vezir Problemi: 12 Benzersiz Çözümün Grafiği ve Yerleştirme Örnekleri

Vezirlerin birbirlerini tehdit etmemesi kuralı dikkate alınarak 12 farklı çözüm elde edildi. Vezir yerleştirme kuralları dikkate alınarak elde edilen çözüm süreleri, klasik grafik renklendirme algoritmalarıyla karşılaştırıldı. Renk tabanlı yöntem, grafik teorisiyle modellenmesi açısından geleneksel arama algoritmalarından ayrılır. Yöntemin temel başarısı, rastgele atamaların veya aşırı değer yöntemlerinin yol açtığı darboğazları önlemede yatmaktadır. Bu başarı, merkeziyet hesaplamalarına dayalı bir sıralama sistemi sayesinde elde edilmiştir. Elde edilen sonuçlar, vezir problemini çözmeye yönelik farklı bir renk tabanlı yaklaşımı ortaya koymaktadır.

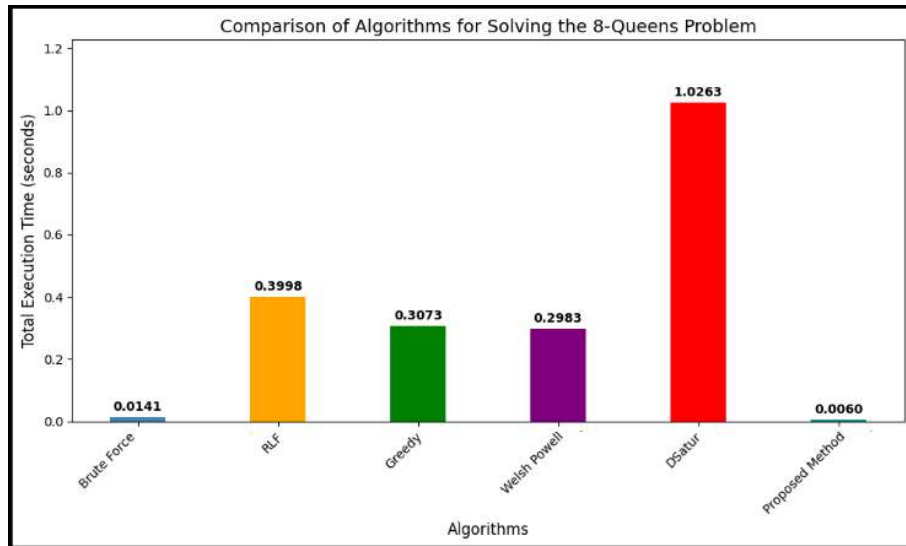
Önerilen MC algoritması, 64x64 satranç tahtasına 64 vezir yerleştirme probleminin çözüm süresi açısından bilinen grafik renklendirme algoritmalarıyla karşılaştırılmıştır. Bu karşılaştırmada DSatur, Brute Force, Rlf, Douglas Rashford, Greedy ve Welsh Powell algoritmaları dikkate alınmıştır. Önerilen algoritma, karşılaştırılan tüm algoritmalarından daha hızlı sonuçlar vermiştir. Önerilen yöntem, literatürde daha önce kullanılan grafik renklendirme algoritmalarıyla iki farklı şekilde karşılaştırıldı. Yöntem, 8 vezir probleminin 12 benzersiz çözümüne ve problemin üstel olarak genişletilmiş versiyonu olan 64

vezir yerleştirme problemine uygulandı ve aşağıdaki sonuçlar elde edildi. Aşağıda, çözüm üretme süreleri açısından MC algoritması ile literatürdeki farklı grafik renklendirme algoritmalarının sonuçları yer almaktadır. Bu karşılaştırmanın amacı, 8 vezir problemine ilişkin 12 farklı çözümü elde etmektir. MC algoritması, algoritmalar arasında en hızlı sonucu verdi.



Şekil 20. 64 Vezir Yerleştirme Probleminin Toplam Çözüm Süreleri

MC algoritması ile literatürdeki çeşitli grafik renklendirme algoritmalarının çözüm üretim sürelerine ilişkin sonuçları aşağıda verilmiştir. Bu karşılaştırmanın amacı, 8 vezir problemini katlanarak artırarak 64 vezir problemini ele almaktır. 64x64 boyutlarındaki bir satranç tahtasına 64 vezir yerleştirilmeye çalışıldı. MC algoritması, algoritmalar arasında en hızlı sonucu elde etti.



Şekil 21. Sekiz-Vezir Benzersiz Çözümlerin Toplam Çözüm Süreleri

5. SONUÇ ve ÖNERİLER

Bu tez çalışmasında, graf teorisini temel alan merkezilik ölçütü hesaplamaya yönelik yeni bir yaklaşım olan MC algoritmasının, kısıt tatmin problemlerinin bilinen örneklerinden olan Sudoku ve Sekiz-Vezir problemlerinin üzerindeki etkinliği araştırılmıştır. Önerilen yöntem de ilk olarak problemler birer graf yapısı olarak tanımlanmış, problemin içerdiği kutucuklar birer düğüm düğümler arası kısıtlara bağlı ilişkiler de kenar bağlantısı olarak tanımlanmıştır. MC algoritması kullanılarak problemin düğüm öncelikleri belirlenmesi adına merkezilik hesaplamaları yapılmış ve bu değerleri gözetilerek renklendirme yöntemi ile çözümler eldedilmek istenmiş ve çözüm uzayı daha etkin bir şekilde yönetilmeye çalışılmıştır.

Sudoku problemine yönelik olarak yapılan çalışmalarda önerilen yöntem; hem farklı boyutlardaki veri kümeleri hemde farklı zorluk seviyelerindeki veri setleri üzerinde değerlendirilmiştir. 1.000.000 adete kadar Sudoku problemi içeren veri kümeleri üzerindeki testler sonucunda, MC algoritması tüm zorluk seviyelerinde %100 çözüm başarısı gösterirken çözüm süresi açısından yaygın olarak kullanılan graf renklendirme algoritmalarından DSatur, Welsh-Powell, Rlf, Greedy, Brute Force ve Douglas-Rashford algoritmaları ile karşılaştırılmış ve daha yüksek performans gösterdiği belirlenmiştir. Merkezilik hesabı ve buna bağlı önem sıralaması sayesinde, klasik yaklaşımlara kıyasla gereksiz düğüm seçimleri ve denemeleri azaltıldığından çözüm süreci daha verimli hale getirmiştir.

Sekiz-Vezir problemi üzerine gerçekleştirilen deneysel test aşamalarının sonuçları incelendiğinde önerilen yaklaşımın değişik yapısal özellikler içeren kısıt tatmin problemlerine uygulanabilir olduğunu göstermiştir. Graf teorisi ve graf renklendirme yaklaşımları ile yeniden modellenen problem, satranç tahtası üzerine vezirlerin birbirleri için tehdit oluşturmayacak şekilde yerleştirilmesi amacıyla kullanılmış ve önerilen yöntem sayesinde elde edilen merkezilik değerleri bir önem sırası oluşturmayı mümkün kılmıştır. Bilinen graf renklendirme algoritmalarından DSatur, Welsh-Powell, Rlf, Greedy, Brute Force ve Douglas-Rashford algoritmaları ile yapılan karşılaştırmalı analizlerde MC algoritması, daha kısa sürede hem 12 farklı çözüm hemde 64x64 boyutunda 64 vezir yerleşim testlerinde yüksek zaman performansı sergilemiştir. Önerilen yöntemin önem sıralaması ve buna bağlı renklendirme yöntemi sayesinde klasik yöntemlerde karşılaşılabilen gereksiz deneme süreçleri veya tıkanıklık gibi problemleri önemli ölçüde azaltmıştır.

Elde edilen sonuçlar incelendiğinde, MC algoritmasının sadece belirli bir problem türüne ait çözüm yöntemi olmadığını, graf yapısına dönüştürülebilecek çok farklı kombinatoryal kısıt tatmin problemlerinde başarılı sonuçlar üretebilecek bir yaklaşım olduğu ortaya konmuştur. Merkezilik temelli düğüm önceliklendirme mekanizması ile çözüm aşamalarında karar verme süreçleri optimize edilmiş, daha az maliyetli hesaplama ve daha az sürede çözüm elde edilmiştir. Bu açıdan önerilen yöntem, graf

teorisi ve kısıt tatmin süreçlerinin bütünleşmesinde katkı sağlayan özgün bir yaklaşım olarak değerlendirilmektedir.

Gelecek çalışmalarda, MC algoritmasının bilinen gerçek zamanlı uygulamalara uygun optimize edilebilmesi, daha farklı ve büyük ölçeklerdeki NP-Tam problemler için uyarlanması ile makine öğrenmesi tabanlı yöntemlere hibrit çözümler getirmesi hedeflenmektedir. Ayrıca algoritmanın çigeleme, kaynak tahsisi, ağ optimizasyonu, kanal paylaşımı, sosyal ağ analizi, biyoinformatik ve karmaşık sistem modelleme gibi graf yapısında modellenebilecek farklı disiplinlerde kullanılması araştırılabilir. Bunun yanında merkezilik ölçütleri sayesinde yapay zeka ve örüntü tanıma gibi teknikler ile birlikte değerlendirilmesi, MC algoritmasının yüksek performanslı ve uyarlanabilir yeni sürümlerinin geliştirilmesi gibi katkılar sağlayabilir.

Sonuç olarak tez kapsamında ele alınan graf teorisini ve merkezilik ölçütünü temel alan çözüm yaklaşımı, büyük ölçekteki Sudoku veri kümelerinde hem de Sekiz-Vezir probleminde başarılı sonuçlar elde etmiş; zaman performansı, doğruluk ve ölçeklenebilirlik anlamında mevcut yöntemlere karşı önemli avantajlar ortaya koyduğunu göstermiştir. Edinilen bulgular, MC algoritmasının sadece benzer problemlerin çözümünde değil, birçok farklı NP-Tam problem için güçlü ve uygulanabilir alternatif bir yöntem olduğunu göstermektedir.

KAYNAKLAR

- [1] **V. Kumar**, “Algorithms for Constraint Satisfaction Problems: A Survey”, 1992.
- [2] **F. J. Aragón Artacho ve R. Campoy**, “Solving Graph Coloring Problems with the Douglas-Rachford Algorithm”, *Set. Valued. Var. Anal.*, c. 26, sy 2, ss. 277-304, Haz. 2018, doi: 10.1007/s11228-017-0461-4.
- [3] **J.-P. Delahaye**, “The Science behind Sudoku”, *Sci. Am.*, c. 294, sy 6, ss. 80-87, Haz. 2006, doi: 10.1038/scientificamerican0606-80.
- [4] **K. Gupta, S. Khatri, ve M. H. Khan**, “A Novel Automated Solver for Sudoku Images”, içinde *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, IEEE, Mar. 2019, ss. 1-6. doi: 10.1109/I2CT45611.2019.9033860.
- [5] **I. Rivin, I. Vardi, ve P. Zimmermann**, “The n-Queens Problem”, *The American Mathematical Monthly*, c. 101, sy 7, ss. 629-639, Ağu. 1994, doi: 10.1080/00029890.1994.11997004.
- [6] **S. Glock, D. M. Correia, ve B. Sudakov**, “The $\$n\$$ -queens completion problem”, May. 2022, [Çevrimiçi]. Erişim adresi: <http://arxiv.org/abs/2111.11402>
- [7] **J. Bell ve B. Stevens**, “A survey of known results and research areas for n-queens”, *Discrete Math.*, c. 309, sy 1, ss. 1-31, Oca. 2009, doi: 10.1016/j.disc.2007.12.043.
- [8] **A. Arteaga, U. Orozco-Rosas, O. Montiel, ve O. Castillo**, “Evaluation and Comparison of Brute-Force Search and Constrained Optimization Algorithms to Solve the N-Queens Problem”, 2022, ss. 121-140. doi: 10.1007/978-3-031-08266-5_9.
- [9] **G. Singh ve K. Deep**, “A new membrane algorithm using the rules of Particle Swarm Optimization incorporated within the framework of cell-like P-systems to solve Sudoku”, *Appl. Soft Comput.*, c. 45, ss. 27-39, Ağu. 2016, doi: 10.1016/j.asoc.2016.03.020.
- [10] **S. Chatterjee, S. Paladhi, ve R. Chakraborty**, “A Comparative Study On The Performance Characteristics Of Sudoku Solving Algorithms”, *IOSR J. Comput. Eng.*, c. 16, sy 5, ss. 69-77, 2014, doi: 10.9790/0661-16566977.
- [11] **S. Yakut, F. Öztemiz, ve A. Karci**, “A new robust approach to solve minimum vertex cover problem: Malatya vertex-cover algorithm”, *J. Supercomput.*, c. 79, sy 17, ss. 19746-19769, Kas. 2023, doi: 10.1007/s11227-023-05397-8.
- [12] **S. Yakut**, “A robust and efficient algorithm for graph coloring problem based on Malatya centrality and sequent independent sets”, *Egyptian Informatics Journal*, c. 30, s. 100676, Haz. 2025, doi: 10.1016/j.eij.2025.100676.
- [13] **S. Yakut ve E. Karagoz**, “A Graph-Theoretic Solution to NP-Complete Sudoku Puzzles: Malatya Centrality for Large-Scale Optimization”, *IEEE Access*, c. 13, ss. 182737-182750, 2025, doi: 10.1109/ACCESS.2025.3622191.
- [14] **E. Karagöz ve S. Yakut**, “Malatya Centrality Algorithm and Graph Colouring Based Effective and Efficient Eight Queen Problem Solution Method”, içinde *2025 9th International Artificial Intelligence and Data Processing Symposium (IDAP)*, IEEE, Eyl. 2025, ss. 1-7. doi: 10.1109/IDAP68205.2025.11222204.

- [15] **S. Kamal, S. S. Chawla, ve N. Goel**, "Detection of Sudoku puzzle using image processing and solving by Backtracking, Simulated Annealing and Genetic Algorithms: A comparative analysis", içinde *2015 Third International Conference on Image Information Processing (ICIIP)*, IEEE, Ara. 2015, ss. 179-184. doi: 10.1109/ICIIP.2015.7414762.
- [16] **M. BATAR**, "Grafik Teorisine Dayalı Etkin Bir Sudoku Çözücü Uygulaması", *International Journal of Innovative Engineering Applications*, c. 5, sy 2, ss. 218-224, Ara. 2021, doi: 10.46460/ijiea.982908.
- [17] **M. Prates ve L. Lamb**, "Problem Solving at the Edge of Chaos: Entropy, Puzzles and the Sudoku Freezing Transition", içinde *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, IEEE, Kas. 2018, ss. 686-693. doi: 10.1109/ICTAI.2018.00109.
- [18] **S.-W. Wang**, "A Dataset of Sudoku Puzzles With Difficulty Metrics Experienced by Human Players", *IEEE Access*, c. 12, ss. 104254-104262, 2024, doi: 10.1109/ACCESS.2024.3434632.
- [19] **S. Ono, R. Miyamoto, S. Nakayama, ve K. Mizuno**, "Difficulty Estimation of Number Place Puzzle and Its Problem Generation Support", 2009.
- [20] **R. Tomaszewski, W. Walendziuk, R. Trochimczuk, A. Wolniakowski, K. Jaroszewicz, ve Z. Slanina**, "Mechatronics system for solving SuDoKu puzzle based on Cartesian robot", *IFAC-Papers-OnLine*, c. 58, sy 9, ss. 149-153, 2024, doi: 10.1016/j.ifacol.2024.07.387.
- [21] **B. V. Indriyono vd.**, "Analysis of the Concept of Solving the Sudoku Game Using Backtracking and Brute Force Algorithms", *International Journal of Artificial Intelligence & Robotics (IJAIR)*, c. 6, sy 1, ss. 40-47, Tem. 2024, doi: 10.25139/ijair.v6i1.8377.
- [22] **M. S. Baltaoglu, B. Onal, ve A. E. Pusane**, "Sudoku puzzle as an LDPC code", içinde *2013 21st Signal Processing and Communications Applications Conference (SIU)*, IEEE, Nis. 2013, ss. 1-4. doi: 10.1109/SIU.2013.6531199.
- [23] **Y. Tang, Z. Wu, ve C. Zhu**, "An improved strategy for solving Sudoku by sparse optimization methods".
- [24] **J. Gunther ve T. Moon**, "Entropy Minimization for Solving Sudoku", *IEEE Transactions on Signal Processing*, c. 60, sy 1, ss. 508-513, Oca. 2012, doi: 10.1109/TSP.2011.2169253.
- [25] **R. Marappan ve G. Sethumadhavan**, "Solution to Graph Coloring Using Genetic and Tabu Search Procedures", *Arab. J. Sci. Eng.*, c. 43, sy 2, ss. 525-542, Şub. 2018, doi: 10.1007/s13369-017-2686-9.
- [26] **J.-L. Kim ve E. Eor**, "A Genetic Algorithm for Solving Sudoku Based on Multi-Armed Bandit Selection", *IEEE Trans. Games*, ss. 1-14, 2024, doi: 10.1109/TG.2024.3487861.
- [27] **K. Waiyapara, W. Wattanapornprom, ve P. Chongstitvatana**, "Solving Sudoku puzzles with node based Coincidence algorithm", içinde *The 2013 10th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, IEEE, May. 2013, ss. 11-16. doi: 10.1109/JCSSE.2013.6567311.
- [28] **S. Streicher ve J. A. Du Preez**, "Strengthening Probabilistic Graphical Models: The Purge-and-Merge Algorithm", *IEEE Access*, c. 9, ss. 149423-149432, 2021, doi: 10.1109/ACCESS.2021.3124760.

- [29] **H. Tsuiki**, "SUDOKU Colorings of the Hexagonal Bipyramid Fractal", 2008, ss. 224-235. doi: 10.1007/978-3-540-89550-3_24.
- [30] **D. Rai**, "POLYNOMIAL 3-SAT REDUCTION OF SUDOKU PUZZLE", *International Journal of Advanced Research in Computer Science*, c. 9, sy 3, ss. 194-197, Haz. 2018, doi: 10.26483/ijarcs.v9i3.6096.
- [31] **D. Rai, N. S. Chaudhari, ve M. Ingle**, "An Efficient Algorithmic 3-SAT Formulation for Sudoku Puzzle using Graph Coloring", içinde *2018 International Conference on Advanced Computation and Telecommunication (ICACAT)*, IEEE, Ara. 2018, ss. 1-6. doi: 10.1109/ICACAT.2018.8933664.
- [32] **S. Davis, M. Henderson, ve A. Smith**, "Modeling Sudoku Puzzles with Python", 2010, ss. 49-55. doi: 10.25080/Majora-92bf1922-009.
- [33] **P. Pdxscholar ve K. Oddson**, "Counting and Coloring Sudoku Graphs", 2019. [Çevrimiçi]. Erişim adresi: https://pdxscholar.library.pdx.edu/mth_grad
- [34] **V. R, J. B. T, ve N. Guruprasad**, "Graph Coloring Approach to Solve Sudoku Problems", içinde *2024 11th International Conference on Computing for Sustainable Global Development (INDIACom)*, IEEE, Şub. 2024, ss. 529-533. doi: 10.23919/INDIACom61295.2024.10498817.
- [35] **J. Maria Jeyaseeli, G. C. Lau, W. C. Shiu, ve S. Arumugam**, "Sudoku number of graphs", *AKCE International Journal of Graphs and Combinatorics*, c. 20, sy 2, ss. 209-216, May. 2023, doi: 10.1080/09728600.2023.2218917.
- [36] **A. Sharma ve G. L. Prajapati**, "Inductive transfer learning applied to graph coloring problem using Sudoku", içinde *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, IEEE, Tem. 2017, ss. 1-4. doi: 10.1109/ICCCNT.2017.8203958.
- [37] **F. Akman**, "Partial Chromatic Polynomials and Diagonally Distinct Sudoku Squares", Nis. 2008, [Çevrimiçi]. Erişim adresi: <http://arxiv.org/abs/0804.0284>
- [38] **R. Tovey ve J. Liang**, "The Fun is Finite: Douglas-Rachford and Sudoku Puzzle -- Finite Termination and Local Linear Convergence", Eyl. 2020, [Çevrimiçi]. Erişim adresi: <http://arxiv.org/abs/2009.04018>
- [39] **J. Postigo, J. Soto-Begazo, V. R. Fiorela, G. M. Picha, R. Flores-Quispe, ve Y. Velazco-Paredes**, "Comparative Analysis of the main Graph Coloring Algorithms", içinde *2021 IEEE Colombian Conference on Communications and Computing (COLCOM)*, IEEE, May. 2021, ss. 1-6. doi: 10.1109/COLCOM52710.2021.9486299.
- [40] **I. Méndez-Díaz, G. Nasini, ve D. Severín**, "A DSATUR-based algorithm for the Equitable Coloring Problem", *Comput. Oper. Res.*, c. 57, ss. 41-50, May. 2015, doi: 10.1016/j.cor.2014.11.014.
- [41] **B. Crawford, C. Castro, ve E. Monfroy**, "Solving Sudoku with Constraint Programming", 2009, ss. 345-348. doi: 10.1007/978-3-642-02298-2_52.
- [42] **A. K. Kumar, N. Ngoc Mai, S. Guo, ve L. Han**, "Entanglement inspired approach for determining the preeminent arrangement of static cameras in a multi-view computer vision system", *Visual Computer*, c. 39, sy 7, ss. 2847-2863, Tem. 2023, doi: 10.1007/s00371-022-02497-z.

- [43] **A. K. Kumar, N. N. Mai, K. Tian, ve Y. Xia**, "Isochronous Temporal Metric for Neighbourhood Analysis in Classification Tasks", *SN Comput. Sci.*, c. 4, sy 6, Eki. 2023, doi: 10.1007/s42979-023-02351-6.
- [44] **A. Saxena ve S. Iyengar**, "Centrality Measures in Complex Networks: A Survey", Kas. 2020, [Çevrimiçi]. Erişim adresi: <http://arxiv.org/abs/2011.07190>
- [45] **M. C. Camur ve C. Vogiatzis**, "A Survey on Optimization Studies of Group Centrality Metrics", May. 2024, [Çevrimiçi]. Erişim adresi: <http://arxiv.org/abs/2401.05235>
- [46] **A. KARCI, S. YAKUT, ve F. ÖZTEMİZ**, "A New Approach Based on Centrality Value in Solving the Minimum Vertex Cover Problem: Malatya Centrality Algorithm", *Computer Science*, Kas. 2022, doi: 10.53070/bbd.1195501.
- [47] **S. YAKUT**, "Kendall rank correlation analysis of Malatya Centrality Algorithm with well-known centrality measures", *Sigma Journal of Engineering and Natural Sciences – Sigma Mühendislik ve Fen Bilimleri Dergisi*, ss. 1339-1354, 2025, doi: 10.14744/sigma.2025.00124.
- [48] **S. Yakut**, "Efficient and Robust Graph Coloring Algorithm for Graph Coloring Problem: Malatya Vertex Coloring Algorithm", *IEEE Access*, ss. 1-1, 2025, doi: 10.1109/ACCESS.2025.3614296.
- [49] **S. Güldal, Saleh Allehaibi, ve Veronica Baugh**, *N-Queens Solving Algorithm by Sets and Backtracking*. IEEE, 2016.
- [50] **U. Sarkar ve S. Nag**, "An Adaptive Genetic Algorithm for Solving N-Queens Problem".
- [51] **V. Jain ve J. S. Prasad**, "Solving N-queen problem using genetic algorithm by advance mutation operator", *International Journal of Electrical and Computer Engineering*, c. 8, sy 6, ss. 4519-4523, Ara. 2018, doi: 10.11591/ijece.v8i6.pp4519-4523.
- [52] **S. R. Jolfaei ve S. K. H. Abadi**, "Application of the Brain Drain Optimization Algorithm to the N-Queens Problem", Nis. 2025, [Çevrimiçi]. Erişim adresi: <http://arxiv.org/abs/2504.18953>
- [53] **O. Moghimi ve A. Amini**, "A Novel Approach for Solving the N-Queen Problem Using a Non-Sequential Conflict Resolution Algorithm", *Electronics (Switzerland)*, c. 13, sy 20, Eki. 2024, doi: 10.3390/electronics13204065.
- [54] **B. KARABULUT, A. ERGÜZEN, ve H. M. ÜNVER**, "A Linear Time Pattern Based Algorithm for N-Queens Problem", *Politeknik Dergisi*, c. 25, sy 2, ss. 615-622, Haz. 2022, doi: 10.2339/politeknik.762967.
- [55] **C. Zhang ve J. Ma**, "Counting Solutions for the N-queens and Latin Square Problems by Efficient Monte Carlo Simulations", Ağu. 2008, doi: 10.1103/PhysRevE.79.016703.
- [56] **S. V. Aravind Krishna, R. Prashanna, J. Midhru Jayan, ve G. Jeyakumar**, "A Comparative Study on using Genetic Algorithm with DEAP framework on Different Optimization Problems", içinde *Proceedings of International Conference on Contemporary Computing and Informatics, IC3I 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, ss. 111-117. doi: 10.1109/IC3I61595.2024.10828765.
- [57] **Yuxin Ding, Ye Li, Min Xiao, Qing Wang, ve Li Dong**, *A High Order Neural Network to Solve N-Queens Problem*. [IEEE], 2010.
- [58] **Raziyeh Moghaddas, Ali Hamzeh, ve Eghbal Mansoori**, *A New Multi-Agent Algorithm for Solving Constraint Satisfaction Problems*. [publisher not identified], 2013.

- [59] **Santhosh G S, Piyush Joshi, Ayan Barui, ve Prasanta K. Panigrahi**, "A Quantum Approach to Solve N-Queens Problem", içinde *2024 16th International Conference on COMMunication Systems and NETWORKS, COMSNETS 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, ss. 1046-1051. doi: 10.1109/COMSNETS59351.2024.10427458.
- [60] **N. Shivappa**, "A Survey on Classic Examples of Constraint Satisfaction Problems with Solutions and Applications", içinde *2023 3rd International Conference on Smart Generation Computing, Communication and Networking, SMART GENCON 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/SMARTGENCON60755.2023.10442726.
- [61] **S. K. Guha, S. Bhadra, S. Hazra, S. Chatterjee, ve A. Bhattacharyya**, "Classical Optimization Problem Solution using Nature Inspired Algorithm", içinde *2025 International Conference on Recent Advances in Electrical, Electronics, Ubiquitous Communication, and Computational Intelligence (RAEEUCCI)*, IEEE, Nis. 2025, ss. 1-5. doi: 10.1109/RAEEUCCI63961.2025.11048319.
- [62] **Z. E. A. Bouneb**, "Distributed algorithm for parallel computation of the n queens solutions", *Expert Syst. Appl.*, c. 255, Ara. 2024, doi: 10.1016/j.eswa.2024.124677.
- [63] **R. Sosii ve J. Gu**, "Fast Search Algorithms for the N-Queens Problem".
- [64] **A. Dehghani, R. Namvar, ve A. Khalili**, "Henry V: A linear time algorithm for solving the N-Queens Problem using only 5 patterns", içinde *2023 28th International Computer Conference, Computer Society of Iran, CSICC 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/CSICC58665.2023.10105390.
- [65] **Jalal eddin Aghazadeh heris ve Mohammadreza Asgari Oskoei**, *Modified Genetic Algorithm for Solving n-Queens Problem*. IEEE, 2014.
- [66] **O. K. Majeed vd.**, "Performance comparison of genetic algorithms with traditional search techniques on the N-Queen Problem", içinde *2023 International Conference on IT and Industrial Technologies, ICIT 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ICIT59216.2023.10335855.
- [67] **A. S. Farhan, W. Z. Tareq, ve F. H. Awad**, "Solving N Queen Problem using Genetic Algorithm", 2015.
- [68] **A. M. Herzberg ve M. R. Murty**, "Sudoku Squares and Chromatic Polynomials".
- [69] **K. D. May ve K. D. May**, "Colorings and Sudoku Puzzles", 2020. [Çevrimiçi]. Erişim adresi: <https://scholar.rose-hulman.edu/rhumj/vol21/iss1/6>
- [70] **Kyubyong Park**, "<https://www.kaggle.com/datasets/bryanpark/sudoku>".
- [71] **S. Yakut**, "An Effective Approach Based on the Malatya Centrality Algorithm for Determining the Maximum Independent Set and Minimum Vertex Cover in Molecular Graphs", içinde *8th International Artificial Intelligence and Data Processing Symposium, IDAP 2024*, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/IDAP64064.2024.10710745.
- [72] **R. W. Topor**, "FUNDAMENTAL SOLUTIONS OF THE EIGHT QUEENS PROBLEM", 1982.
- [73] **T. P. Chatzinikolaou, R.-E. Karamani, I.-A. Fyrigos, ve G. Ch. Sirakoulis**, "Handling Sudoku puzzles with irregular learning cellular automata", *Nat. Comput.*, c. 23, sy 1, ss. 41-60, Mar. 2024, doi: 10.1007/s11047-024-09975-4.

- [74] **A. Z. Sevkli ve K. A. Hamza**, “General variable neighborhood search for solving Sudoku puzzles: unfiltered and filtered models”, *Soft comput.*, c. 23, sy 15, ss. 6585-6601, Ağu. 2019, doi: 10.1007/s00500-018-3307-6.
- [75] **K. Nishikawa ve T. Toda**, “Exact Method for Generating Strategy-Solvable Sudoku Clues”, *Algorithms*, c. 13, sy 7, s. 171, Tem. 2020, doi: 10.3390/a13070171.
- [76] **K. Chen**, “Exploring A Better Way to Constraint Propagation Using Naked Pair”, *ITM Web of Conferences*, c. 70, s. 04025, Oca. 2025, doi: 10.1051/itmconf/20257004025.
- [77] **L. Tao vd.**, “Blended Glial Cell’s Spiking Neural Network”, *IEEE Access*, c. 11, ss. 43566-43582, 2023, doi: 10.1109/ACCESS.2023.3267856.
- [78] **A. SayedChowdhury ve S. Akhter**, “Solving Sudoku using Boolean Algebra”, *Int. J. Comput. Appl.*, c. 52, sy 21, ss. 1-5, Ağu. 2012, doi: 10.5120/8331-9024.



- **S. Yakut ve E. Karagoz**, “A Graph-Theoretic Solution to NP-Complete Sudoku Puzzles: Malatya Centrality for Large-Scale Optimization”, *IEEE Access*, c. 13, ss. 182737-182750, 2025, doi: [10.1109/ACCESS.2025.3622191](https://doi.org/10.1109/ACCESS.2025.3622191).
- **E. Karagöz ve S. Yakut**, “Malatya Centrality Algorithm and Graph Colouring Based Effective and Efficient Eight Queen Problem Solution Method”, içinde *2025 9th International Artificial Intelligence and Data Processing Symposium (IDAP)*, IEEE, Eyl. 2025, ss. 1-7. doi: [10.1109/IDAP68205.2025.11222204](https://doi.org/10.1109/IDAP68205.2025.11222204).

