



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



KÜTÜPHANELERDE KİTAP TOPLAMA VE
YERLEŞTİRME OPTİMİZASYONU İÇİN BİR
KARAR DESTEK SİSTEMİ

Hasan Basri ÇAKIR

YÜKSEK LİSANS

Endüstri Mühendisliği Anabilim Dalı

ARALIK-2022
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Hasan Basri ÇAKIR tarafından hazırlanan “KÜTÜPHANELERDE KİTAP TOPLAMA VE YERLEŞTİRME OPTİMİZASYONU İÇİN BİR KARAR DESTEK SİSTEMİ” adlı tez çalışması 24/01/2023 tarihinde aşağıdaki jüri tarafından oy birliği ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Endüstri Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Prof. Dr. Öğr. Üyesi Mehmet AKTAN

.....

Danışman

Dr. Öğr. Üyesi Alper DÖYEN

.....

Üye

Dr. Öğr. Üyesi Yakup ATASAGUN

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Saadettin Erhan KESEN
Enstitü Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Hasan Basri ÇAKIR

Tarih: 30.12.2022

ÖZET

YÜKSEK LİSANS

KÜTÜPHANELERDE KİTAP TOPLAMA VE YERLEŞTİRME OPTİMİZASYONU İÇİN BİR KARAR DESTEK SİSTEMİ

Hasan Basri ÇAKIR

**Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Endüstri Mühendisliği Anabilim Dalı**

Danışman: Dr. Öğr. Üyesi Alper DÖYEN

2022, 81 Sayfa

Jüri

**Dr. Öğr. Üyesi Alper DÖYEN
Prof. Dr. Mehmet AKTAN
Dr. Öğr. Üyesi Yakup ATASAGUN**

Üniversitelerdeki öğrenci sayısı arttıkça, kütüphanelerdeki ödünç alınan ve kullanılan kitap sayısı da artmaktadır, bu da daha fazla kitabın iade edilmesine ve yeniden rafa kaldırılmasına yol açmaktadır. İlgili kütüphane personelleri kütüphane içinde daha fazla mesafeler kat etmek zorunda kalmakta ve çalışma verimleri düşmektedir. Yeniden raflara yerleştirme sürecini iyileştirmek amacıyla, bu tezde bir karar destek sistemi (KDS) önerilmiştir. KDS ile kütüphaneler için uygulanabilecek çeşitli kitap toplama ve yerleştirme senaryoları bir şebeke optimizasyonu problemi olarak incelenmiştir. Kütüphane içerisindeki masalar ve raflar düğümleri, yürüyüş koridorları ise ayrıtları (hatları) oluşturmaktadır. Bu problem temelde, ziyaret edilecek düğümler arasında öncelik ilişkilerinin olduğu ve kat edilecek toplam mesafenin en küçüklenmesini amaçlayan bir ardışık sıralama problemi olarak ele alınmaktadır. Senaryolar, farklı sezgisel ve meta-sezgisel yöntemlerle çözülmüştür. Literatürde benzer problemler için kullanılabilen en yakın komşu sezgiseli ve süpürme algoritması sezgisellerine ilave olarak parçacık sürü optimizasyonu meta sezgiseli uygulanmıştır. Ayrıca, bu tezde incelenen probleme özgü bir sezgisel de önerilmiştir. Geliştirilen rastsal problemler ile hem farklı senaryoların hem de çözüm yöntemlerinin performanslarının karşılaştırılması yapılmıştır. Kitapların kütüphane içerisinde bir veya iki konumdan toplandığı durumları içeren senaryoların diğer senaryolara göre çok daha verimli olduğu görülmüştür. Tezde incelenen kütüphaneler için kitap toplama ve yerleştirme optimizasyonu problemi için literatürde herhangi bir çalışmaya rastlanmamıştır. İlave olarak, problemin çözümü için kullanılan KDS; kütüphane veri tabanı, şebeke yapısı, kitap toplama ve yerleştirme senaryoları çözüm yöntemleri ve çözümlerin karşılaştırılması gibi işlemlere sahip bilinen ilk KDS'dir. Bu nedenle, tezin hem literatüre hem de kütüphanelerdeki verimlilik sorunu üzerine önemli katkıları olacağı düşünülmektedir.

Anahtar Kelimeler: Ardışık Sıralama Problemi, En Yakın Komşu Sezgiseli, Kitap Toplama ve Yerleştirme Problemi, Kütüphane Karar Destek Sistemi, Parçacık Sürü Optimizasyonu, Süpürme Algoritması

ABSTRACT

MS THESIS

A DECISION SUPPORT SYSTEM FOR OPTIMIZING BOOK PICKUP AND RE-SHELVING IN LIBRARIES

Hasan Basri ÇAKIR

**Konya Technical University
Institute of Graduate Studies
Department of Industrial Engineering**

Advisor: Asst. Prof. Dr. Alper DÖYEN

2022, 81 Pages

**Jury
Asst. Prof. Dr. Alper DÖYEN
Prof. Dr. Mehmet AKTAN
Asst. Prof. Dr. Yakup ATASAGUN**

As the number of students in universities increases, the number of books borrowed and used in libraries also increases, leading to more books returned and re-shelved. Library personnel need to walk greater distances within the library and their work efficiency decreases. This thesis proposes a Decision Support System (DSS) to improve the re-shelving process. Different book pickup and re-shelving scenarios for libraries are studied as a network optimization problem in the DSS. Tables and shelves in the library form nodes and walking corridors form edges. This problem is essentially a sequential ordering problem, with the objective of minimizing the total distance traveled and relationships of priority between nodes to be visited. Scenarios are solved by using different heuristic and meta-heuristic methods. These include particle swarm optimization meta-heuristic in addition to the nearest neighbor and the sweep algorithm heuristics which are commonly used for similar problems. Additionally, a specific heuristic is proposed for the problem examined in this thesis. The performance of different scenarios and solution methods is compared using randomly generated problem instances. It is shown that scenarios involving book pickup from just one or two locations within the library are significantly more effective than others. The book pickup and re-shelving optimization problem considered in this thesis has not been studied in the literature before. In addition, the DSS utilized to solve the problem is the first DSS to provide features including a library database, network structure, book pickup and re-shelving scenarios, solution methodologies, and solution comparison. For this reason, it is believed that the thesis will make important contributions to both the literature and efficiency issues in libraries.

Keywords: Book Pickup and Re-Shelving Problem, Library Decision Support System, Nearest Neighbor Heuristics, Particle Swarm Optimization, Sequential Ordering Problem, Sweep Algorithm

ÖNSÖZ

Lisans ve yüksek lisans sürecimde benimle hiç yorulmadan, kızmadan, sıkılmadan ilgilenen, bilgilerini ve tecrübelerini gün saat fark etmeksizin paylaşan, yeri geldiğinde öğretmen-öğrenci, yeri geldiğinde abi-kardeş olma şansını bana tanıyan danışmanım, Dr. Öğr. Üyesi Alper DÖYEN'e; özellikle yüksek lisans sürecim boyunca sürekli bir şeyler danıştığım, istisnasız her sorunuma çözüm bulmaya çalışan, maddi manevi hiçbir desteğini esirgemeyen Arş. Görevlisi Aslıhan Fatma KULA'ya; bana kattıkları, yaşattıkları, öğrettikleri her şey için beni her zaman destekleyen Sinerji Topluluğu'nun gelmiş geçmiş her bir üyesine; ve son olarak tüm eğitim hayatım boyunca benden desteklerini esirgemeyen ve her daim yanımda olan başta sevgili ablam Gonca ÇAKIR olmak üzere aileme sonsuz teşekkür ederim.

Hasan Basri ÇAKIR
KONYA-2022

İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
SİMGELER VE KISALTMALAR	ix
1. GİRİŞ	1
2. KAYNAK ARAŞTIRMASI	5
2.1. Ardışık Sıralama Problemi (ASP) ve Öncelik Kısıtlamaları Olan Gezgin Satıcı Problemi (ÖKGSP)	6
2.2. KTYP için Kullanılan Çözüm Yöntemleri	9
2.2.1 En Yakın Komşu Sezgiseli (EYKS)	10
2.2.2 Süpürme Algoritması (SÜP_A)	11
2.2.3 Parçacık Sürü Optimizasyonu (PSO)	13
3. MATERYAL VE YÖNTEM.....	17
3.1. KTYP'nin Konya Teknik Üniversitesi Kütüphanesine (KTUNK) Uyarlanması 18	
3.2. Geliştirilen Karar Destek Sistemi Yapısı	21
3.2.1 Kitap bilgilerinin (Ekle-Sil-Güncelle-Filtrele) işlemleri	21
3.2.2 Raf ve masa (Ekle-Sil-Güncelle) işlemleri	22
3.2.3 Veri tabanında kullanılan gerekli sanal tablolar	25
3.2.4 KDS'deki algoritmalara gerekli olan saklı yordam	30
3.3. KTYP Senaryoları.....	32
3.3.1 Senaryo-1	33
3.3.2 Senaryo-2	34
3.3.3 Senaryo-3	35
3.3.4 Senaryo-4	35
3.3.5 Senaryo-5	36
3.3.6 Senaryo-6	36
3.3.7 Senaryo-7	39
3.3.8 Senaryo-8	41
3.4 Çözüm Yöntemleri.....	41
3.4.1 Rastsal Çözüm Algoritması	44
3.4.2 En Yakın Komşu Sezgiseli (EYKS)	45
3.4.3 Bölgesel En Yakın Komşu Sezgiseli (BEYKS)	47
3.4.4 Süpürme Algoritması (SÜP_A)	50
3.4.5 Süpürme Algoritması-2 (SÜP_A-2)	51
3.4.6 Parçacık Sürü Optimizasyonu (PSO)	52

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA.....	59
4.1. Deneysel Problemlerin Oluşturulması	59
4.2. Geliştirilen Çözüm Yöntemlerinin KTYP Senaryoları ile Karşılaştırılması	62
5. SONUÇLAR VE ÖNERİLER	76
5.1 Sonuçlar	76
5.2 Öneriler	77
KAYNAKLAR	78



SİMGELER VE KISALTMALAR

Simgeler

A = Ziyaret sırasına göre ataması gerçekleşen düğümlerin kümesi .

AB = Atanabilir düğümler kümesi.

$NA = N - (A \cup AB)$: N kümesinde olup hem A 'da hem de AB 'de olmayan düğümlerin kümesi.

$N = (0, 1, \dots, i, \dots, n)$ Tüm düğümlerin bulunduğu set.

t = iterasyon sayısı

p = parçacık sayısı

$SF = \{ (a, b) \}$ a düğümünde başla b düğümünde bitir.

$\ddot{O} = \{ (j, i) \}$ j düğümüne gitmeden önce i düğümüne gidilmesi gerektiğini gösteren öncelik ilişkileri seti.

$C_{i,j}$ = i 'den j 'ye gitme maliyeti.

$P_{t,p}$ = t . iterasyonda p . parçacığın pozisyonu.

$PU_{t,p}$ = t . iterasyonda p . parçacığının uygunluk değeri.

$Pbest_p$ = p . parçacığın şimdiye kadarki en iyi pozisyonu.

$PUBest_p$ = p . parçacığın şimdiye kadarki en iyi uygunluk değeri.

$Gbest$ = parçacıklar arasında en iyi pozisyon.

$GUBest$ = parçacıklar arasında en iyi uygunluk değeri.

$v_{t+1,p}$ = bir sonraki iterasyonda p . parçacığının hızı.

c_1 = Bilişsel öğrenme katsayısı.

c_2 = Sosyal öğrenme katsayısı.

SO_{Gbest} = $PU_{t,p}$ ziyaret sırası $Gbest$ 'e göre farklı olan düğümler.

SO_{Pbest} = $PU_{t,p}$ ziyaret sırası $Gbest$ 'e göre farklı olan düğümler.

$Ghız$ miktarı = $c_2 * SO_{Gbest}$ miktarı

$Ghız$ = $Ghız$ miktarı kadar SO_{Gbest} 'ten rastgele yer değiştirme işlemleri.

$Phız$ miktarı = $c_1 * SO_{Pbest}$ miktarı

$Phız$ = $Phız$ miktarı kadar SO_{Pbest} t'ten rastgele yer değiştirme işlemleri

$v_{t+1,p} = Ghız + Phız = p$. parçacığın $t + 1$. zamandaki hızı

Kısaltmalar

AÇP	Araç Çağırma Problemi
ASP	Ardışık Sıralama Problemi
BEYKS	Bölgesel En Yakın Komşu Sezgiseli
BEÇM	Bilinen En İyi Çözümün Maliyeti
BÇS	Başlangıç Çözüm Seti
ÇM	Çalışma Masası
DM	Danışma Masası
EYKS	En Yakın Komşu Sezgiseli
EZTY	Eş Zamanlı Topla-Yerleştir
GSP	Gezgin Satıcı Problemi
KDS	Karar Destek Sistemi
KKO	Karınca Kolonisi Optimizasyonu
KTUNK	Konya Teknik Üniversitesi Kütüphanesi
KTYP	Kitap Toplama ve Yerleştirme Problemi
ÖKAGSP	Öncelik Kısıtlamaları olan Asimetrik Gezgin Satıcı Problemi
ÖKGSP	Öncelik Kısıtlamaları olan Gezgin Satıcı Problemi
ÖTSY	Önce Topla Sonra Yerleştir
PSO	Parçacık Sürü Optimizasyonu
SSMS	Microsoft Sql Server Management Studio
R-PSO	Rastsal Parçacık Sürü Optimizasyonu
SÜP_A	Süpürme Algoritması
TM	Toplama Masası
UÇP	Uygun Çözüme Çevirme Prosedürü
ZK-ÖKGSP	Zaman Kısıtlı Öncelik Kısıtlamaları Olan Gezgin Satıcı Problemi
ZPARP	Zaman Pencereli Araç Rotalama Problemi

1. GİRİŞ

Yüksek Öğretim Kurumu (YÖK) verilerine göre üniversitelerde öğrenci sayısı her geçen yıl artmaktadır (Yükseköğretim Bilgi Yönetim Sistemi, 2022). Bu nedenle kütüphanelerde iade şartı ile ödünç alınan kitapların ve kütüphane içerisinde kullanılan kitapların miktarları artmıştır. Bir kitabın iade edilme durumu, kütüphane dışarısına götürülen kitabın tekrar kütüphaneye teslim edilmesidir. İade işlemi genellikle kütüphane görevlilerinin bulunduğu danışma masalarında gerçekleşir. Belirli kontroller sağlandıktan sonra görevli iade edilecek kitapları teslim alır. Hem iade edilen hem de kütüphane içerisinde kullanılıp çalışma masalarında bırakılan kitapların sayısının artması sebebiyle kitapların raflara yerleştirilmesi giderek zorlaşmaktadır. Bu kitap trafiği yüzünden kütüphane görevlileri, kitapları yerleştirme işlemleri sırasında daha fazla mesafe katetmektedir. Bu durum sadece üniversitelerdeki kütüphaneler için değil genel olarak il/ilçe kütüphaneleri, halk kütüphaneleri vb. kütüphaneler için ciddi bir sorun teşkil etmektedir.

Bu tez, kitapları raflardan belirli bir amaç için alan insanlardan ‘kullanıcılar’ olarak bahsetmektedir. Danışma masasında bulunan iadesi gerçekleşmiş kitapları ‘iade edilen kitap’, çalışma masalarında bulunan ve raflara yerleştirilmesi gereken kitapları ise ‘kullanılmış kitaplar’ olarak adlandırmaktadır.

Her kütüphanede belirli bir kodlama sistemine göre kitaplar raflara yerleştirilir. Kullanıcıların çoğu bu kodlamayı bilmediği için kütüphane görevlileri kullanılan kitapların kullanıcılar tarafından raflara yerleştirilmesini istememektedir. Dolayısıyla görevliler, kullanıcılardan kütüphanelerde kullandıkları kitap(lar)ı ya çalışma masalarına ya danışma masasına ya da varsa kütüphane içerisinde belirlenen bir toplama masasına bırakmalarını istemektedirler.

İade edilen kitaplar ait oldukları rafa yerleştirilirken, kütüphane içerisinde bulunan çalışma masalarında da kitaplar bulunabilir. Dolayısı ile bu çalışma masalarında bulunan kitapların da ait oldukları raflara tekrar yerleştirilmesi gerekmektedir. Uygun bir en kısa yerleştirme rotası oluşturulurken, çalışma masalarında kitap olma durumu da ele alınmıştır. İade edilen kitapların raflara yerleştirilmesi dışında, çalışma masalarında da kitapların olma durumu ve bu kitapların da raflara yerleştirilmesi çalışmanın en önemli unsurlarından biridir.

Genellikle kütüphanelerde görevliler, iade edilen kitapları yeniden raflara yerleştirmek için danışma masasında yeterince kapasitesi olan bir taşıma aracı ile taşırken

çalışma masalarını da teker teker dolaşp kitap varsa kitabı alıp, ait oldukları rafı da rotalarına eklemektedir. Görevli veya kütüphanelerde kullanılan karar destek sistemleri, hangi çalışma masasında hangi kitap var gibi soruların cevabını önceden bilmediği için problem gerçek hayatta belirsizlik içermektedir. Bu belirsizlik ancak masalar ziyaret edildikçe ortadan kalkmakta ve her masa ziyaretinde yerleştirilecek yeni kitaplar ortaya çıkabilmektedir. Kitapların ait oldukları raflara yerleştirilmesi için görevlinin oluşturacağı gezinti sırası, görevlinin bilgi ve tecrübesine dayanan tamamen sezgisel bir yöntem ile oluşturulur. Görevli her yeni masa ziyaretinde ortaya çıkan yeni kitaplar için gezinti rotasını (yerleştirme rotası) güncellemektedir. Böylece problem gerçekte dinamik bir yapıya sahiptir.

Bu tezde öncelikle, bu dinamik durumun iyileştirilmesi adına bir öneride bulunmaktadır. Her çalışma masasına kitap barkod okuyucuları konulursa ve kullanıcılar masaya oturduklarında ellerindeki kitabı bu okuyucuya okutulursa kütüphane veri tabanı içerisinde her masada hangi kitabın bulunduğu bilinebilir hale gelmektedir. Görevli bastan masalardaki kitap dağılımını bileceği için gezinti rotasını bastan bir kere oluşturacaktır (her masa ziyaretinde yenilemek durumunda kalmayacaktır). Bu sebeple problemin dinamik yapısı statik hale dönüşecektir. Tez kapsamında 5 adet statik, 3 adet dinamik olmak üzere 8 farklı kitap toplama ve yerleştirme senaryosu incelenmektedir.

Yeniden raflara yerleştirme sürecini iyileştirmek amacıyla, bu tezde optimizasyon temelli bir karar destek sistemi (KDS) önerilmiştir. KDS içerisindeki veri tabanı yapısında kitapların kayıtları tutulmakta, yeni kitap ekleme, kitap çıkarma, kitapların raf bilgilerinin güncellenmesi işlemleri kolaylıkla yapılabilmektedir.

KDS ile kütüphaneler için uygulanabilecek kitap toplama ve yerleştirme senaryoları (statik ve dinamik senaryolar) bir şebeke optimizasyonu problemi olarak incelenmiştir. Bu amaçla kütüphane içerisindeki danışma masası, toplama masası, raflar ve çalışma masalarının belirli bölümleri düğüm olarak, bu düğümler arasındaki yürüyüş koridorları ise ayrıt (hat) olarak modellenmiştir. Oluşturulan şebeke KDS içerisinde tutulmakta ve oluşturulan senaryolara ait optimizasyon problemi bu şebeke yapısı üzerinde çözülmektedir.

Tez içerisinde Kitap Toplama Yerleştirme Problemi (KTYP) olarak isimlendirilen optimizasyon probleminin amacı ziyaret edilmesi gereken tüm masaları ve rafları (düğümleri) toplamda en kısa mesafe katetecek şekilde ziyaret edecek rotanın elde edilmesidir. Masalardan kitapları toplamadan o kitapların yerleştirileceği raflara

gidilemeyecektir. Dolayısıyla raf düğümlerine uğramadan o rafa konulacak kitabın bulunduğu masa düğümüne uğramak gerekmektedir. Yani bazı düğümlere uğramadan önce ziyaret edilmesi gereken düğümler mevcuttur, bu sebeple düğümler arasında öncelik ilişkisi mevcuttur. Bu problem literatürde ‘Sequential Ordering Problem (SOP)’ yani ‘Ardışık Sıralama Problemi (ASP)’ olarak isimlendirilmektedir.

Geliştirilen tüm senaryolarda bu probleme çözüm aranmaktadır. Problemi çözmek için sezgisel ve meta-sezgisel yöntemler önerilmiştir. Bu bağlamda, gezgin satıcı problemi, araç rotalama problemi gibi benzer problemlerin çözümü için literatürde kullanılan En Yakın Komsu Sezgiseli (EYKS), Süpürme Algoritması gibi sezgisellerin yani sıra Parçacık Sürü Optimizasyonu meta-sezgiseli de kullanılmıştır. Ayrıca bu çalışmaya özgü EYKS’nin bir modifikasyonu olarak değerlendirilebilecek Bölgesel EYKS (BEYKS) adı verilen sezgisel bir çözüm yöntemi geliştirilmiştir. Toplam 5 farklı çözüm yöntemi senaryolara uygulanmış ve performansları karşılaştırılmıştır. Senaryolar da kendi aralarında karşılaştırılarak kütüphane için en iyi toplama ve yerleştirme senaryosu elde edilmeye çalışılmıştır.

Literatürde kütüphaneler ile ilgili yapılan optimizasyon çalışmaları araştırılmış fakat tezde Kitap Toplama Yerleştirme Problemi (KTYP) olarak isimlendirilen kütüphane içerisinde kitap toplanması ve yerleştirilmesi esnasında oluşacak toplam gezinti mesafesinin minimize edilmesini amaçlayan herhangi bir çalışmaya rastlanmamıştır.

Literatürde bulunan sınırlı sayıdaki çalışmalardan aşağıda bahsedilmiştir.

Min (1989) yaptığı çalışmada kütüphanelerle ilgili bir Eş Zamanlı Topla-Dağıt Problemini ele almıştır. Bu çalışmada, Columbus Halk Kütüphanesi ve Franklin County kütüphanelerine ait kütüphane materyallerini (kitaplar, filmler, video kasetler, kutular, zarflar vb.) 22 şubeye eş zamanlı dağıtılması ve toplanması matematiksel olarak modellenmiştir. Benzer bir problem de Francis ve ark. (2006) tarafından ele alınmıştır. Chicago'nun kuzeyindeki banliyölerde bulunan 50 üye kütüphaneye 4 araçla kütüphaneler arası ödünç kütüphane materyali sağlayan Perodik Araç Rotalama Problemi incelenmiştir. Tuna ve ark. (2013), kütüphanelerin dijitalleşme optimizasyonu hakkında makale ele almış ve dijitalleşmelerde harcanan çaba, zaman ve maliyeti azaltmak için kullanılan dijitalleşme yazılımların verimliliğini arttırmak için önerilerde bulunmuşlardır.

Bilindiği kadarıyla bu tez, KTYP’yi ilk kez ele alan bir çalışma olacaktır. Ayrıca, KTYP optimizasyonu tez kapsamında geliştirilen bir Karar Destek Sistemi (KDS)

içerisinde yapılmaktadır. Kütüphaneler için KTYP gibi optimizasyon problemlerini çözen herhangi bir karar destek sistemine de literatürde rastlanmamıştır. Geliştirilen KDS, literatürde ve gerçek hayatta KTYP optimizasyonu yapabilen ilk çalışma olma özelliğine sahiptir.

Konu hakkında detaylı literatür araştırması tezde; Bölüm 2’de verilmiştir. Bölüm 3’te geliştirilen KDS, senaryolar, çözüm yöntemleri ve problemin Konya Teknik Üniversitesi Kütüphanesi (KTUNK) için uygulanması detaylı olarak anlatılmıştır. Elde edilen tüm araştırma sonuçlar 4. Bölüm ’de sunulmuştur. Bölüm 5’te ise tezde elde edilen genel sonuçlar özetlenerek, tezdeki çalışmanın iyileştirilmesi üzerine çeşitli öneriler yapılmıştır.



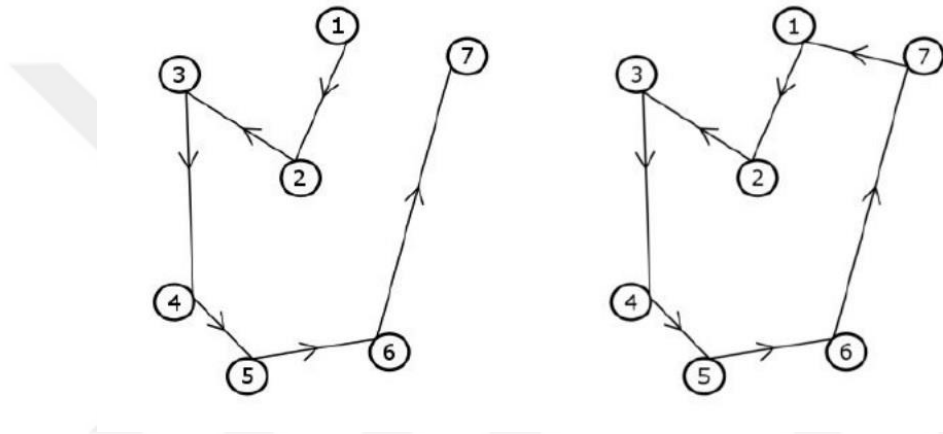
2. KAYNAK ARAŞTIRMASI

KTYP’de masalar ve raflar birer düğümdür. KTYP senaryolarında bir kitap ilgili masadan alınmadan ait olduğu rafa yerleştirilemez. Dolayısı ile önce ilgili masaya sonrasında ilgili rafa gidilebilir. Burada düğümler arası bir öncelik ilişkisi bulunmaktadır. Senaryolarda ilgili tüm düğümlerin ziyaret edildiği ve öncelik ilişkilerine uyulduğu en kısa Hamilton yolu bulunması gerekmektedir. Statik ve dinamik senaryolar arasındaki farklılıklar nedeniyle problem türleri birbirlerine benzerlik gösterse de biraz farklıdır. Bu farklılıklar giderilerek KTYP’nin tüm senaryolarında tek bir problem türünü ele alarak çözüm aramaktadır.

Statik senaryolarda karar destek sistemleri veya görevlilerin danışma masasındaki kitapların yanı sıra hangi çalışma masasında kitap hangi rafa var ve hangi rafa yerleştirileceği gibi kritik soruların cevaplarını masaya uğramadan bildiği varsayılmaktadır. Bu durumda danışma masasında başlayıp danışma masasında biten öncelik ilişkilerini dikkate alan en kısa rotayı bulmak hedeflenmiştir. Bu problem literatürde Öncelik Kısıtlamaları olan Gezgin Satıcı Problemi (ÖKGSP) olarak adlandırılır.

Dinamik senaryolarda, karar destek sistemleri veya görevliler, hangi masada hangi kitabın olduğunu bilmemektedir. Dolayısı ile görevliler danışma masasında bulunan iade edilmiş kitapları ait oldukları rafa yerleştirirken, çalışma veya toplama masalarını da ziyaret ederek bu masalarda kitap olup olmadığını kontrol etmek zorundadırlar. Eğer ziyaret ettiği çalışma veya toplama masasında kitap var ise ve bu kitabın ait olduğu raf mevcut gidilecek düğümler arasında değil ise kitabı aldığı masada başlayan ve danışma masasında biten bir rota planlaması yapmaktadır. Bu durumda ilk çözülecek olan problem; Danışma masasında başlayan ve biten, gerekli bütün düğümlere gidilen bir klasik Gezgin Satıcı Problemi (GSP). Çünkü başlangıç düğümünde olduğu için yerleştirilmesi gereken kitapların rafları ve danışma masası arasında herhangi bir öncelik ilişkisi kalmamıştır. Görevli ilk çözümde elde ettiği rotada devam ederken, bir masada kitapla karşılaştığı zaman elindeki barkod okuyucu cihaz sayesinde kitabı okutarak yeni bir rota elde edecektir. Oluşan yeni rota, kitabı aldığı çalışma masasında başlayan; danışma masasında biten ve ilgili tüm düğümlere uğrayan bir Hamilton turudur. Bu durumda çözülmesi gereken problem Ardışık Sıralama Problemidir (ASP). ASP İngilizce literatürde Sequential Ordering Problem (SOP) olarak adlandırılır.

ASP ve ÖKAGSP arasındaki tek fark; ÖKAGSP seçilen başlangıç ve bitiş düğümü aynı iken ASP için seçilen başlangıç ve bitiş düğümleri birbirinden farklıdır (Alves, 2018). ÖKGSP olarak ele alınan senaryolarda ufak bir değişiklik yaparak ASP'ye çevirmek mümkündür. Bu değişiklik ise danışma masasının bir kopyasını oluşturup bitiş düğümü olarak seçilmesini sağlamaktır. Yani ÖKAGSP'ler ve GSP'ler danışma masasında başlayan, kopya danışma masasında biten bir ASP'ye çevrilebilir. Bu çalışmada da bu varsayımıyla çözümler elde edilmiştir. Çok ufak bir fark olduğu için literatürde çoğu kaynak tarafından çok dikkate alınmayarak aynı problem çatısı altında incelenmiştir. Alves (2018), Şekil 2.1'deki gibi her iki probleme de örnek vermiştir.



Şekil 2.1. ASP (sol) ve ÖKAGSP (sağ) örnek (Alves, 2018)

2.1. Ardışık Sıralama Problemi (ASP) ve Öncelik Kısıtlamaları Olan Gezgin Satıcı Problemi (ÖKGSP)

Literatürde ASP ve Öncelik Kısıtlamaları olan Gezgin Satıcı Problemi (ÖKGSP), çeşitli alanlarda uygulamaları olan bir NP-Zor problemidir. Sorun basitçe aşağıdaki gibi açıklanabilir. Düğümler arasındaki maliyeti temsil eden bir ağırlık (uzaklık, maliyet vb.) grafiği ve bir öncelik grafiği verildiğinde, ağırlık grafiğinde öncelik grafiğindeki öncelik kısıtlamalarını karşılayan minimum maliyetli bir Hamilton turu bulmaya çalışır. Bir Hamilton turu, şebekedeki her düğümü tam olarak bir kez ziyaret eden basit bir yoldur. ASP, iyi bilinen bir NP-zor problem olan GSP'den daha genel olduğu için, ASP'de doğal olarak NP-zor problemidir (Shobaki ve Jamal, 2015). Daha önce de belirtildiği üzere klasik GSP'ler ve ÖKGSP'ler de ASP olarak ele alınabilir.

ÖKGSP terimi ilk kez Kusiak ve Finke (1987) tarafından kesin çözüm yöntemi ile makine çizelgeleme problemini çözmek için 1987 yılında kullanıldı. GSP turlarını karakterize etmek için iki emtialı ağ akış modeli kullanılmıştır. Bu çalışmada önerilen doğrusal model pratikte kabul edilebilir olduğu, Dal-Sınır yöntemleri ile kolayca geliştirilebildiği gözlemlenmiştir.

Ardışık Sıralama Problemi (ASP) olarak adlandırma ise ilk olarak Escudero (1988)'de kullanılmıştır. Bu çalışmada önerilen algoritma 2 farklı varsayım üzerinde değerlendirilmiş, bunlardan biri öncülleri atanmış düğümün öncüllerinden hemen sonra atanması durumu ikincisi ise hemen atanmaması durumudur. Önerilen algoritma kesin sonuç veren bir algoritma değilse de zaman zaman optimal veya optimale yakın sonuçlar verdiği gözlemlenmiştir. Ardından Ascheuer ve ark. (1993)'de 0-1 tam sayılı bir matematiksel model önermişlerdir. Önerilen modelin Escudero (1988)'de verilen daha etkili olduğu gözlemlenmiştir. Daha sonrasında yayımlanan Escudero ve ark. (1994), ASP için tam sayılı bir model önerilmiştir.

Balas ve ark. (1995), ÖKGASP üzerinde öncül ardıl kısıtları ve iki düğüm arasındaki "arc" yay gibi bazı eşitsizlikler üzerinde durarak çözüm uzayını azaltarak çok düzlemli bir algoritma üzerinde çalışmışlardır

Moon ve ark. (2002), ASP'de öncelik ilişkilerine uygun çözümler üretebilmek için topolojik sıralama tekniklerini entegre ederek etkili bir algoritma sunmuştur. Bu çalışmada Kusiak ve Finke (1987)'de önerilen iki emtialı ağ akış modelini kullanarak yeni bir model geliştirmiş, küçük ve orta ölçekli problemlerde optimal sonuçlar elde etmişlerdir. Ayrıca büyük ölçekli ASP'ler için geliştirdikleri Genetik Algoritma ile optimal sonuçların da elde edildiği gözlemlenmiştir.

ÖKAGSP ve ÖKGSP arasındaki fark ise ÖKAGSP asimetrik bir yapıya sahip olmasıdır. ÖKAGSP için geliştirilen bazı yaklaşım ve çözüm yöntemleri ÖKGSP içinde kullanılabilir. Sherali ve Driscoll (2002) tarafından Miller-Tucker-Zemlin (MTZ) alt tur eliminasyon yöntemini kullanarak temel Asimetrik GSP üzerine kurulmuş, ÖKAGSP ile ilgili iki model geliştirmişlerdir. Bu modeller çözüm uzayındaki dallanmayı azaltsa da büyük problemleri çözmekte oldukça zorlanmakta veya çözememektedir.

Gambardella ve Dorigo (2000) tarafından ASP için ASP-3 değişimi adı verilen yerel arama sezgiseli geliştirmişlerdir. Bu sezgisel, Alfabetik Arama Sezgiseli'nin öncelik kısıtlarını dikkate alması ile geliştirilmiş yeni bir etiketleme algoritmasıdır. Karınca Kolonisi Optimizasyonu (KKO)'ya entegre edilmiş ve ASP çözümü için kullanmışlar. Sonuçlara göre etkili bir çözüm hızına sahip algoritma ASP-3 değişimi

sezgiseli ile üretilen çözümlerde öncelik ilişkileri kısıtlarından kaynaklı zaman kaybı yaratmadığı gözlemlenmiştir (Gambardella ve Dorigo, 2000). Daha sonra 2011 yılında Anghinolfi ve ark. (2011) tarafından ASP-3 değişimi olarak adlandırılan yerel arama sezgiselini Parçacık Sürü Optimizasyonu (PSO)'ya entegre edilmiş ve Gambardella ve Dorigo'nun geliştirdiği "HAS-SOP" olarak adlandırılan KKO algoritması ile karşılaştırılmıştır. Yapılan karşılaştırılmada geliştirilen PSO, HAS-SOP ile benzer sonuçlar verse de çözüm zamanı açısından HAS-SOP kadar etkili olmadığı gözlemlenmiştir.

Shobaki ve Jamal (2015) tarafından kod derleyicileri (C, C++, LLVM gibi) için anahtarlama optimizasyonu ve yüksek CPU tüketimini en aza indirmek için ASP için Dal-Sınır Algoritması sunmuştur. Aynı zamanda anahtarlama optimizasyonunda yöneylem çalışmalarının fazla dikkate alınmadığı da vurgulanmıştır. Sunulan algoritma LLVM (Low Level Virtual Machine) derleyicisi ve ARM işlemcisi üzerinde test edilmiş, EYKS gibi geleneksel sezgisel yöntemlere göre %99,8 daha iyi olduğu, enerji tüketiminin de daha az olduğunu gözlemlemişlerdir.

Dewil ve ark. (2019) tarafından yapılan çalışmada Çoklu Delik Delme Probleminde sıralı delme görevleri bulunduğu durumlar için ASP modeli uygulamış ve ASP için geliştirilen çözüm yöntemlerinin geliştirilmesi bu tür problemler açısından önemli olduğunu vurgulamışlardır.

Literatürde ÖKGSP veya ASP'nin kapasiteli versiyonları da incelenmiştir. Hernández ve Salazar (2009) tarafından Topla-Dağıt problemlere uyarlanmış bir matematiksel model sunmuştur. Bu topla dağıt problemi temel olarak Kapasiteli ASP'ye benzemektedir. Diğer Topla-Dağıt Problemleri (TDP)'nin de alt problemlerinden biri ASP'dir. Ayrıca Letchford ve Salazar (2016) bu modeli geliştirerek test etmişler ve Hernandez ve Salazar'ın 2009'daki çalışması ile karşılaştırmışlardır. Sonuç olarak daha iyi doğrusal gevşetmelere sahip olduğu ve çözülememiş bazı problemlerde optimal sonuç elde ettikleri gözlemlenmiştir. Yine de daha iyi ve hızlı sonuçlar elde edebilmek için bazı önerilerde bulunmuşlardır.

ÖKGSP'ye öncelik ilişkilerine ek olarak zaman kısıtları da eklendiğinde, literatürde Zaman Kısıtlı ÖKGSP (ZK-ÖKGSP) olarak adlandırılır. Bianco ve ark. (1994) tarafından ZK-ÖKGSP için kesin sonuç veren matematiksel model ve sezgisel geliştirmişlerdir. Ardından Mingozzi ve ark. (1997) tarafından bir dinamik programlama algoritması geliştirilmiştir. Geliştirdikleri bu algoritma, (Bianco ve ark., 1994)'deki algoritmadan daha iyi sonuçlar verdiğini gözlemlemişlerdir. Fagerholt ve Christiansen

(2000) tarafından ZK-ÖKGSP ile ilgili bir algoritma önerdi. Algoritmanın uygulaması gemi çizelgelemesinde kullanılmıştır. ZP-ÖKGSP, zaman pencereleri ve çoklu tahsis problemi ile birleştirilmiş çok gemili teslim alma ve teslim etme problemi olan gerçek bir dökme gemi çizelgeleme probleminde verilen bir dizi liman ziyaretini en iyi şekilde sıralamanın bir alt problemi olarak ortaya çıkar. Filodaki her bir gemi, belirli sayıda küçük ambarlara bölünebilen esnek bir kargo ambarıyla donatılmıştır, böylece aynı gemi tarafından birden fazla ürünün aynı anda taşınmasına izin verilmektedir. ZK-ÖKGSP'nin tahsis kısıtları, geminin esnek kargo ambarının bölümlenmesinin ve kargoların daha küçük ambarlara tahsisinin ziyaret dizisi boyunca uygulanabilir olmasını sağlar. ZK-ÖKGSP, düğümleri yoldaki düğüm kümesini, en son ziyaret edilen düğümü ve birikmiş kargo tahsisini temsil eden durumlar olan bir grafik üzerinde en kısa yol problemi olarak çözülür. Şebekedeki hatlar (oklar) bir durumdan diğerine geçişleri temsil eder. Algoritma ileri dinamik programlama algoritmasıdır. Durum uzayını azaltmak için bir dizi eleme kısıtları uygulanmıştır. Hesaplama sonuçları, ZK-ÖKGSP için önerilen algoritmanın çalıştığını ve gerçek gemi çizelgeleme problemine optimum çözümler elde edildiğini göstermektedir.

ASP'ye kapasite ve öncelik kısıtlarına ek olarak bir de zaman kısıtları eklendiğinde problem literatürde "Dial-a-Ride" olarak bilinen Araç Çağırma Problemi (AÇP)'ye benzerlik göstermektedir. Araç çağırma hizmeti, özellikle hasta taşıma veya kamu hizmetleri olarak evden eve bir taşıma hizmeti olarak ortaya çıktı. 1978 yılında AÇP için Stein tarafından hem dinamik hem statik versiyonları modellendi (Stein, 1978). Psaraftis (1980) tarafından AÇP'nin her iki versiyon için kesin sonuç veren dinamik programlama algoritması geliştirmiştir. Literatürde AÇP için yüzlerce çalışma mevcuttur. Daha kapsamlı AÇP araştırması için (Ho ve ark., 2018) incelenebilir.

2.2. KTYP için Kullanılan Çözüm Yöntemleri

Bu çalışmada herhangi bir kapasite veya zaman kısıtlaması olmadığı içi klasik ASP olarak ele alınmış ve geliştirilen sezgisel ve metasezgisel yöntemlerle senaryolar çözümlenip karşılaştırılmıştır. Uygulanan sezgisellerden birisi Bölgesel En Yakın Komşu Sezgiseli (BEYKS) ise En Yakın Komşu Sezgiseli (EYKS)'nin bir versiyonu olarak bu çalışmaya özel ve literatürde bulunan sezgisellere ilave olarak geliştirilmiştir.

Sezgisel ve metasezgisel yöntemler arasında ki fark metasezgiseller her probleme uyarlanabilen yöntemlerken, sezgiseller ise probleme özgü geliştirilmiş çözüm

yöntemleridir. Her iki yöntemin amacı mümkün olduğunca kısa zamanda optimal veya optimale yakın sonuçlar elde edilmesini sağlamak olarak değerlendirilebilir. Çünkü ASP gibi NP-Zor olan problemlerde kesin sonuç veren matematiksel modellerle kısa sürelerde optimal sonuç elde etmek oldukça zordur. Bu zorluğu aşmak için genelde problem çözücü kişiler tarafından NP-Zor problemlere sezgisel veya metasezgisel yöntemler geliştirilir.

Bu çalışmada KTYP için uygulanan çözüm yöntemleri literatürde araştırılmış önemli görülen çalışmalardan bahsetmiştir. Ayrıca bu çözüm yöntemlerinin KTYP'ye uyarlanması detaylı şekilde tezin 'MATERYAL VE YÖNTEM' kısmında anlatılmıştır.

2.2.1 En Yakın Komşu Sezgiseli (EYKS)

EYKS literatürde birçok NP-Zor probleme uygulanan oldukça basit bir sezgiseldir. Bu sezgisel çoğu zaman optimal sonuç vermese de genel olarak hızlı ve iyi sonuçlar vermektedir.

EYKS çalışma mekanizması genel olarak şu şekildedir;

- $C_{i,j} = i$ ve j düğümleri arasındaki uzaklık, süre gibi maliyeti versin.
- Tüm düğümlerin birer kez ziyaret edileceği varsayılır.
- İlk ziyaret edilen düğüm probleme göre ya rastsal olarak seçilir ya da belirlenir.
- Ziyaret edilen son düğüm i ise ; i' 'den sonra i' 'den en az maliyete ulaşılabilir olan j düğümü ziyaret edilir. Burada j düğümünün, i düğümünden hemen sonra ziyaret edilebilir olması gerekmektedir. Tüm düğümler birer kez ziyaret edilene kadar bu işlem devam eder.

ASP gibi bazı problemlerde, bazı düğümlerin ziyaret edilebilmesi için bazı düğümlerin önce ziyaret edilmiş olması gerekmektedir. Dolayısı ile burada uygun bir çözüm bulabilmek için en son ziyaret edilen düğümden sonra tüm öncülleri ziyaret edilmiş düğümler arasından, son ziyaret edilen düğüme en yakın düğüm ziyaret edilir. Yani öncelik ilişkileri dikkate alınır.

Lin ve Hsueh (1994), GSP için geliştirdikleri Tavlama Benzetimi meta sezgiselini EYKS ile hibrit bir şekilde kullanmış ve düşük sıcaklıkta klasik tavlama yöntemlerinden daha iyi sonuçlar elde etmişlerdir

Kizilateş ve Nuriyeva (2013) tarafından GSP için EYKS'nin bir modifikasyonunu geliştirip aç gözlü hibrit bir sezgisel geliştirmişler. Geliştirdikleri bu sezgiselin verimli sonuçlar elde ettiği gözlemlenmiştir.

Khan ve Agrawal (2016), GSP için geliştirdikleri Genetik Algoritmayı EYKS ile karşılaştırmışlardır.

Asani ve ark. (2020) tarafından EYKS ve dış bükey gövde oluşturma tekniği ile GSP için hibrit bir sezgisel geliştirmiştir. Geliştirilen hibrit EYKS ise klasik EYKS'ye göre daha iyi sonuçlar vermiştir.

Dorigo ve Gambardella (1997), GSP için geliştirdikleri Karınca Kolonisi Optimizasyonu (KKO) algoritmasında ilk feromonun değerini EYKS ile üretilen çözümden elde etmiştir. Skinderowicz (2015) tarafından ASP için geliştirdiği KKO algoritmasında ilk feromonun değerini, (Dorigo ve Gambardella, 1997) çalışmasındaki gibi EYKS ile üretilen çözümden elde etmiştir.

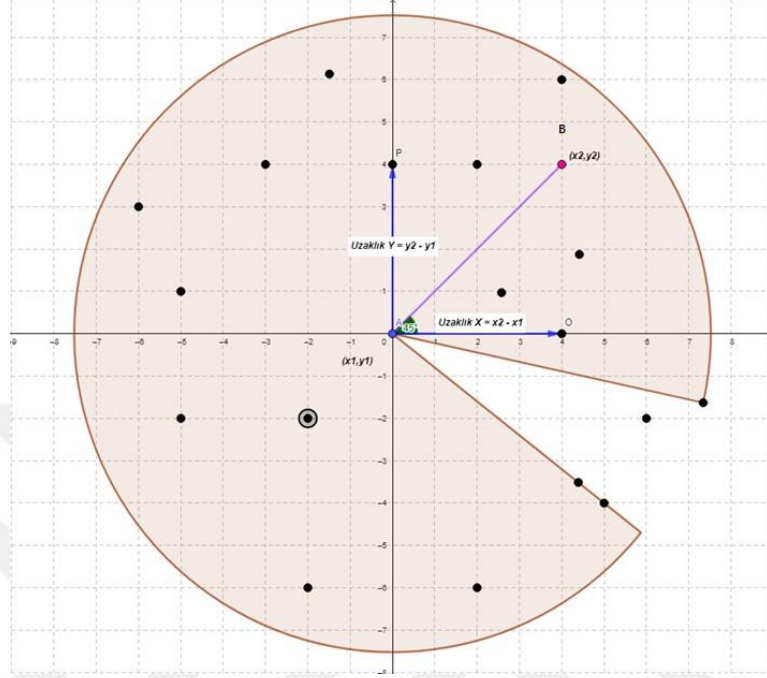
Genelde literatürdeki uygulamalarda genel olarak EYKS'nin modifikasyonu veya EYKS ile hibrit şekilde kullanılan algoritmalar geliştirilmiştir. EYKS, genelde optimizasyon problemleri için geliştirilen diğer algoritmalar ile karşılaştırma amaçlı çalışmalarda yer almıştır.

2.2.2 Süpürme Algoritması (SÜP_A)

Süpürme Algoritması (SÜP_A), ilk olarak Gillett ve Miller tarafından 1974 yılında araç rotalama problemlerine uygulandı. Uygulanan bu algorithmada tüm düğümler iki boyutlu koordinat sisteminde ve kutupsal (x,y) koordinatlarına sahiptir. Tüm düğümlerin referans düğüme veya koordinata olan açısı hesaplanarak küçükten büyüğe veya büyükten küçüğe sıralanır ve bu sıraya göre ve kısıtlar doğrultusunda düğümlere ziyaretler gerçekleşir. Bu sıralamalar sayesinde saat yönüne veya saat yönün tersine doğru hareket edilmiş olur (Gillett ve Miller, 1974). Şekil 2.2. Düğümlerin (x,y) koordinat sisteminde gösterilmesi ve açı örnek düğümlerin koordinat sisteminde gösterilmesine ve açıların hesaplanmasına örnek gösterilmiştir.

Örneğin, nokta A'nın koordinatları (x_1,y_1) ve nokta B'nin koordinatları (x_2,y_2) olsun ve bu iki nokta arasındaki açı (2.1)'de verilen denklem ile bulunur. Bu denklem, nokta B'den nokta A'ya doğru çizgi çekilen düzlemin yatay düzleme göre açısını verir.

$$Açı = \tan^{-1} \frac{y_2 - y_1}{x_2 - x_1} = \tan^{-1} \frac{dy}{dx} \quad (2.1)$$



Şekil 2.2. Düzgünlerin (x,y) koordinat sisteminde gösterilmesi ve açı örneği

Süpürme Algoritması (SÜP_A) literatürde yaygın olarak kullanılan coğrafi bir sezgiseldir. Birçok alanda yapılan optimizasyon çalışmalarında SÜP_A kullanılmıştır.

Nurcahyo ve ark. (2003) tarafından belirsiz talep, seyahat mesafesi ve yol türü ile ilgili halk otobüsü rotalama problemine çözüm olarak SÜP_A önerdiler. Amaç, belirsiz verileri yakalayabilen değiştirilmiş bir arama tekniği kullanarak rota seçimi yapmaktır.

Yousefikhoshbakht ve Sedighpour (2012) tarafından çoklu GSP için başlangıç çözümlerini SÜP_A ile üreten bir KKO algoritması geliştirmişlerdir. Önerilen algoritma ile genetik algoritma, tabu arama ve tavlama benzetimi algoritmalarının kombinasyonunun daha iyi sonuçlar vereceği görülmektedir.

Fonseca ve ark (2006) tarafından çok amaçlı optimizasyon problemlerinde en iyi çözüm setini bulabilmek amacı ile “hipervolume” hiper hacim indikatörü hesaplamasında SÜP_A önermişlerdir. Bu öneri While ve ark (2006) tarafından geliştirilen “Hypervolume by Slicing Objectives (HSO)” ile karşılaştırılmıştır. Nokta sayısı arttıkça HSO’dan daha iyi sonuçlar verdiği gözlemlenmiştir. Ayrıca 3 boyutlu durum için algoritmanın daha fazla geliştirilmesi gerektiğini vurgulamışlardır.

Çok işlemcili bilgisayarların işlemci mimarilerinin tasarımı için ele alınmıştır (McKenney ve McGuire, 2009), (McKenney ve ark., 2017).

2.2.3 Parçacık Sürü Optimizasyonu (PSO)

Reynolds (1987) ve Heppner ve Grenander (1990) yaptıkları çalışmada kuş sürülerinin hareketlerini simüle eden modeller geliştirmiş ve sürü hareketlerini anlamlandırmaya çalışmışlardır. Bu çalışmalarda her bir sürü elemanın diğer sürü elemanlarına göre değişen hızları ve pozisyonları incelenmiştir. Kennedy ve Eberhart (1995) tarafından bu çalışmalar dikkate alınarak doğrusal olmayan fonksiyonların optimizasyonu ve yapay zekada kullanılan yapay sinir ağlarının eğitimi için ilk Parçacık Sürü Optimizasyonu (PSO) algoritması geliştirilmiştir.

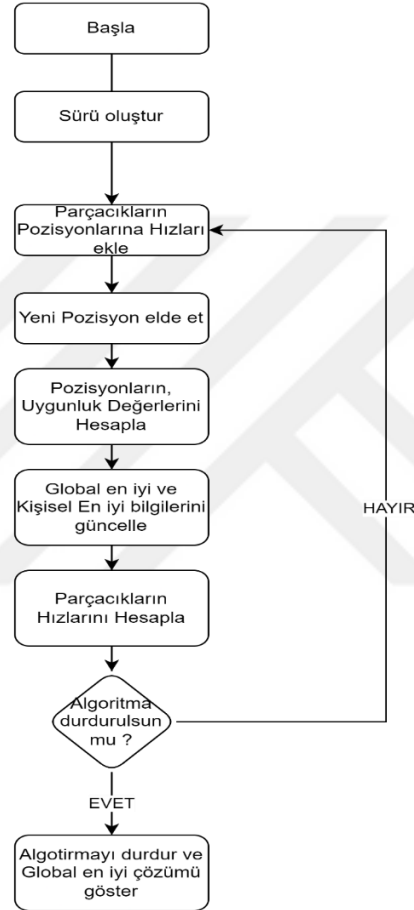
PSO meta sezgisel bir yöntem olmakla birlikte Genetik Algoritma (GA), Karınca Kolonisi Optimizasyonu (KKO) gibi evrimsel veya memetik(taklitçi) algoritmalarla da benzerlik göstermektedir. PSO da benzer evrimsel veya memetik algoritmalar gibi bir popülasyon içerisinde öğrenmeye dayalı bir optimizasyon algoritmasıdır.

PSO algoritması hakkında literatürde yüzlerce çalışma mevcuttur. Bu çalışmalardan bir kısmı PSO algoritmasını geliştirmek ve test etmek üzerine bir kısmı ise optimizasyon problemlerini çözmek üzerinedir. Problemlerin yapılarına göre uyarlanmış PSO algoritmasının onlarca modifikasyonu vardır. Tezin bu bölümünde PSO'nun ilk çalışmaları ve genellikle GSP benzeri rotalama problemlerine uyarlandığı bir dizi kaynak incelenmiştir.

PSO algoritması, böcekler, sürüler, kuşlar ve balıklar da dahil olmak üzere hayvanların sosyal davranışlarını simüle eder. Bu sürüler yemek bulmak için veya göç etmek için birbirleri ile iş birliği içindedirler. PSO algoritmasının ana tasarım fikri iki araştırma ile yakından ilgilidir: Birincisi evrimsel algoritmadır, tıpkı evrimsel algoritma gibi; PSO da optimize edilmiş amaç fonksiyonunun çözüm uzayında geniş bir bölgeyi eşzamanlı olarak aramasını sağlayan bir sürü modu kullanır. Diğeri ise yapay yaşamdır, yani yaşam özelliklerine sahip yapay sistemleri inceler. Yapay yaşam teorisi ile sosyal hayvanların davranışlarının incelenmesinde, sürü yapay yaşamının nasıl oluşturulacağı bilgisayar tarafından işbirlikçi davranışa sahip sistemler oluşturmaktadır (D. Wang ve ark 2018).

Şekil 2.3'de PSO'ya ait örnek bir akış şeması verilmektedir. PSO'da her parçacığın kendine ait bir pozisyonu ve uygunluk değeri vardır. Parçacıklar, aralarındaki

global en iyi uygunluk değeri ve global en iyi pozisyon ($Gbest$)'e göre ve kendi geçmişlerindeki kişisel en iyi pozisyon ($Pbest$) ve kişisel en iyi uygunluk değerlerine göre hızlarını belirlemekte. Fizik kuralları göz önüne alınarak elde edilen hızlar, o anki pozisyonlarına yani konumlarına eklenerek parçacığın bir sonraki iterasyondaki pozisyonu elde edilir. Genellikle ilk iterasyonda, belirlenen sayıda parçacık için pozisyonlar ve hızlar rastgele oluşturulur.



Şekil 2.3. PSO Algoritmasının Akış Şeması

Kennedy ve Eberhart (1995), bir parçacığın pozisyonunu x ve y koordinat sisteminde göstermiştir. Bir parçacığın x düzlemindeki konumunun ve hızının hesaplamasını denklem (2.2)'deki gibi yapmışlardır.

$$v_x = v_x + 2 * Rand * (Pbest - xKoordinatı) \quad (2.2)$$

$$xKoordinatı = x Koordinatı + v_x$$

Herhangi bir minimize etme problemini örnek olarak alırsak, başlangıçta PSO algoritmasının versiyonunda, kişisel en iyi pozisyonunun (P_{best}) güncellemesi ise denklem (2.3)'deki gibidir.

- D = **Boyut.**
- i = *parçacık.*
- t = *iterasyon veya zaman.*
- $X_{i,t}^D$ = *D. boyuttaki i. parçacığın t. iterasyondaki pozisyonu.*
- $v_{i,t}^D$ = *D. boyuttaki i. parçacığın t. iterasyondaki hızı.*
- $P_{i,t}^D$ = *D. boyuttaki i. parçacığın t. iterasyona kadar ki kişisel en iyi uygunluk değerine sahip pozisyon*
- $P_{g,t}^D$ = *D. boyuttaki tüm parçacıklar arasında t. iterasyona kadar ki en iyi uygunluk değerine sahip pozisyon.*
- $f(X_{i,t})$ = *D. boyuttaki i. parçacığın t. iterasyonundaki uygunluk değeri.*

$$P_{best} = \begin{cases} X_{i,t+1}^D, & \text{Eğer } f(X_{i,t+1}) < f(P_{i,t}) \\ P_{i,t}^D, & \text{Diğer durumlarda} \end{cases} \quad (2.3)$$

Denklem 2.1.'de bulunan denklemler çok ilkel olduğu için parçacıklara bilişsel c_1 ve sosyal öğrenme c_2 faktörü eklenmiş ve Denklem (2.4) ve (2.5)'de yeniden formülize edilmiştir. Genel olarak parçacığın pozisyon ve hız denklemleri aşağıdaki gibidir.

$$v_{i,t+1}^D = v_{i,t}^D + c_1 * Rand * (P_{i,t}^D - X_{i,t}^D) + c_2 * Rand * (P_{g,t}^D - X_{i,t}^D) \quad (2.4)$$

$$X_{i,t+1}^D = X_{i,t}^D + v_{i,t+1}^D \quad (2.5)$$

Shi ve Eberhart (1998), hız denklemine atalet ağırlığı olarak bilinen ivmeyi (ω) de ekleyerek 2.4. numaralı hız denklemini (2.6)'deki gibi geliştirmişlerdir. Bu geliştirmeden sonra genel olarak kullanılan PSO hız denklemini olmuştur.

$$v_{i,t+1}^D = \omega * v_{i,t}^D + c_1 * Rand * (P_{i,t}^D - X_{i,t}^D) + c_2 * Rand * (P_{g,t}^D - X_{i,t}^D) \quad (2.6)$$

Wang ve ark. (2003), PSO algoritmasını Gezgin Satıcı Problemi (GSP)'ye uyarlamış ve bu modifikasyona "*TSP_PSO*" adını vermişlerdir. Geliştirilen bu algoritma 14 düğüme sahip bir problem üzerinde test edilmiş ve optimal sonuç elde edilmiştir. Bu çalışmada bir parçacığın pozisyonu uygun bir çözümde düğüm sıralamasını, parçacığın hızı ise pozisyona uygulanan yer değiştirme operatörlerinin işlemlerini ifade eder.

Clerc (1999), iki farklı fonksiyonu optimize etmek için PSO algoritması kullanmıştır. Bu algortmada arama uzayı giderek azaldığında *umut* adı verilen işlemle seçilen *kraliçe* parçacıklarının pozisyonlarından tekrar çözümler üretmeye başlar. Daha sonra Clerc (2004), PSO'nun kraliçe modifikasyonunu asimetrik GSP'ye uyarlamıştır. 17 düğümlü problemde optimal sonuçlar elde edildiği gözlemlenmiştir.

Zhao ve ark. (2004), Zaman Pencereci Araç Rotalama Problemi (ZPARP) için iki farklı PSO geliştirmiş ve bunları Solomon eklenti sezgiselleri ile karşılaştırmışlardır. Elde edilen sonuçlara göre geliştirilen PSO'ların daha iyi sonuçlar verdiği gözlemlenmiştir. Zhu ve ark. (2006), ZPARP için bir PSO algoritması geliştirmiş ve Genetik Algoritma (GA) ile karşılaştırmışlardır. Elde edilen sonuçlara göre PSO, GA'dan daha iyi olduğunu gözlemlenmişlerdir.

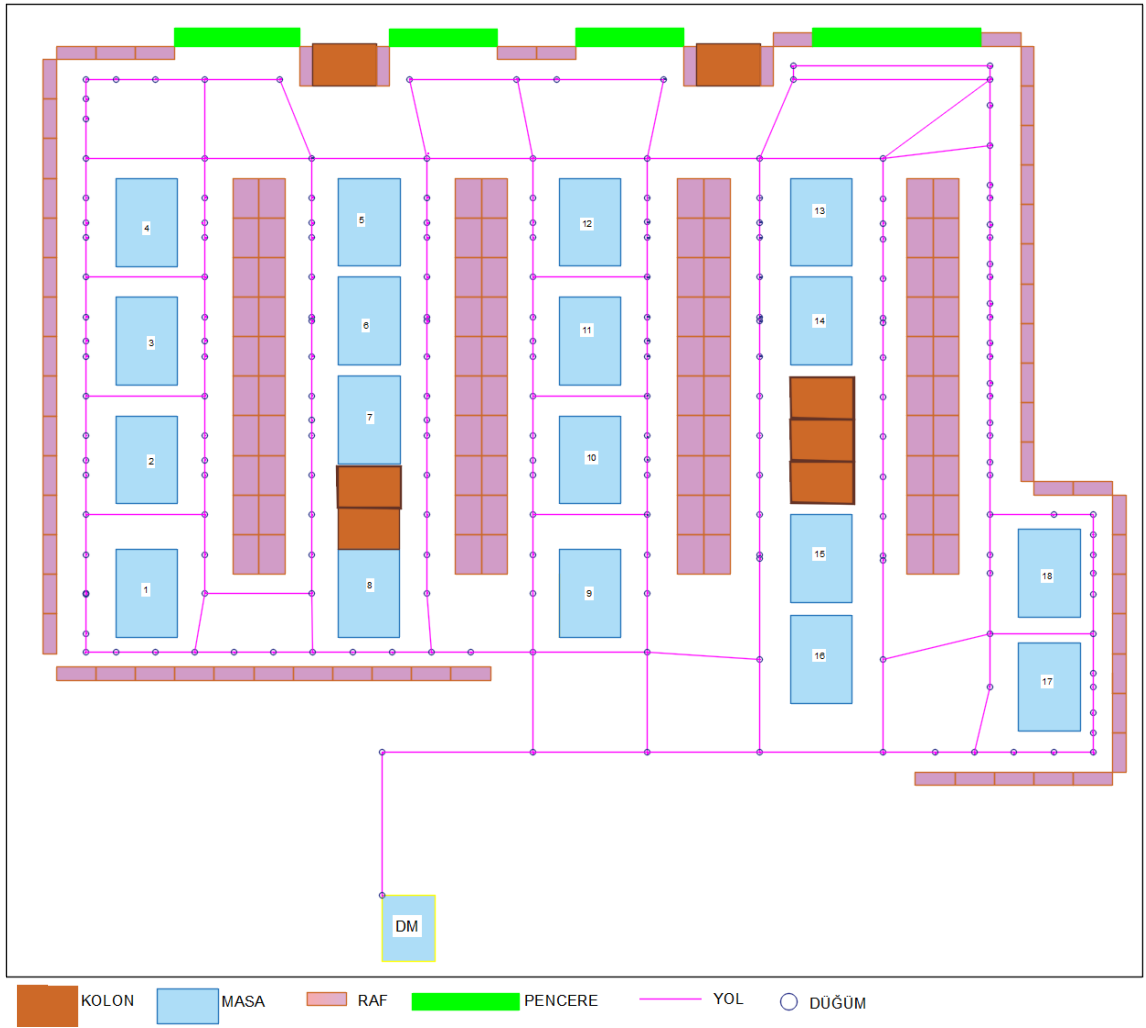
Hamta ve ark. (2013) tarafından çok amaçlı montaj hattı dengeleme problemi için PSO ve değişken komşu arama sezgiseli ile hibrit bir algoritma geliştirmişlerdir. Geliştirilen PSO ise geliştirdikleri çok amaçlı GA ile karşılaştırılmış ve daha iyi performans sergilediği gözlemlenmiştir.

Daha önce de bahsedildiği üzere Anghinolfi ve ark. (2011), ASP-3 değişimi olarak adlandırılan yerel arama sezgiselini Parçacık Sürü Optimizasyonu (PSO)'a entegre edilmiş bir algoritma geliştirmişlerdir.

3. MATERYAL VE YÖNTEM

KTYP bir kütüphane içerisinde kitap toplama ve yerleştirme problemidir. KTYP’de bir kitap, kullanıldıktan sonra masalara (arz noktası) bırakılır. Daha sonra görevli tarafından arz noktalarından alınıp ait oldukları raflara (talep noktası) yerleştirilmesi gerekmektedir.

Şekil 3.1.’de görüldüğü üzere KTUNK’e ait düğümler raflar, çalışma masaları ve danışma masalarından oluşan bir yerleşim planı verilmiştir.



Şekil 3.1. Konya Teknik Üniversitesi Kütüphanesi (KTUNK) Yerleşim Planı (Ocak, 2021)

3.1. KTYP'nin Konya Teknik Üniversitesi Kütüphanesine (KTUNK) Uyarlanması

KTYP problemi gerçek hayatta uygulamalı şekilde KTUNK'e uyarlanmıştır. KTUNK'e ait verilerin, yerleşim planının ve gerekli ekipmanların bilgileri doğrultusunda, senaryolar türetilmiş ve optimizasyon için kullanılan çözüm yöntemleri geliştirilmiştir.

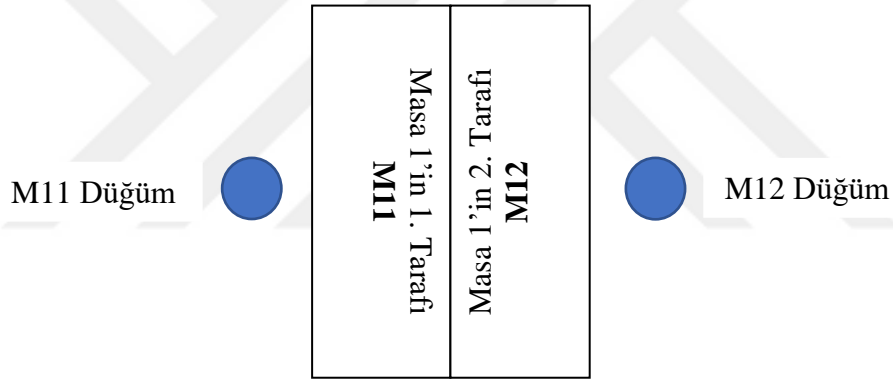
Şekil 3.1.'de KTUNK'e ait yerleşim planı verilmiştir. Bu yerleşim planı gerçeğe yakın bir şekilde AutoCAD adlı bilgisayar tabanlı tasarım programı ile çizilmiştir. Bu yerleşim planı rotalama için önemli parametrelere sahiptir. KTUNK içerisinde bulunan 141 adet raf, 18 adet çalışma masası ve 1 adet danışma masasının konumları ve birbirlerine olan uzaklıkları rotalama optimizasyonları için önemli parametrelerdir. Ayrıca bu raflarda yaklaşık 12500'e yakın kitap bulunmaktadır. Bu veriler 2021 Ocak itibarıyla elde edilmiştir.

KTUNK'de bulunan çalışma masalarının özel bir durumu vardır Şekil 3.2.'de görüldüğü üzere çalışma masaları oldukça büyüktür. Bu masalar 4 kişilik ve ortalarından bir paravan geçerek iki eşit parçaya ayrılmaktadır. Kitap toplama esnasında görevli masanın bulunduğu tarafından diğer tarafına ulaşamaz. Bu bilgiler doğrultusunda KTYP'yi KTUNK'e uyarlarken tüm çalışma masaları iki farklı masa olarak değerlendirilmiştir. Bu nedenle 18 tane olan çalışma masası ikiye katlanarak 36'ya yükselir. Masaların her bir tarafı ayrı birer masa olduğu için farklı düğümlerdir. Şekil 3.3.'te masaların bölünmesinden kaynaklı oluşan iki tarafın düğümleri gösterilmiştir. Senaryo 4, 5, 7, 8 de bulunan toplama masası ise KTUNK'de bulunan 10 numaralı çalışma masasının birinci tarafıdır. Bu masanın toplama masası olarak seçilmesinin nedeni ise kütüphanenin tam ortasında bulunmasıdır.

KTUNK'de bulunan her bir raf 5 kısımdan oluşur. Ancak, görevli rafın orta noktasında durarak bütün kısımlara ulaşabilir. Bu durumda rafların kısımları önemli değildir. Ziyaret edilecek raf düğümleri rafın orta noktası olarak ele alınır.



Şekil 3.2. KTUNK'E ait bir fotoğraf



Şekil 3.3. KTUNK çalışma masalarının birinci tarafı (sağ) ve ikinci tarafı (sol)

Şekil 3.1.'de gösterilen yollar ise kitap taşımak için kullanılan aracın kütüphane içerisinde hareket edebileceği yolları göstermektedir. Yollar üzerinde bulunan düğümler ise masalar, raflar ve yolların bağlantı noktalarıdır. Bir düğümün başka bir düğümlerle birleşiminden oluşan yolların uzaklıkları teknik çizimden elde edilmiştir. Doğrudan bağlantısı olmayan düğümlerin birbirlerine en yakın uzaklığı ise birbirlerine doğrudan bağlantısı olan düğümler üzerinden Dijkstra Algoritması uygulanarak elde edilmiştir. Toplam 195 düğümün birbirine olan en kısa mesafesi bu tezde kullanılacak olan maliyet matrisini temsil etmektedir. Tablo 1'de örnek olarak danışma masası ile 79 numaralı raf arasında en kısa yol ve bu yolun uzaklığı verilmiştir. Tüm düğümlerin birbirlerine olan uzaklıklarını gösteren matris veri tabanında tutulmaktadır. Ayrıca AutoCAD'de yapılan vektörel çizim ile Şekil 3.1.'e yaklaştırıldığında Şekil 3.4.'deki gibi düğüm numaraları gözükmemektedir.



Şekil 3.4. KTUNK Yerleşim Planı çiziminde düğüm numaraları (örnek)

Tablo 1. Düğümlerin birbirine en yakın uzaklığı gösterimi (örnek)

i	j	i düğüm açıklaması	j düğüm açıklaması	Uzaklık (cm)	En kısa yol
0	98	Danışma Masası (DM)	79 Numaralı Raf (R79)	1739	0,8,9,1,30,41,51,62,71,80,83,98

KTUNK içinde bulunan düğümlerin koordinatları Tablo 2'deki gibidir. Bu tablonun da tamamı veri tabanında tutulmaktadır. Koordinatların tutulması KTYP'de kullanılacak olan Süpürme Algoritması (SÜP_A) için gereklidir. Ayrıca tüm düğümlerin birbirine olan açısı veri tabanında tutulmaktadır.

Tablo 2. Düğümlerin (x,y) Koordinatları (1/3 oranda örnek)

Düğüm	X Koordinatı	Y Koordinatı
DM	300	60
R79	75	390

Tablo 3'te kütüphanede bulunan kitapların adlarını, numaraları, kitap kodları, yayın yılı, yazar adı ve ait olduğu rafları içeren örnek bir tablo verilmiştir. Ayrıca kitapların tamamı ve barkod bilgileri dahil kitaplara ait bütün bilgiler veri tabanında tutulmaktadır. Bu sadece görevli kitapların üzerinde bulunan barkodu okuttuğu zaman hangi rafa ait olduğu bilgisini öğrenebilmektedir. KTYP için optimizasyon işlemleri yapılırken kitapların ait oldukları rafları, rafların koordinatlarını ve bir başka noktaya olan uzaklıklarını hesaplayabilmektedir. KTUNK'de kitapların ait oldukları rafların bilgileri mevcut karar destek sistemlerinde tutulmamaktadır. Bu veriler, iki ay boyunca düzenli olarak yapılan kütüphane ziyaretleri ile özenle toplanmıştır.

Tablo 3. KTUNK kitaplarına ait bilgiler (örnek)

ID	Kod	Yayın Yılı	Adı	Yazar	Raf
15202	PG 3327/D67	2019	Yeraltından notlar	Dostoyevsky, Fyodor	R1
12676	B 3144/S34	2019	Aşkın metafiziği	Schopenhauer, Arthur	R96
16566	PL 248/A75	2019	Tutunamayanlar	Atay, Oğuz	R9
13382	DR 592/A83	2006	Nutuk : 1919-1927	Atatürk, Mustafa Kemal	R126

KTUNK'e ait verilerin Microsoft SQL Server 2014 Management Studio adlı veri tabanı programında tutulmaktadır.

3.2. Geliştirilen Karar Destek Sistemi Yapısı

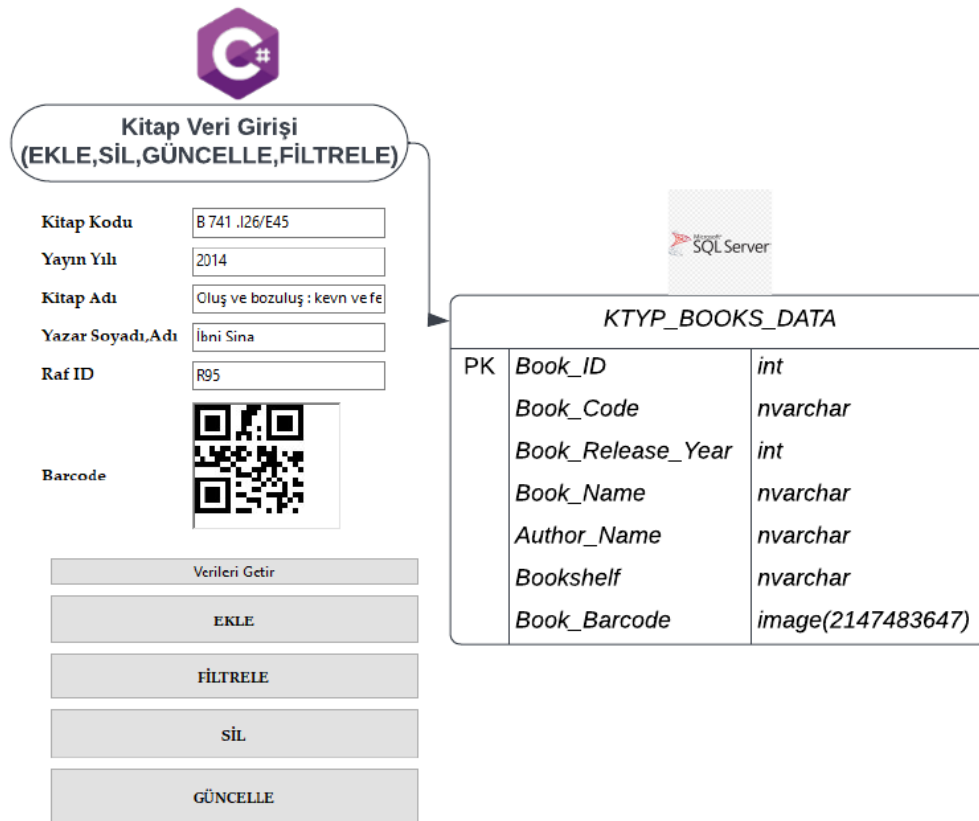
Bu bölümde KTUNK verileri kullanılarak, KTYP için geliştirilen (KDS) detaylı olarak anlatılacaktır.

Bu tez kapsamında geliştirilen KDS, kütüphaneler için geliştirilen diğer programların yanı sıra kitapların ait oldukları raf bilgilerini ve kullanıldıysa eğer hangi masada olduğunu takip edebilmesi üzerine tasarlanmıştır. Ayrıca kütüphane içerisindeki yolların haritalaması da bu KDS içerisinde yer almaktadır. Bu sayede, toplama ve yerleştirilme işlemlerinde kütüphane görevlilerinin bir düğümden başka bir düğüme giderken kullanabileceği en kısa yollar tespit edilebilir hale gelmiştir. Ayrıca kütüphane içerisinde bulunan her bir masanın ve rafın kütüphane içerisindeki yeri, koordinatı, diğer düğümlere olan mesafesi ve diğer düğümlere gidebileceği en kısa yol bilgisi de veri tabanında tutulur. Bu sayede KTYP'de ele alınan optimizasyon problemlerinin çözülmesi için geliştirilen algoritmaların ihtiyaç duyduğu veri akışı sağlanmaktadır.

Geliştirilen KDS'de kullanılan veri tabanı programı Microsoft SQL Server Management Studio 2014 (SSMS). Arayüz ve algoritmaların kodlaması ise Microsoft Visual Studio (MVS) entegre geliştirme ortamında (IDE) C# .NET 6.0 dili kullanılarak geliştirilmiştir. Programda algoritmaların ihtiyaç duyduğu birçok prosedür ve veriler SQL dili kullanılarak veri tabanında gerçekleştirilmektedir.

3.2.1 Kitap bilgilerinin (Ekle-Sil-Güncelle-Filtrele) işlemleri

Bu bölümde geliştirilen KDS'nin kitap bilgilerinin eklenmesi, silinmesi, barkod oluşturulması ve güncellenmesi prosedürleri için veri akışı işlemleri anlatılmıştır.

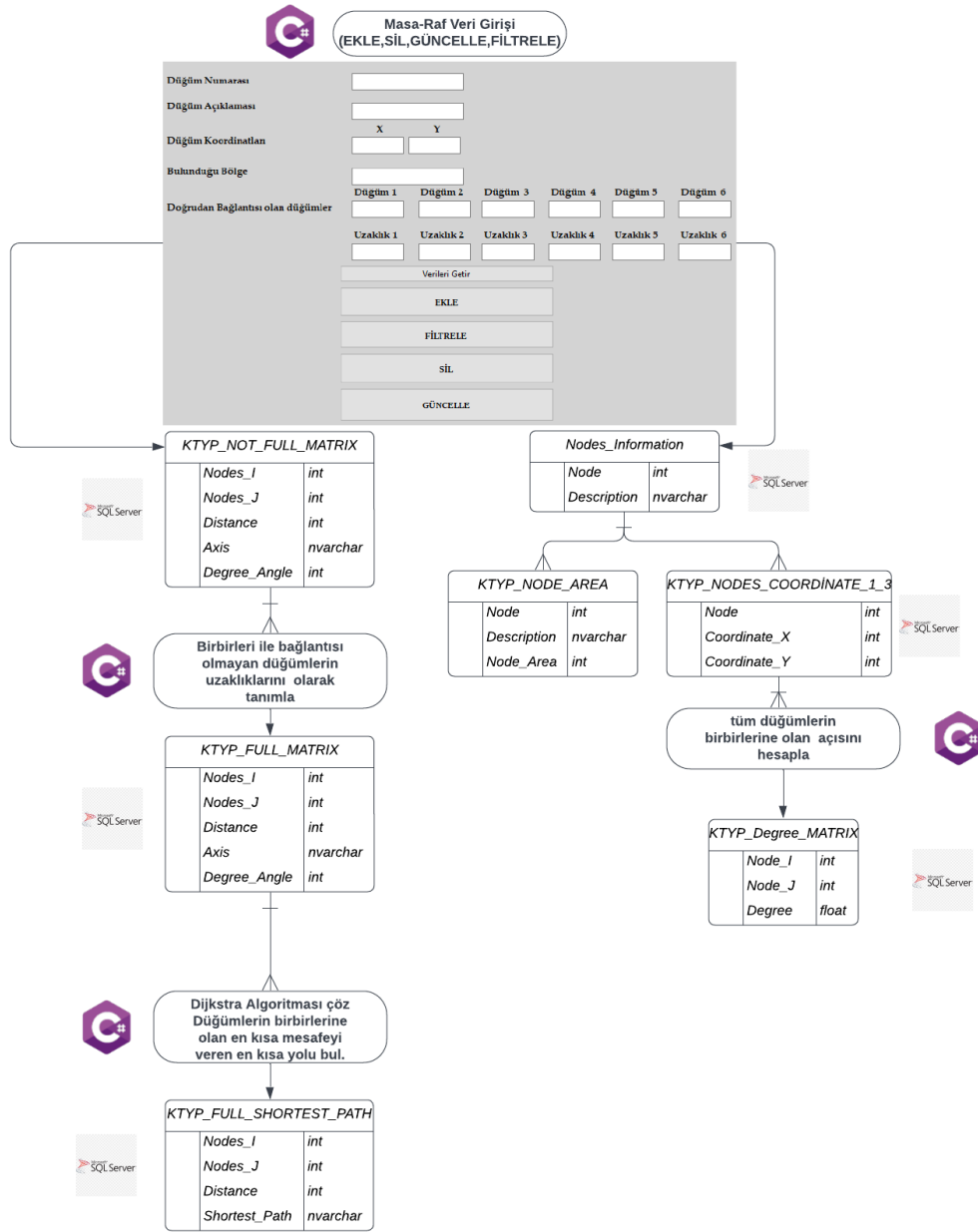


Şekil 3.5. Kitap veri girişi veri akışı

Kitap bilgileri üzerinde yapılan ekle, sil, güncelle, filtrele gibi veri tabanı işlemlerinin, MVS ortamında tasarlanan form sayesinde kullanıcıdan alınıp veri tabanına aktarılmasındaki veri akışı Şekil 3.5’te verilmiştir. Eğer yeni bir kitap ekleniyorsa formda bulunan bütün veri alanları doldurulur. Kitap barkodu oluşturma işlemleri ise kitap kodunu referans alarak C# için hazırlanmış *MessagingToolkit.QRCode.Codec* kütüphanesi kullanılarak oluşturulur. Tüm veriler, veri tabanı yapısına göre değişken tipleri ayarlandıktan sonra gerekli SQL komutları ile veri tabanda bulunan *KTYP_BOOKS_DATA* adlı tabloya aktarılır.

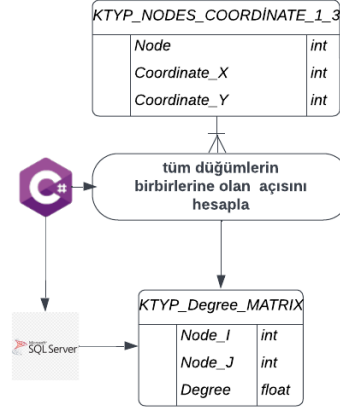
3.2.2 Raf ve masa (Ekle-Sil-Güncelle) işlemleri

Bu bölümde raf-masa veya yol gibi düğümlerin eklenmesi, silinmesi veya güncellenmesi esnasında oluşacak verilerin işlenmesi ele alınmıştır.

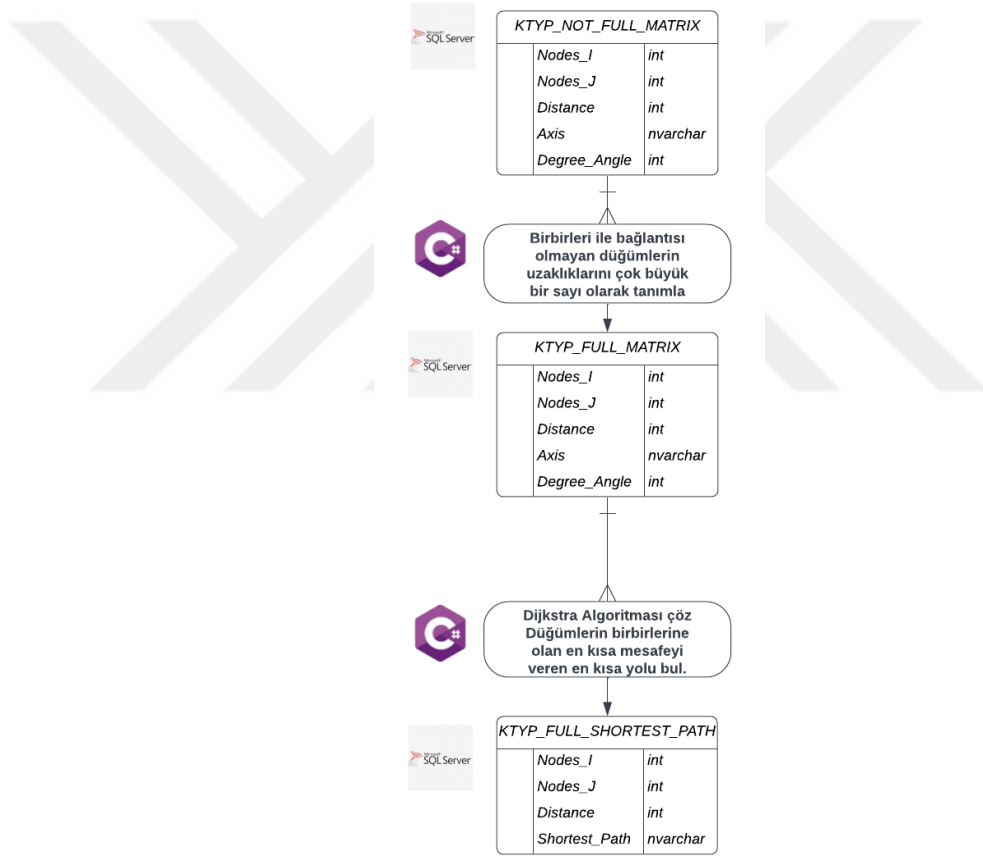


Şekil 3.6. Masa-Raf Bilgileri Veri Akışı

Şekil 3.6’da masa ve rafların ekle, sil, güncelle, filtrele gibi veri tabanı işlemlerinin, MVS ortamında tasarlanan form sayesinde veriler kullanıcıdan alınıp veri tabanında bulunan Nodes_Information, KTYP_NODE_AREA ve KTYP_NODES_COORDINATE_1_3 tablolarına aktarılmasındaki veri akışı verilmiştir. KTYP_NODES_COORDINATE_1_3 tablosu yerleşim planına göre olan koordinatların $\frac{1}{3}$ oranında küçültülmüş halini tutmaktadır.



Şekil 3.7. Tüm düğümlerin birbirlerine olan açısı hesaplanmasındaki veri akışı



Şekil 3.8. Tüm düğümlerin birbirlerine olan en kısa yol ve en kısa yolun uzaklığının hesaplanmasındaki veri akışı

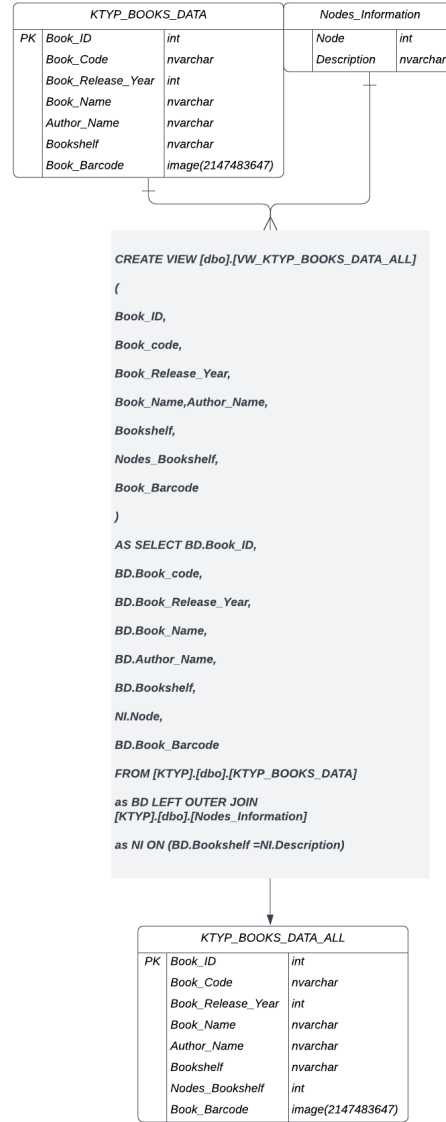
Eğer yeni düğüm eklenir veya mevcut düğümlerden biri güncellenirse tüm düğümlerin birbirlerine olan açısı, en kısa mesafe santimetre (cm) cinsinden ve en kısa yol gösterimi yeniden hesaplanarak matrisler güncellenir. Açılar Denklem (2.1)'deki gibi hesaplanır. Ortaya çıkan koordinat matrisi ise veri tabanında bulunan

KTYP_Degree_Matrix adlı tabloya eklenir. Bu durumda oluşan veri akışı Şekil 3.7’de verilmiştir.

Şekil 3.8’de düğümlerin birbirlerine olan en kısa mesafe ve yol gösterimi ile elde edilen matrisin veri akışı gösterilmiştir. Şekil 3.6’da form üzerinden kullanıcılardan veri alınır. Alınan veriler içerisinde üzerinde işlem yapılan düğümün doğrudan bağlantısı olan düğümler ve onlara olan mesafesi de istenmektedir. Bu veriler düğüm yay çiftleri bilgileri için veri tabanında bulunan *KTYP_NOT_FULL_MATRIX* adlı tabloya aktarılır. Bu tabloda ise düğümlerin doğrudan bağlantısı olan diğer düğümlere olan mesafesi ve kartezyen sisteme göre (x,y,-x,-y) bağlantı yönü, ve o yöndeki açı bilgileri tutulur. Daha sonrasında *KTYP_NOT_FULL_MATRIX* tablosunda birbirlerine doğrudan bağlantısı olmayan düğüm çiftleri uzaklıkları büyük bir sayı olarak belirlenir ve bu şekilde tüm düğümlerin birbirlerine doğrudan uzaklıkları *KTYP_FULL_MATRIX* adlı tabloya aktarılır. Ardından *KTYP_FULL_MATRIX* tablosundaki veriler çekilerek, Dijkstra Algoritması yardımı ile tüm düğüm çiftlerin birbirlerine en yakın mesafesi ve bu mesafe katedilirken izleyecekleri düğüm sıralamasını gösteren en kısa yol bilgileri *KTYP_FULL_SHORTEST_PATH* isimli tabloya eklenir.

3.2.3 Veri tabanında kullanılan gerekli sanal tablolar

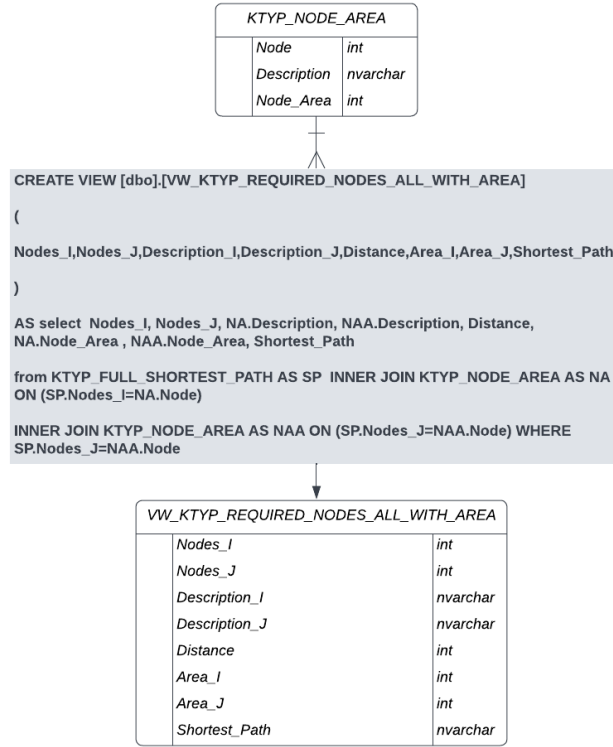
Veri tabanında bir veya daha fazla tablolardan verileri çekerek bir “*view*” yani sanal tablo oluşturulur. Bu sayede birden fazla tabloda bulunan bir ögenin çeşitli özellikleri tek bir tabloya indirgenmiş olur. Bir sanal tablo oluşturulduktan sonra verilerini kullandığı diğer tablolara yeni veriler eklense de sanal tablo yeni verileri de istenilen şekilde kullanabilir. Yani her yeni veri eklendiğinde sanal tablo üzerinde manuel olarak güncelleme yapılmaz. Tamamen diğer tabloların verilerini istenilen formatta birleştirerek getirir.



Şekil 3.9. *KTYP_BOOK_DATA_ALL* adlı Sanal Tablonun Oluşturulması ile ilgili Veri Akışı

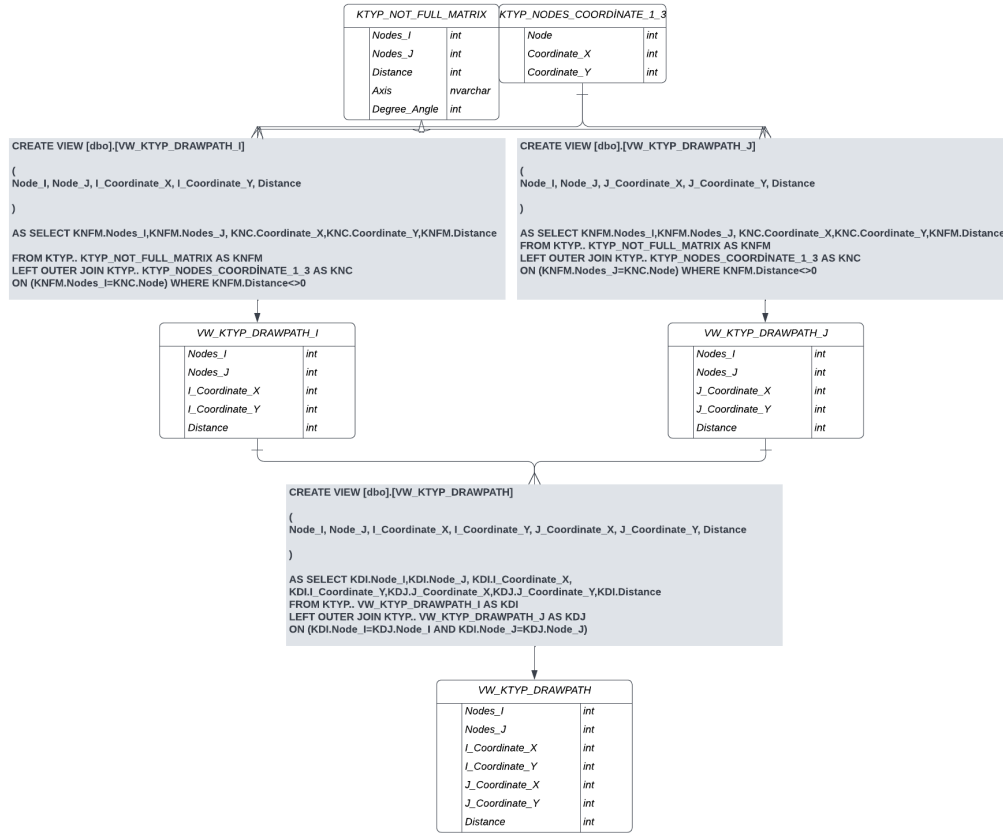
Şekil 3.9’da *KTYP_BOOK_DATA_ALL* adlı sanal tablo oluşturması ile ilgili akış şeması verilmiştir. Bu şemaya göre *KTYP_BOOK_DATA_ALL* adlı sanal tablo *KTYP_BOOK_DATA* tablosundan kitapların, *Book_ID*, *Book_code*, *Book_Release_Year*, *Book_Name*, *Author_Name*, *Bookshelf*, *Book_Barcode* verilerini çeker ve bunlardan *Bookshelf* alanı *Nodes_Information* adlı tablodaki *Description* alanı ile eşleşenleri *Nodes_Bookshelf* olarak tanımlanan alana *Nodes_Information* tablosundaki *Node* verisini ekler. Yani kitapların buldukları raf bilgileri *KTYP_BOOK_DATA* adlı tabloda varken bu rafların düğüm numaraları bu tabloda yoktur. *Nodes_Information* adlı tablodan bu

rafların düğüm numaraları çekilir ve *KTYP_BOOK_DATA_ALL* adlı sanal tablo olarak oluşturulur.



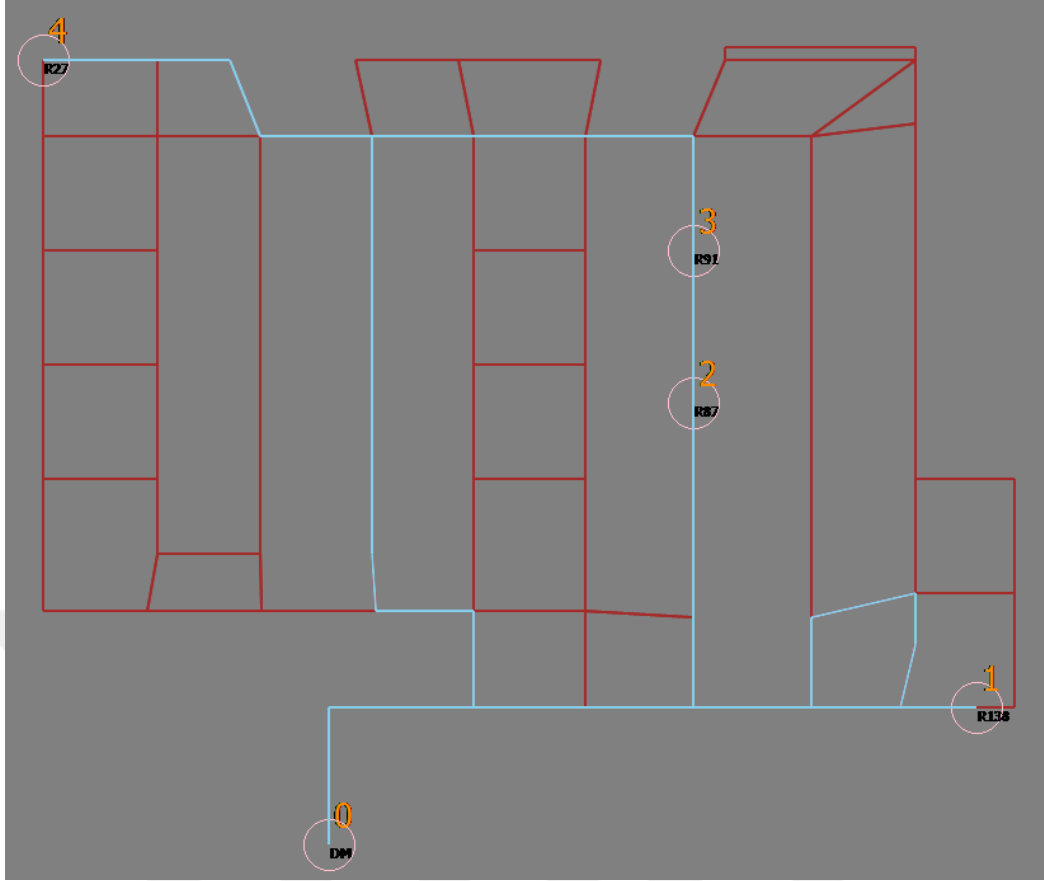
Şekil 3.10. *VW_KTYP_REQUIRED_NODES_ALL_WITH_AREA* adlı Sanal Tablonun Oluşturulması ile ilgili Veri Akışı

Şekil 3.10'da *VW_KTYP_REQUIRED_NODES_ALL_WITH_AREA* adlı sanal tablonun veri akış şeması verilmiştir. Bu sanal tablo *KTYP_NODE_AREA* tablosundan türetilmiş bir matris tablosudur. Bu matriste düğümlerin birbirlerine olan en kısa mesafesi, düğüm numaraları, düğüm açıklamaları, düğümlerin hangi bölgede bulunduğu ve bir düğümden diğerine giderken kullanacağı en kısa yol bilgisi yer almaktadır.

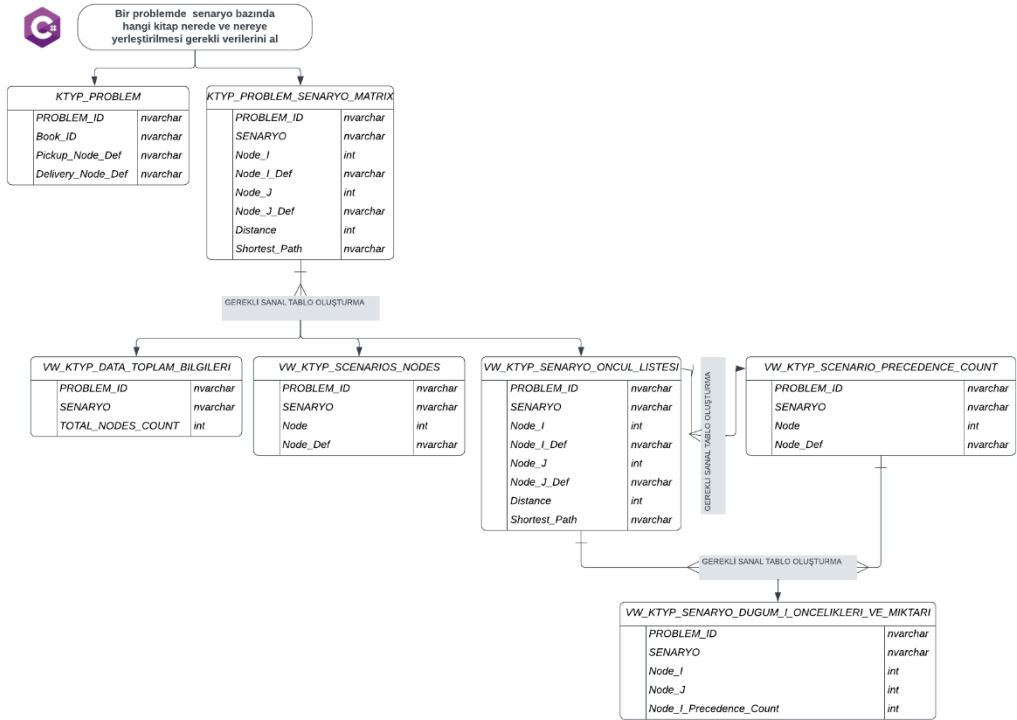


Şekil 3.11. `VW_KTYP_DRAWPATH` Adlı Sanal Tablonun Veri Akışı

Şekil 3.11’de verilen veri akış şemasına göre oluşan `VW_KTYP_DRAWPATH` adlı sanal tablo KDS tarafından çözülecek optimizasyon problemlerinde `KTUNK` içerisinde bulunan yolların çizimi için kullanılır. Bu yollar sadece birbirlerine doğrudan bağlantısı olan düğüm çiftleri üzerinden gösterilir. Birbirlerine doğrudan bağlantısı olan düğüm çiftlerinin verisi `KTYP_NOT_FULL_MATRIX` adlı tablodan, düğümlerin koordinat bilgileri de `KTYP_NODES_COORDINATE_1_3` adlı tablodan alınır. Bu sanal tablo ile kütüphane içerisindeki yolların çizilmesinde kullanılması gereken veri seti elde edilmiş olur. C# ortamında `System.Drawing.Drawing2D` kütüphanesi yardımı ile bu veri seti alınarak kütüphane içerisindeki yolların şebekesi çizilir. Ayrıca çözülen optimizasyon problemlerinde elde edilen nihai rotanın görselleştirilmesi ise bu sayede gerçekleşir. Aşağıda verilen Şekil 3.12’de `KTYP.B4.N6.T2.R4.1` adlı problemin Senaryo-1 altındaki EYKS ile çözümününün `VW_KTYP_DRAWPATH` adlı sanal tablodaki veriler kullanılarak görselleştirilmesi verilmiştir.



Şekil 3.12. *KTYP.B4.N6.T2.R4.1* adlı problemin Senaryo-1'ine ait EYKS ile elde edilen çözümün KDS tarafından görselleştirilmesi



Şekil 3.13. Bir problemin optimizasyonu için gerekli veri setlerini sağlayan sanal tabloların elde edilmesini gösteren veri akışı

KTYP kapsamında ortaya çıkan kitapların toplanması ve yerleştirilmesi ile ilgili optimizasyon problemlerinin verileri, kullanıcıdan alınır ve veri tabanına aktarılır. Şekil 3.13’de bu veri akışını ve sonucunda oluşturulan sanal tabloları göstermektedir. Bu sanal tablolar, geliştirilen çözüm yöntemleri için gerekli verileri ihtiyaç duyulan şekilde tutmaktadır. Buradaki amaç ise optimizasyon işlemleri esnasında ihtiyaç duyulan verileri daha hızlı elde etmektir. Problemin senaryo bazında tüm düğümlerin birbirlerine olan en kısa mesafesinin olduğu matris verileri *KTYP_PROBLEM_SENARYO_MATRIX* adında ana tabloda tutulur. Bu ana tablodan bir problemin senaryolarına ait toplam düğüm sayılarının bilgilerini tutacak *VW_KTYP_DATA_TOPLAM_BILGILERI* adlı sanal tablo oluşturulur. Bir problemin senaryolarında yer alan düğümlerin neler olduğu bilgisi *VW_KTYP_SCENARIOS_NODES* adlı sanal tabloda tutulur. Bir problemin senaryolarında yer alan düğümlerin öncülü olan düğümlerin bilgisi de *VW_KTYP_SENARYO_ONCUL_LISTESI* adlı sanal tablo oluşturularak elde edilir. Bu sanal tabloya bağlı olarak bir düğümün kaç adet öncül düğüme sahip olduğu bilgisi de *VW_KTYP_SCENARIO_PRECEDENCE_COUNT* adlı sanal tablo sayesinde elde edilir.

3.2.4 KDS’deki algoritmalara gerekli olan saklı yordam

VW_KTYP_SENARYO_ONCUL_LISTESI ve ondan türetilmiş *VW_KTYP_SCENARIO_PRECEDENCE_COUNT* sanal tablolar sayesinde sade ve amaca yönelik *VW_KTYP_SENARYO_DUGUM_I_ONCELIKLERI_VE_MIKTARI* adlı sanal tablo elde edilir. Elde edilen sanal tablolardaki veri setleri sayesinde SSMS’de SQL dili ile kodlanmış *SP_KTYP_SENARYO_ONCELIKPROSEDURU* Saklı Yordam (Stored Procedure) oluşturulmuştur. Herhangi bir senaryo altında KTYP için optimizasyon problemi çözümlenirken tezin ilerleyen bölümlerinde bahsedilen atanabilir düğümlerin olduğu kümelerin güncelleme işlemleri bu prosedür ile gerçekleşir. Bu prosedür geliştirilen algoritmaların atanabilir güncelleme şartına göre çalıştırılır. C# dilinde geliştirilen algoritma, atanabilir güncelleme şartına göre o an ki rotaya dahil edilmiş düğümleri (atanan düğümler kümesi), rotaya dahil edilebilir düğümleri (atanabilir düğümler kümesi), hangi problemi hangi senaryo altında çözdüğünü SQL dilinde “Exec” komutu ile veri tabanına gönderir. Bu komut ile prosedür , atanabilir ve atanabilir düğümler kümesinde olmayan ama tüm öncülleri atanabilir düğümler kümesinde bulunan düğümleri listeler. Sonrasında C# da geliştirilen algoritma ise bu listedeki düğümleri atanabilir

düğüm kümesine ekleyerek güncelleme işlemini gerçekleştirir. Atanabilir düğümler kümesi güncelleme veri akışı Şekil 3.14’de gösterilmiştir. Ayrıca tezin ilerleyen aşamasında atanabilir düğümler kümesinin güncelleme daha detaylı anlatılmıştır.

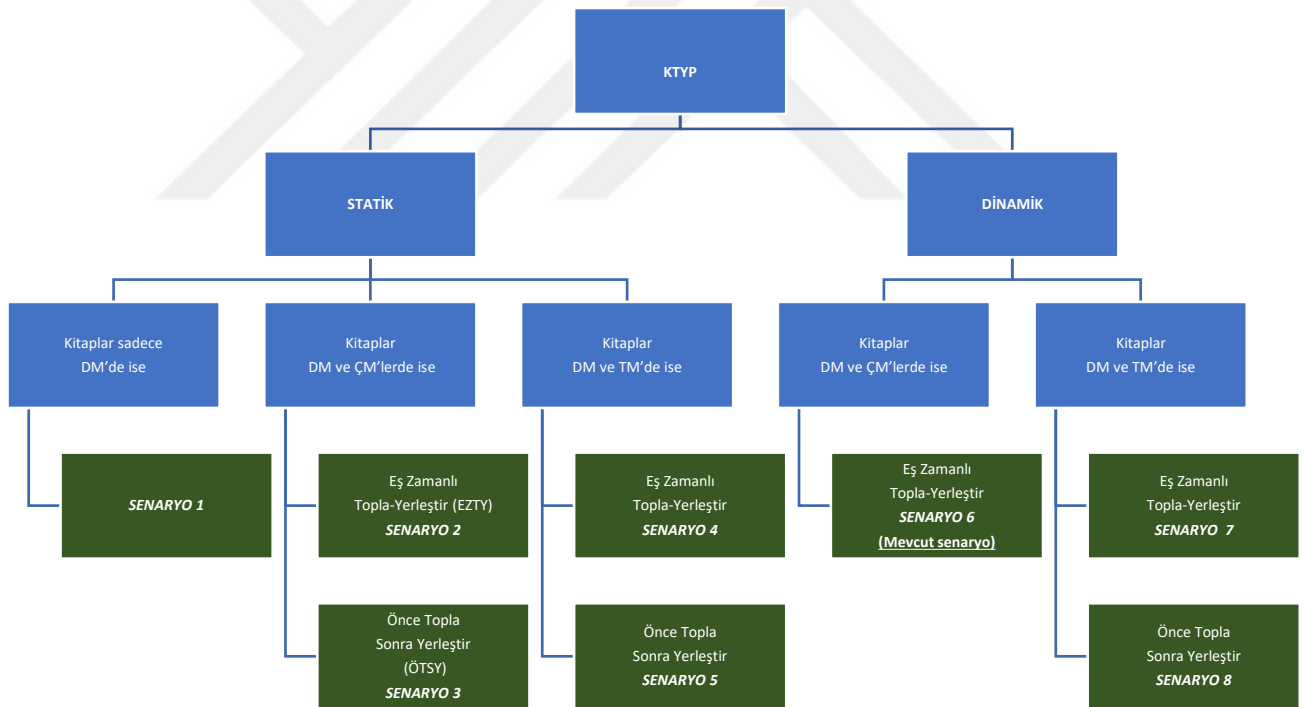


Şekil 3.14. Atanabilir Düğümler Kümesi (AB) Güncelleme prosedürü için veri akışı

3.3. KTYP Senaryoları

Şekil 3.15. KTYP Senaryolarının sınıflandırılması verilmiştir. Dinamik ve statik olarak problemler iki ana sınıfa ayrılmıştır. Bir senaryoda toplama ve yerleştirme rotası oluşturulmadan önce arz noktalarında bulunan kitaplar KDS tarafından biliniyor ise o senaryo statik, bilinmiyorsa dinamiktir. Kütüphanelerde bulunan farklı masalar (arz noktaları), farklı toplama ve yerleştirme prosedürleri ile farklı senaryolar ortaya çıkmaktadır. Kütüphanelerde bulunan arz noktalarındaki varsayımlar şu şekildedir;

- Kullanılan ve iade edilen tüm kitaplar danışma masasında olursa.
- Kullanılan kitaplar çalışma masalarında, iade edilen kitaplar danışma masasında olursa.
- Kullanılan kitaplar kütüphane içerisinde belirlenen bir toplama masasındadır, iade edilen kitaplar danışma masasında olursa.



Şekil 3.15. KTYP Senaryolarının sınıflandırılması

3.3.1 Senaryo-1

Senaryo-1 için varsayımlar şunlardır;

- Kütüphane içerisinde okumak veya çalışmak amacıyla kullanıcılar tarafından alınan ve kitaplar, sonrasında kullanıcılar tarafından danışma masasına bırakılır. İade koşulu ile alınan kitapların da iadesi danışma masasında gerçekleşir. Dolayısıyla iade edilen kitaplar ve kullanılan kitapların tümü danışma masasındadır.

- Danışma masasında, kitapları yerleştirmek için yeterli kapasitede bir tane kitap taşıma aracı mevcuttur.

- Danışma masasında bulunan bilgisayar ve barkod okuyucu sayesinde yerleştirilecek olan kitapların rafları bilinmektedir. Bu sayede problem sınıfı statiktir.

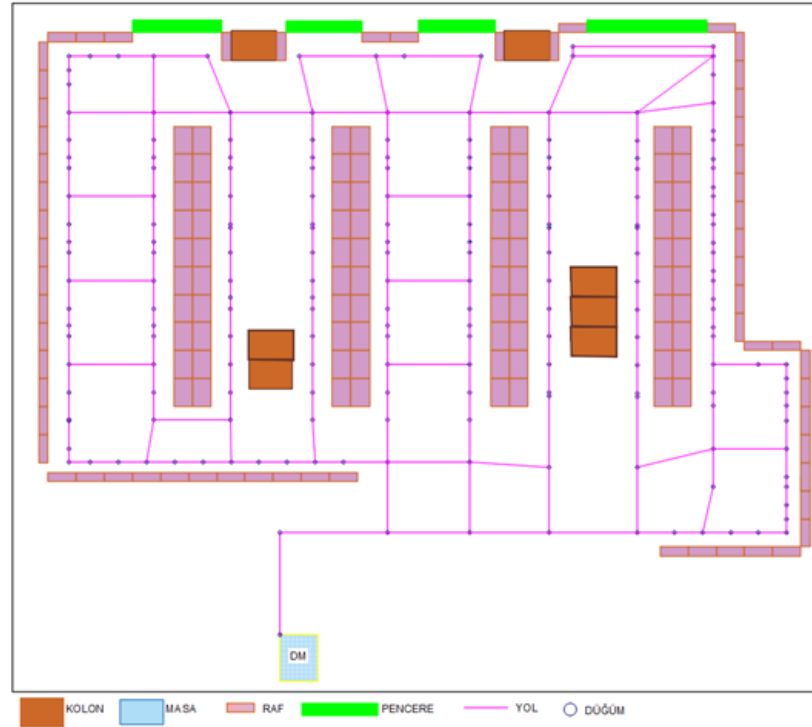
- Danışma masasında başlayan ve biten, ilgili rafları ziyaret edilerek kitapların yerleştirilmesi için en kısa rota bulunmalıdır.

- Her düğüm bir kez ziyaret edilir.

- Yerleştirme işlemine görevli uygun gördüğü zaman çıkmaktadır.

Bu varsayımlar altında Senaryo-1 literatürde çok iyi bilinen klasik Gezgin Satıcı Problemi (GSP) olarak ele alınabilir ama diğer senaryolar ve çözüm yöntemleri ile karşılaştırılabilmesi için bitiş düğümünü danışma masasının kopyası olan bir düğümde bitirilmesi varsayımıyla problem Ardışık Sıralama Problemi (ASP)'ye çevrilmiştir.

Şekil 3.16'da Senaryo-1 için toplama ve yerleştirme yapılacak düğümler yerleşim planı üzerinde gösterilmiştir. Kitapların tamamı danışma masasında olduğu için diğer masalarda kitap toplama işlemi gerçekleşmez. Problem daha anlaşılabilir olması açısından Şekil 3.16'da diğer masalar gösterilmemiştir.



Şekil 3.16. Senaryo-1 arz ve talep düğümleri (KTUNK)

3.3.2 Senaryo-2

Senaryo-2 için varsayımlar şunlardır;

- Kullanıcılar oturdukları çalışma masalarına kullanılmış kitapları bırakmaktadır. İade koşulu ile alınan kitapların da iadesi danışma masasında gerçekleşir. Dolayısıyla iade edilen kitaplar danışma masasında, kullanılmış kitaplar ise çalışma masalarındadır.

- Çalışma masalarında ve danışma masasında birden fazla kitap bulunabilir.
- Danışma masasında, kitapları yerleştirmek için yeterli kapasitede bir tane kitap taşıma aracı mevcuttur.

- Danışma masasında bulunan bilgisayar ve barkod okuyucu sayesinde iade edilen kitapların rafları bilinmektedir. Çalışma masalarında da barkod okuyucular olduğu varsayılarak hangi masa(lar)da hangi kitap(lar) var ve kullanılmış kitapların da ait olduğu raflar bilinir hale gelmektedir. Bu durumda senaryo statik bir yapıya çevrilmiştir.

- Danışma masasında başlayacak ve bitecek, ilgili rafları ve kitap olan çalışma masalarını öncelik ilişkilerini dikkate alarak ziyaret edecek şekilde kitapların toplanması ve yerleştirilmesi için en kısa rota bulunmalıdır.

- Toplama ve yerleştirme işlemleri eş zamanlı olarak gerçekleştirilir.

- Bir kitap bulunduğu masadan alınmadan ait olduğu rafa yerleştirilemez.
- Her düğüme birer kez ziyaret gerçekleştirilir.
- Yerleştirme işlemine görevli uygun gördüğü zaman çıkmaktadır.

Bu varsayımlar altında Senaryo-2 ÖKGSP olarak ele alınabilir ama diğer senaryolar ve çözüm yöntemleri ile karşılaştırılabilmesi için bitiş düğümünü danışma masasının kopyası olan bir düğüme bitirilmesi varsayımıyla problem Ardışık Sıralama Problemi (ASP)'ye çevrilmiştir.

3.3.3 Senaryo-3

Senaryo-3 , Senaryo-2'den tek farkı toplama ve yerleştirme kuralıdır. Senaryo-2'de toplama ve yerleştirme işlemleri eş zamanlı yapılırken, Senaryo-3'te ise önce kitapların tamamı ilgili arz noktalarından toplanır daha sonra yerleştirilir. Bu varsayım ele alındığında Senaryo-3, danışma masasından başlayarak önce ilgili masa(lar)ı ziyaret ederek kitapları toplar daha sonra toplanan kitapların ait oldukları raflar ziyaret edilerek toplanan kitaplar yerleştirilir tekrar danışma masasına geri dönlür. Özet olarak arz düğümleri, talep düğümlerinden önce ziyaret edilir. Bitiş düğümü ise diğer tüm senaryolarda olduğu gibi danışma masasının kopyası olan bir düğüm olduğu için problem ASP olarak ele alınabilir.

3.3.4 Senaryo-4

Senaryo-4 ise Senaryo-2 ile aynı varsayımlar altında incelenir. Tek fark kullanılan kitaplar kullanıcılar tarafından toplama masası adı verilen arz noktasına bırakılır. Dolayısıyla hiçbir çalışma masasında kullanılmış kitap bulunmamaktadır. Bu varsayım altında çalışma masaları arz noktası olarak değerlendirilmez. Senaryo-2'deki gibi iade edilen kitapların danışma masasında bulunması varsayımı bu senaryo içinde geçerlidir. Ayrıca toplama masası ve danışma masasında gerekli barkod okuyucu ve KDS olduğu varsayıldığı için her iki masadaki kitaplar ve yerleştirilecekleri raflar önceden bilinir. Bu varsayım ile Senaryo-4 statiktir. Toplama masası, Şekil 3.17'deki gibi kütüphanenin içerisinde belirli bir noktada olan görevli masalarıdır. KTUNK için toplama masası olarak kütüphanenin tam ortasında bulunan bir masa seçilmiştir. Bu masa gerçekte toplama masası olarak kullanılmasa da tezde bu senaryo altında toplama masası

- Kullanıcılar oturdukları çalışma masalarına kullanılmış kitapları bırakmaktadır. İade koşulu ile alınan kitapların da iadesi danışma masasında gerçekleşir. Dolayısıyla iade edilen kitaplar danışma masasında, kullanılmış kitaplar ise çalışma masalarındadır.

- Danışma masasında, kitapları yerleştirmek için yeterli kapasitede bir tane kitap taşıma aracı mevcuttur.

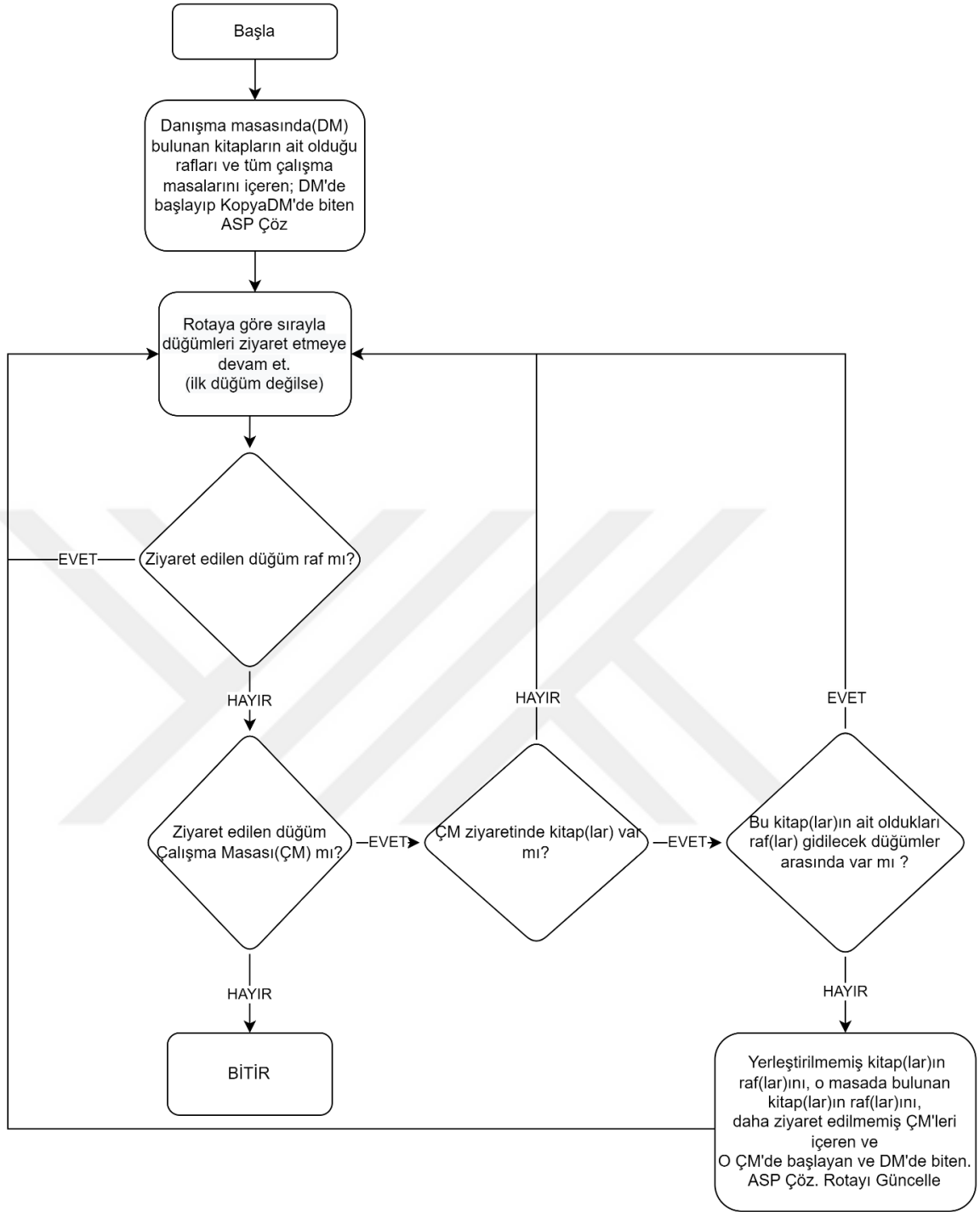
- Danışma masasında bulunan bilgisayar ve barkod okuyucu sayesinde iade edilen kitapların rafları bilinmektedir. Çalışma masalarında herhangi bir karar destek sistemine bağlı bir barkod okuyucu vb. bulunmamaktadır. Dolayısıyla, çalışma masalarında bırakılan kitaplar ise bilinmemektedir. Bu durumda hangi masa(lar)da hangi kitap(lar) var bilinmez. Bu durumda görevli danışma masasından iade edilen kitapları yerleştirirken aynı zamanda da çalışma masalarını gezmektedir.

- Danışma masasında başlayan ve biten, ilgili rafları ve tüm çalışma masalarını ziyaret ederek kitapların toplanması ve yerleştirilmesi için en kısa rota bulunmalıdır.

- Rota ziyaret sırası ve işlemleri görevlide bulunan el terminaline aktarıldığı varsayılmaktadır.

- Toplama ve yerleştirme işlemleri eş zamanlı olarak gerçekleştirilir.
- Bir kitap bulunduğu masadan alınmadan ait olduğu rafa yerleştirilemez.
- Yerleştirme işlemine görevli uygun gördüğü zaman çıkmaktadır.

Senaryo-6 için akış şeması Şekil 3.18'de gösterilmiştir. Örneğin, bir görevli danışma masasından itibaren başladığı toplama ve yerleştirme ziyaretlerine devam ederken, gittiği her düğümü elindeki barkod okuyucu ile "gidildi" olarak işaretlemektedir. Bu durumda görevli bir çalışma masasını ziyaret ettiğini ve orada bir kitapla karşılaştığını varsayalım. El terminali sayesinde kitabın barkodu okutulur. Böylece kitabın hangi rafa ait olduğu, daha önce o rafın ziyaret edilip edilmediği karar destek sistemi tarafından bilinir hale getirilir. Bu bilgiler doğrultusunda, eğer kitabın rafı, o çalışma masasına gelmeden önce ziyaret edilmişse veya mevcut ziyaret edilmesi gereken düğümler arasında değilse, rota güncellenir. Yeni rota, o çalışma masasında başlayan ve danışma masasında biten, ziyaret edilmemiş çalışma masalarını ve ziyaret edilmesi gereken rafları içeren bir ASP çözümü olarak oluşturulur. Ayrıca Şekil 3.19'da Senaryo-6'ya ait *pseudo* kodlaması verilmiştir.



Şekil 3.18. Senaryo-6 Akış Şeması

Adım 0: BAŞLA

ASP için seçilen çözüm yöntemine göre toplama ve yerleştirme rotası oluştur;

Adım 1:

$N = \text{Adım 0'da elde edilen ziyaret edilecek düğümleri temsil etsin } (0,1,2,\dots,i,\dots,n)$

$i=0$

Adım 2:

Eğer ($i = 0$ veya i . Düğüm talep düğümü ise)

Eğer ($i = n$) ise

Adım 3'e git;

Değilse

i . Düğümü ziyaret et ve kitapları yerleştir;

$i=i+1$;

Adım 2'e git;

Değilse

Düğümde bulunan kitapları topla ve barkodlarını okut;

Eğer (bu kitapların ait oldukları raflar ziyaret edilecekler arasında) ise

Bu kitapların ait oldukları rafları ziyaret edilecek düğümler arasına ekle;

Ziyaret edilecek düğümleri güncelle;

Adım 1'a git;

Değilse

$i=i+1$;

Adım 2'e git;

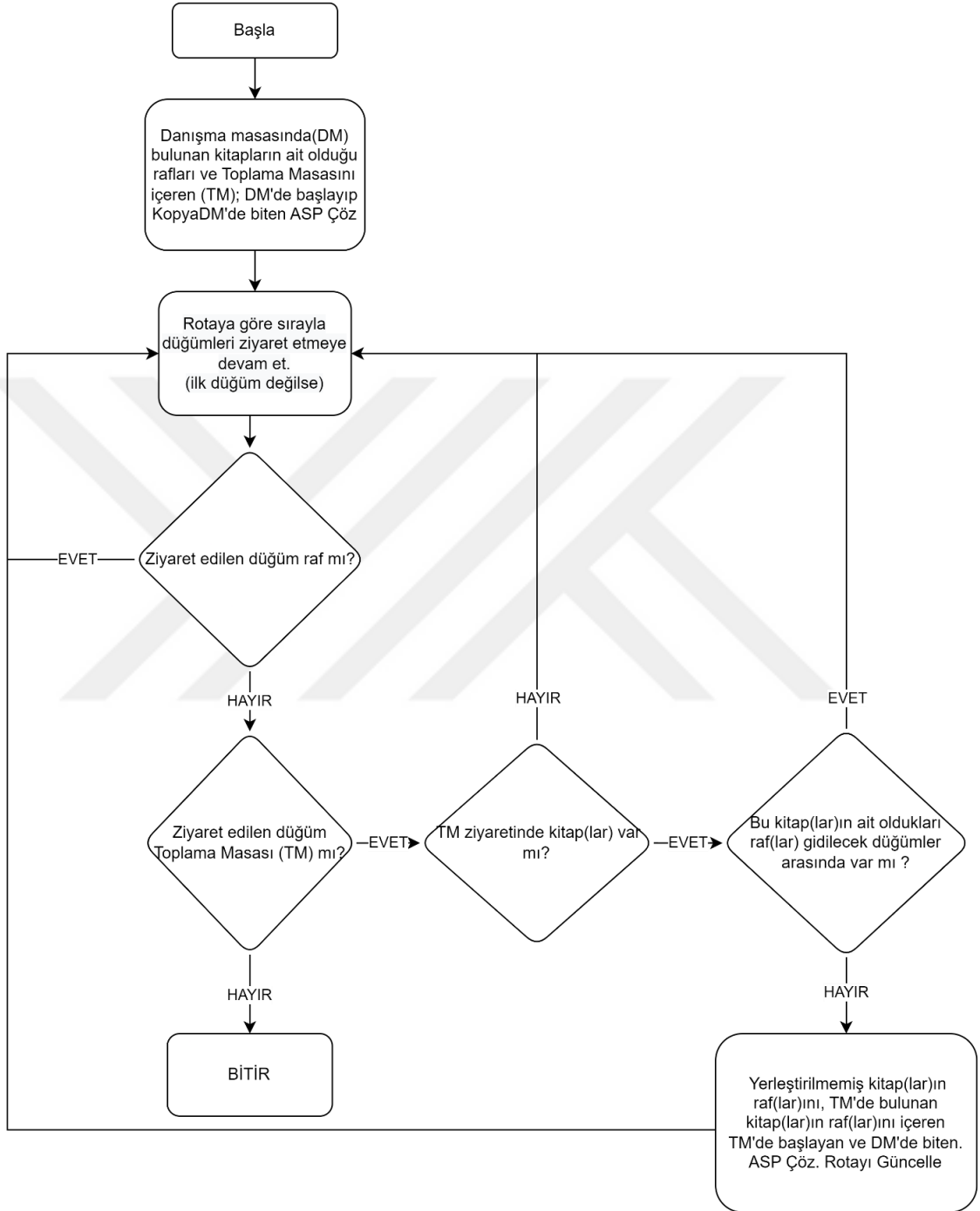
Adım 3: Durdur

Şekil 3.19. Senaryo-6-7 Pseudo Kod

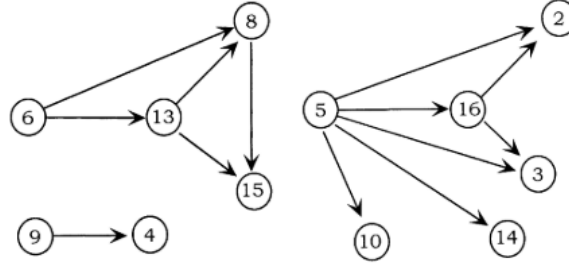
3.3.7 Senaryo-7

Senaryo-7'nin, Senaryo-6'dan tek farklı çalışma masalarının birer arz düğümü olarak değerlendirilmediği, bunun yerine Şekil 3.17'deki gibi KTUNK ortasında bir toplama masası olduğu varsayılır. Yani iade edilen kitaplar danışma masasında iken kullanılmış kitaplar toplama masasındadır. Fakat, toplama masasında bulunan kitaplar hakkında görevliler ve karar destek sistemi herhangi bir bilgiye sahip değildir. Buna bağlı olarak bu senaryo da Senaryo-6 gibi dinamik olarak sınıflandırılır. Dolayısıyla, ilk rota danışma masasında başlayan ve danışma masasında bulunan kitapların raflarını ve toplama masasını içeren nihayetinde danışma masasının kopyasında biten bir ASP çözerek oluşturulur. Görevli ziyaret esnasında toplama masasında kitapları aldıktan sonra bu kitapların ait oldukları raflar mevcut rotası üzerinde henüz ziyaret edilmemiş raflar arasında yoksa; toplama masasında başlayan danışma masasında biten ve yerleştirilmemiş kitapların ait olduğu rafları ziyaret edecek şekilde bir

ASP rotası oluşturulur. Ayrıca Şekil 3.19'da Senaryo-7'ye ait *pseudo* kodlaması verilmiştir.



Şekil 3.20. Senaryo-7 Akış Şeması



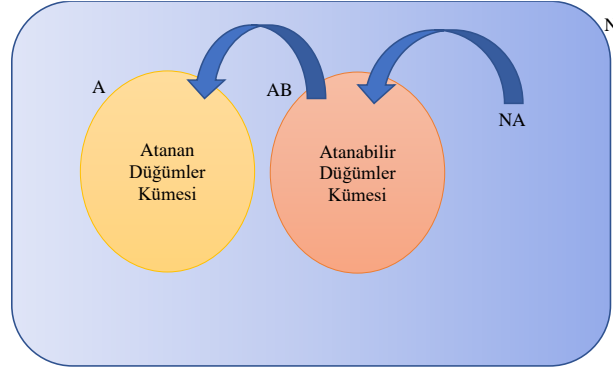
Şekil 3.21. TSPLib'de yer alan br17 adlı ASP örneğinin Öncelik Kısıtlamaları yapısı

Süpürme Algoritması (SÜP_A) yapısı itibariyle düğümlerin koordinat bilgilerine de ihtiyaç duyar. Bölgesel En Yakın Komşu Sezgiseli BEYKS ise düğümlerin kütüphane içerisinde bulunan koridor veya bölge bilgisine ihtiyaç duymaktadır. KTYP için uyarlanan ASP optimizasyon yöntemlerinden Süpürme Algoritması (SÜP_A) ve Bölgesel En Yakın Komşu Sezgiseli (BEYKS) koordinat bilgisi veya bölge bilgisi gibi parametrelere sahip olduğu için TSPLib’de bulunan ASP örneklerine uygulanamaz

Çözüm yöntemlerinde kullanılan 4 temel küme aşağıda açıklanmıştır.

1. N = Tüm düğümleri içeren küme $\{0,1,2,\dots,i,\dots,n\}$
2. A = Ziyaret sırasına göre ataması gerçekleşen düğümlerin kümesi .
3. AB = Atanabilir düğümler kümesi.
4. $NA = N - (A \cup AB)$: N kümesinde olup hem A 'da hem de AB 'de olmayan düğümlerin kümesi.

Bir düğümün atanabilmesi yani A kümesinde eklenebilmesi için önce AB kümesinde olması gerekmektedir. Nihayetinde tüm düğümler A kümesine eklendiğinde A kümesi ziyaret sırasına göre elemanları sıraladığı için bir ASP çözümü ifade eder. Bu durum Şekil 3.22’de tasvir edilmiştir. Bir düğümün atanabilir düğümler kümesinde (AB) olabilmesi için kendisi NA kümesinde, tüm öncüllerinin ise A kümesinde olması gerekmektedir.



Şekil 3.22. Kümeler arası düğüm transferi

Adım 0: BAŞLA

$i : [0, 1, \dots, a)$

$NA_i = a$ sayıda düğüm içeren kümenin i .ninci elemanı ($a \geq 0$);

$i = 0$;

Eğer ($a = 0$) ise

Tüm düğümler atanmış;

Adım 3'e git

Değilse (Devam et)

$A = b$ sayıda düğüm içersin ($b \geq 0$);

Eğer ($b = 0$) ise

Başlangıç düğümünü AB 'ye ekle;

Adım 3'e git;

Değilse(Devam et)

$AB = c$ sayıda düğüm içersin ($c \geq 0$);

Adım 2:

$NA_i = x$ numaralı düğüm olsun;

Eğer (x 'in tüm öncülleri $\in A$) ise

x 'i AB 'ye ekle;

Eğer ($i = a-1$) ise

Adım 3'e git;

Değilse (Devam Et)

Değilse (Devam et)

$i = i + 1$;

Adım 2'ye git;

Adım 3: BİTİR;

Şekil 3.23. Atanabilir Düğümler Kümesi (AB) Güncelleme Pseudo Kod

AB , geliştirilen algoritmalarda ihtiyaca göre Şekil 3.23'deki gibi güncellenir. Bu tezde kullanılan algoritmalarda iki durumda AB kümesi güncellenir. Birinci durum, her A kümesine bir düğüm eklendiğinde güncellenir. İkinci durum ise AB kümesinde eleman kalmayınca güncellenir. Yani algoritma tarafından AB kümesinden düğüm seçilir. Seçilen düğüm A kümesine atanır. Bu işlem devam ederken AB kümesi güncellenmez ta ki AB kümesinde eleman kalmayana kadar. Bu nedenle tezdeki algoritmaların iki versiyonu vardır. Birinci versiyonu $v1$, ikinci versiyonu ise $v2$ ile ifade edilmektedir.

3.4.1 Rastsal Çözüm Algoritması

Rastsal çözüm yöntemi ASP'leri rastsal olarak çözer. Ancak bu çözüm yöntemi akıllıca bir optimizasyon tekniği değildir. Bu yöntem KTYP için geliştirilmiş meta sezgisel olan PSO için uygun başlangıç çözümlerini üretir.

0 'da başlayıp n 'de biten düğümlerin rastsal bir şekilde ziyaret edilmesini sağlayan ASP algoritmanın pseudo kodlaması Şekil 3.24'te gösterilmiştir. Algoritma sonlandığında A kümesi çözümü ifade eder. AB güncellemesine göre rastsal çözümün ikinci versiyonu Şekil 3.25'deki gibidir.

Adım 0: BAŞLA

$i = 0$;

$N = \{0,1,2,\dots,j,\dots,n\}$

$A = \emptyset$;

$AB = \emptyset$;

$NA = N$;

Adım 1:

Eğer ($i \neq n$) ise

AB kümesini güncelle;

$x = AB$ içerisinden rastgele seçilen düğüm

A kümesine x 'i ekle $= A + \{x\}$;

AB kümesinden x 'i sil $= AB - \{x\}$;

$i = i + 1$;

Adım 1'e git;

Değilse

Adım 2'ye git;

Adım 2: BİTİR

Şekil 3.24. Rastsal ASP çözümü $v1$ için Pseudo Kod

Adım 0: BAŞLA

$i = 0 ;$

$N = \{0, 1, 2, \dots, j, \dots, n\}$

$A = \emptyset;$

$AB = \emptyset;$

$NA = N;$

Adım 1:

Eğer ($i \neq n$) ise

Eğer (AB eleman sayısı $= 0$) ise

AB kümesini güncelle;

Değilse (*Devam et*)

$x = AB$ içerisinde rastgele seçilen düğüm

A kümesine x 'i ekle $= A + \{x\};$

AB kümesinden x 'i sil $= AB - \{x\};$

$i = i + 1 ;$

Adım 1'e git;

Değilse

Adım 2'ye git;

Adım 2: BİTİR

Şekil 3.25. Rastsal ASP çözümü v2 için Pseudo Kod

3.4.2 En Yakın Komşu Sezgiseli (EYKS)

Literatürde birçok alanda sıkça kullanılan EYKS ASP'ye uyarlanmıştır. Bu yöntemde AB kümesinde olan düğümler arasında A kümesine son eklenen düğüme en yakın düğüm atanır. EYKS'nin pseudo kodlaması Şekil 3.26'de gösterilmiştir. EYKS'nin ikinci versiyonu da AB kümesinde eleman kalmayınca güncellendiği versiyondur. EYKS versiyon iki pseudo kodu Şekil 3.27'de verilmiştir.

Adım 0: BAŞLA

$i = 0 ;$

$N = \{0, 1, 2, \dots, j, \dots, n\}$

$A = \emptyset;$

$AB = \emptyset;$

$NA = N;$

Adım 1:

Eğer ($i \neq n$) ise

AB kümesini güncelle;

Eğer ($A = \emptyset$) ise

$A + \{0\};$

$AB - \{0\};$

$i = i + 1;$

Adım 1'e git;

Değilse

$x = AB$ içerisinde A kümesinin son elemanına en yakın düğüm;

$A + \{x\};$

$AB - \{x\};$

$i = i + 1;$

Adım 1'e git;

Değilse

Adım 2'ye git;

Adım 2 : BİTİR

Şekil 3.26. ASP için EYKS v1 Pseudo Kod

Adım 0: BAŞLA

$i = 0 ;$

$N = \{0, 1, 2, \dots, j, \dots, n\}$

$A = \emptyset;$

$AB = \emptyset;$

$NA = N;$

Adım 1:

Eğer ($i \neq n$) ise

Eğer (AB eleman sayısı = 0) ise

AB kümesini güncelle;

Değilse (Devam et)

Eğer ($A = \emptyset$) ise

$A + \{0\};$

$AB - \{0\};$

$i = i + 1;$

Adım 1'e git;

Değilse

$x = AB$ içerisinde A kümesinin son elemanına en yakın düğüm;

$A + \{x\};$

$AB - \{x\};$

$i = i + 1;$

Adım 1'e git;

Değilse

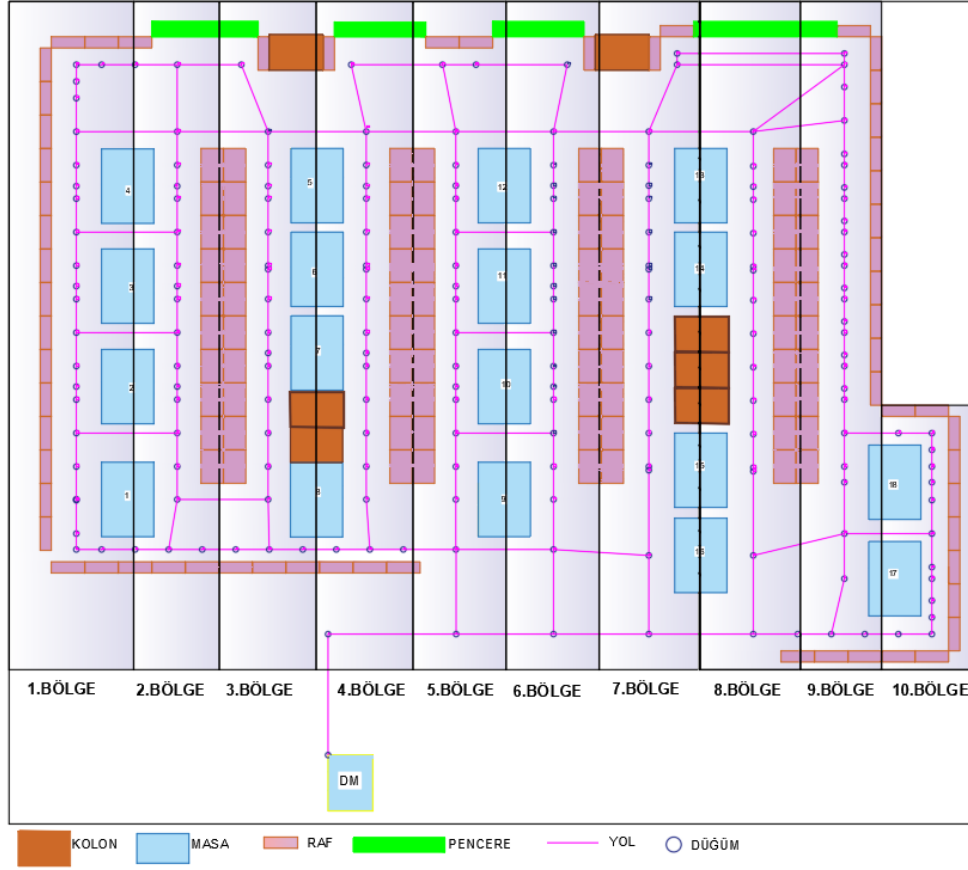
Adım 2'ye git;

Adım 2 : BİTİR

Şekil 3.27. ASP için EYKS v2 Pseudo Kod

3.4.3 Bölgesel En Yakın Komşu Sezgiseli (BEYKS)

Daha önce bahsedildiği üzere bu sezgisel EYKS'nin bir modifikasyonu olarak KTYP'ye özel olarak geliştirilmiştir. Bu sezgiselde kütüphane içerisindeki koridorlara göre Şekil 3.28'deki gibi bölgelere ayrılmıştır. Bu bölgeler soldan sağa doğru 1'den başlayarak numaralandırılır ya da sağdan sola doğru da numaralandırılabilir.



Şekil 3.28. Dügümlerin bölgelere ayrılması (KTUNK üzerinden Örnek)

BEYKS Algoritması, ziyaret edilebilir düğümler arasında bölge numarası en küçük olan düğümü ziyaret eder. Eğer bu bölgede birden fazla düğüm var ise ziyaret edilen son düğüme en yakın düğüm seçilir. BEYKS pseudo kodlaması Şekil 3.29'de gösterilmiştir. BEYKS ikinci versiyonu pseudo kodlaması da Şekil 3.30'de verilmiştir.

Adım 0: BAŞLA

$i = 0$;

$N = \{0, 1, 2, \dots, j, \dots, n\}$

$A = \emptyset$;

$AB = \emptyset$;

$NA = N$;

Adım 1:

Eğer ($i \neq n$) ise

AB kümesini güncelle;

Eğer ($A = \emptyset$) ise

$A + \{0\}$;

$AB - \{0\}$;

$i = i + 1$;

Adım 1'e git;

Değilse

$x = AB$ kümesi içerisinde bölge numarası en küçük değere sahip düğümler arasından A kümesinin son elemanına en yakın düğüm;

$A + \{x\}$;

$AB - \{x\}$;

$i = i + 1$;

Adım 1'e git;

Değilse

Adım 2'ye git;

Adım 2 : BİTİR

Şekil 3.29. BEYKS v1 Pseudo Kod

Adım 0: BAŞLA

$i = 0;$

$N = \{0, 1, 2, \dots, j, \dots, n\}$

$A = \emptyset;$

$AB = \emptyset;$

$NA = N;$

Adım 1:

Eğer ($i \neq n$) ise

Eğer (AB Kümesi eleman sayısı = 0) ise

AB kümesini güncelle;

Değilse (Devam et)

Eğer ($A = \emptyset$) ise

$A + \{0\};$

$AB - \{0\};$

$i = i + 1;$

Adım 1'e git;

Değilse

$x = AB$ içerisinde A kümesinin son elemanına en yakın düğüm;

$A + \{x\};$

$AB - \{x\};$

$i = i + 1;$

Adım 1'e git;

Değilse

Adım 2'ye git;

Adım 2 : BİTİR

Şekil 3.30. BEYKS v2 Pseudo Kod

3.4.4 Süpürme Algoritması (SÜP_A)

Süpürme Algoritması (SÜP_A), KTYP uyarlamasında önce bütün düğümlerin danışma masasına olan açısı denklem (2.1)'deki gibi hesaplanır. Referans noktası olarak danışma masası seçilmiştir. Ancak bu denklemde çıkan sonuçlar $-\pi$ ile π arasındadır. Eğer sonuç negatif ise 360° eklenerek pozitif açısı elde edilir. Elde edilen sonuçların veri seti veri tabanında tutulur. AB kümesinde olan düğümler içerisinde en küçük açiya sahip olan düğüm A kümesine eklenir. Bu işlem tüm düğümler A kümesine eklenene kadar devam eder. SÜP_A pseudo kodlaması Şekil 3.31'de verilmiştir. SÜP_A'da AB kümesi güncellemesi durumuna göre iki versiyona sahiptir.

Adım 0: BAŞLA

$i = 0 ;$

$N = \{0, 1, 2, \dots, j, \dots, n\}$

$A = \emptyset;$

$AB = \emptyset;$

$NA = N;$

Adım 1:

Eğer ($i \neq n$) ise

AB kümesini güncelle;

Eğer ($A = \emptyset$) ise

$A + \{0\};$

$AB - \{0\};$

$i = i + 1;$

Adım 1'e git;

Değilse

$x = AB$ içerisinde Danışma Masasına açısı en küçük olan düğüm;

$A + \{x\};$

$AB - \{x\};$

$i = i + 1;$

Adım 1'e git;

Değilse

Adım 2'ye git;

Adım 2 : BİTİR

Şekil 3.31. SÜP_A v1 Pseudo Kod

3.4.5 Süpürme Algoritması-2 (SÜP_A-2)

SÜP_A-2'de tüm düğümlerin birbirlerine olan açısı denklem (2.1)'deki gibi hesaplanır ve bu veri seti, veri tabanında tutulur. Eğer sonuç negatif bir açı ise 360° eklenerek pozitif açığa dönüştürülür. Bu algoritmanın SÜP_A'dan tek farkı referans noktasının son ziyaret edilen düğüm seçilmesidir. Yani tüm düğümler ziyaret edilene kadar AB kümesi içerisinde olan düğümler arasından A kümesine en son eklenen düğüme en küçük açığa sahip düğümün A kümesine eklenmesidir. SÜP_A-2'de diğer algoritmalar gibi AB kümesinin güncellenme durumuna göre iki versiyona sahiptir.

3.4.6 Parçacık Sürü Optimizasyonu (PSO)

PSO'nun yapısı, parametreleri ve değişkenleri bu tezde ele alınan diğer sezgisellerden daha karmaşıktır. Şöyle ki elde edilen uygun bir ASP çözümü bu algoritma da bir parçacığı temsil etmektedir. Bu çözümün düğüm sıralaması parçacığın pozisyonunu, amaç fonksiyonu ise parçacığın uygunluk değerini ifade eder. Bir parçacığın hızı ise o anki pozisyonda yapılacak yer değiştirme işlemleridir. Bu tez de ASP optimizasyonu için geliştirilen PSO, Rastsal PSO (R-PSO) olarak adlandırılmıştır. R-PSO algoritması için gereken parametreler, değişkenler ve veri setleri aşağıda verilmiştir;

- t = iterasyon sayısı
- p = parçacık sayısı
- N = Tüm düğümlerin bulunduğu set.
- $SF = \{ (a,b) \}$ a düğümünde başla b düğümünde bitir.
- $\ddot{O} = \{ (j,i) \}$ j düğümüne gitmeden önce i düğümüne gidilmesi gerektiğini gösteren öncelik ilişkileri seti.
- $C_{i,j} = i'den j'ye gitme maliyeti.$
- $P_{t,p} = t. iterasyonda p. parçacığın pozisyonu.$
- $PU_{t,p} = t. iterasyonda p. parçacığının uygunluk değeri.$
- $Pbest_p = p. parçacığın şimdiye kadarki en iyi pozisyonu.$
- $PUbest_p = p. parçacığın şimdiye kadarki en iyi uygunluk değeri.$
- $Gbest = parçacıklar arasında en iyi pozisyon.$
- $GUbest = parçacıklar arasında en iyi uygunluk değeri.$
- $v_{t+1,p} = bir sonraki iterasyonda p. parçacığının hızı.$
- $c_1 = Bilişsel öğrenme katsayısı.$
- $c_2 = Sosyal öğrenme katsayısı.$
- $SO_{Gbest} = PU_{t,p}$ ziyaret sırası $Gbest'e$ göre farklı olan düğümler.
- $SO_{Pbest} = PU_{t,p}$ ziyaret sırası $Gbest'e$ göre farklı olan düğümler.
- $Ghız miktarı = c_2 * SO_{Gbest}$ miktarı
- $Ghız = Ghız$ miktarı kadar SO_{Gbest} 'ten rastgele yer değiştirme işlemleri.
- $Phız miktarı = c_1 * SO_{Pbest}$ miktarı
- $Phız = Phız$ miktarı kadar SO_{Pbest} t'ten rastgele yer değiştirme işlemleri

- $v_{t+1,p} = G_{hız} + P_{hız} = p. \text{parçacığıın } t + 1. \text{ zamandaki hızı}$

R-PSO'da ilk iterasyonda parçacık sayısı kadar pozisyon Şekil 3.24 veya Şekil 3.25'deki gibi rastgele şekilde üretilir. R-PSO'da iki versiyona sahiptir. R-PSO algoritmasında p numaralı parçacığın t . iterasyondaki işlemleri ve bir sonraki iterasyonun hızını ve pozisyonunun nasıl hesaplandığı aşağıdaki verilen örneklerle anlatılmıştır.

R-PSO ÖRNEK :

Örnek Parametreler ve değişkenlerin belirlenmesi

$$N = \{0,1,2,3,4,5,6\}$$

$$SF = \{(0,6)\}$$

$$Ö = \{(3,4)\}$$

$$P_{best\ p} = \{0,5,4,1,2,3,6\}$$

$$PU_{best\ p} = 27$$

$$G_{best} = \{0,5,4,1,4,2,3,6\}$$

$$GU_{best} = 23$$

$C_{i,j}$ matrisi Tablo 5'de

$$P_{t,p} = \{0,2,5,4,3,1,6\}$$

$$PU_{t,p} = 28$$

$c2 = [0.5,1)$ arasında rastsal bir sayı

$$c1 = 1 - c2$$

$c1 = 0.5$ olsun

$c2 = 0.5$ olsun

Tablo 5. Örnek ASP maliyet matrisi

İ,J	0	1	2	3	4	5	6
0	0	5	4	8	7	3	10000
1	-1	0	7	2	5	4	6
2	-1	7	0	4	4	3	5
3	-1	2	4	0	-1	4	3
4	-1	5	4	8	0	5	2
5	-1	4	3	4	5	0	1
6	-1	-1	-1	-1	-1	-1	0

Yer deęiřtirme operatörlerinin uygulanması

p. parçacıęın t. iterasyondaki hızı ise $P_{t,p}$ 'de kaç adet düęümü yer deęiřtirilmesi gerekiyor ve bu düęümleri hangi sıraya yerleřtirilmesi ile bulunur. Yer deęiřtirilmesi gereken düęüm sayısı P_{best} ve G_{best} pozisyonlarında bulunan düęümlerin sıralaması ile doğrudan bağlantılıdır. Çünkü hız yer deęiřtirme operatörleri ile ifade edilir.

Örneęin $P_{t,p}$ 'de 1 numaralı düęüm altıncı sırada yer alırken P_{best} 'te bu düęüm dördüncü sırada, G_{best} 'te ise üçüncü sırada yer almaktadır. Bu durum $SO_{P_{best}} = \{ (1,4) \}$ ve $SO_{G_{best}} = \{ (1,3) \}$ olarak gösterilir.

$P_{t,p}$ 'de de bulunan düęümlerin toplamda 4 tanesi G_{best} 'e göre farklı konumdadır.

- $SO_{G_{best}} = \{ (2,5) , (5,2) , (3,6) , (1,3) \}$

$P_{t,p}$ 'de de bulunan düęümlerin toplamda 5 tanesi P_{best} 'e göre farklı konumdadır.

- $SO_{P_{best}} = \{ (2,5) , (5,2) , (4,3) , (3,6) , (1,4) \}$

Bu bilgiler doğrultusunda önce $SO_{G_{best}}$ yer deęiřtirme miktarı ile c_2 çarpılır ve sonuç tam sayı deęilse üst tam sayıya yuvarlanır. Gösterimi $\lceil \cdot \rceil$ şeklindedir. $SO_{P_{best}}$ yer deęiřtirme miktarı ile c_1 çarpılır ve bir alt tam sayıya yuvarlanır. Gösterimi $\lfloor \cdot \rfloor$ şeklindedir.

Hızın Hesaplanması

Parçacıęın hızının hesaplanmasının sırasıyla işlem adımları ařaęıdaki gibidir.

1. Ghız miktarı = $c_2 * SO_{G_{best}} = \lceil 0.5 * 4 \rceil = 2$
2. Ghız = $\{(2,5) , (5,2)\}$ rastsal olarak 2 adet seçildi
3. Phız miktarı = $c_1 * SO_{P_{best}} = \lfloor 0.5 * 5 \rfloor = 2$
4. Ghız'da bulunan düęümler ve düęüm sıraları $SO_{P_{best}}$ 'de olanlar çıkartılır. Kalanlar içerisinde 2 adet yer deęiřtirme seçilir. Eęer iki adetten düşükse tamamı seçilir.
5. $SO_{P_{best}}$ 'den $(2,5), (5,2)$ yer deęiřtirmeleri $SO_{G_{best}}$ 'te oldukları için çıkarıldı
6. Phız = $\{ (1,4) , (3,6) \}$ kalan $SO_{P_{best}}$ elemanları içerisinde rastsal şekilde 2 tanesi seçildi.
7. Toplam hız miktarı = Ghız miktarı + Phız miktarı = 4
8. $v_{t+1,p} = \{(5,2), (1,4), (2,5), (3,6)\}$

Yukarıdaki adımlarda c_2 sosyal öğrenme katsayısı olduęu için global en iyi pozisyon ile işleme alındı . Aynı şekilde c_1 de biliřsel öğrenme katsayısı olduęu için parçacıęın kendi geçmişindeki en iyi pozisyonu ile işleme alındı. Ayrıca TSPLib'de

bulunan örnek ASP problemleri üzerinde geliştirilen R-PSO test edilmiş. Sosyal öğrenme katsayısı c_2 rastgele 0.5 ile 0.8 arasında seçilmesi daha iyi sonuçlar verildiği gözlemlenmiştir.

Bir sonraki iterasyonda parçacığın pozisyonunun hesaplanması

Hızda elde edilen yer değiştirmeler şu an ki pozisyona uygulanarak yeni pozisyon elde edilir. Bu işlem “ \oplus ” sembolü ile gösterilir. Denklem (2.5)’deki pozisyon denklemi, aşağıdaki Denklem (3.1)’deki gibi güncellenmiştir.

$$P_{t+1,p} = P_{t,p} \oplus v_{t+1,p} \quad (3.1)$$

Bu işlemler doğrultusunda örnekteki p. parçacığın bir sonraki iterasyonun pozisyonu aşağıdaki gibi hesaplanır.

1. $P_{t+1,p} = \{0,2,5,4,3,1,6\} \oplus \{(5,2), (1,4), (2,5), (3,6)\}$
2. $P_{t+1,p} = \{0,5,4,1,2,3,6\}$
3. $PU_{t+1,p} = 27$

Eğer elde edilen yeni pozisyonda öncül kuralları ihlal edilseydi yani 3 numaralı düğümün ziyaret sırası, 4 numaralı düğümün ziyaret sırasından önce gelseydi. Uygun çözüme çevirme işlemleri yapılırdı. Bu işlemde 3 numaralı düğümü, 4 numaralı düğüm hemen ardına eklenerek uygun pozisyon elde edilir. Uygun çözüme sahip olmayan bir pozisyonun uygun çözüme çevirme prosedürü (UÇP) pseudo kodlaması Şekil 3.32’de gösterilmiştir. Şekil 3.32’de verilen algoritma aynı zamanda kontrol edilen pozisyonun uygun çözüm olup olmadığını da kontrol eder. Eğer uygun çözüm ise kontrol edilen pozisyonda herhangi bir yer değiştirme işlemi gerçekleşmez.

Adım 0 : BAŞLA

$n = \text{düğüm sayısı};$

$P_{t+1,p}^i = \text{kontrol edilen pozisyonun } i. \text{ düğümü};$

Adım 1 :

$i = 0;$

Adım 2 :

$j = i. \text{ düğümün öncüllerinin arasından } P_{t+1,p} \text{ içerisindeki en yüksek ziyaret sırasına sahip olan öncül düğümünün ziyaret sırası};$

Eğer ($i < j$) **ise**

$P_{t+1,p}^i$ $i. \text{ düğümü } j. \text{ düğümün hemen arkasına yerleştirerek}$

$P_{t+1,p}$ pozisyonunu aşağıdaki gibi güncelle

$$P_{t+1,p} = P_{t+1,p} \oplus \{(i. \text{ düğüm}, j)\}$$

Adım 1' e git;

Değilse

Eğer ($i = n-1$) **ise**

Adım 3' e git;

Değilse

$i = i + 1;$

Adım 2'ye git;

Adım 3 : BİTİR

Şekil 3.32. Uygun çözüm kontrol ve UÇP Pseudo Kod

R-PSO pseudo kodları Şekil 3.33 ve Şekil 3.34'te gösterilmiştir. Bu gösterime göre R-PSO başlangıç çözümü üretme şekline göre iki versiyona sahiptir. Ayrıca R-PSO'da herhangi bir bitirme kriteri bulunmaz. Belirlenen sayıda iterasyon gerçekleştikten sonra elde edilen G_{best} ise R-PSO'da bulunan en iyi çözümü ifade eder.

Adım 0 : BAŞLA

Parçacık sayısını belirle;

İterasyon sayısını belirle;

Başlangıç çözümü için algoritma seç = {Rastsal v1 , Rastsal v2};

t = iterasyon;

p = parçacık;

t= 0;

Adım 1:

p = 0;

Adım 2 :

Eğer (p = parçacık sayısı) ise

t = t + 1;

Adım 1'e git;

Değilse

Eğer (t = 0) ise

$P_{t,p} =$ Seçilin başlangıç algoritmasına göre pozisyon üret;

$PU_{t,p} = P_{t,p}$

$H_{t,p} =$ Rastgele hız belirle

$Pbest_p = P_{t,p}$

$PUbest_p = PU_{t,p}$

$Gbest =$ Global en iyi pozisyonu

$GUbest = Gbest$ maliyetini

p = p + 1;

Adım 2'ye git

Değilse

Eğer (t != iterasyon sayısı - 1) ise

$P_{t,p} = P_{t-1,p} \oplus H_{t,p};$

$P_{t,p}$ Uygun çözüm kontrol ve uygun çözüme çevirme işlemleri

uygula;

$PU_{t,p} = P_{t,p}$ maliyetini hesapla

Eğer ($PU_{t,p} < PUbest_p$) ise

$Pbest_p = PU_{t,p};$

$Pbest_p = P_{t,p};$

Şekil 3.33. R-PSO Pseudo Kod Part I

Değilse (*Devam et*)

Eğer ($PU_{t,p} < GUbest$) ise

$$Gbest = P_{t,p};$$

$$GUbest = PU_{t,p};$$

Değilse (*Devam et*)

$H_{t+1,p}$ Hesapla;

$$p = p + 1;$$

Adım 2'ye git;

Değilse

Adım 3'e git;

Adım 3:

$Gbest$ ve $GUbest$ elde edilen en iyi sonucu göster

BİTİR.

Şekil 3.34. R-PSO Pseudo Kod Part II

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Bu bölümde geliştirilen çözüm yöntemi algoritmaları, KTYP senaryolarında rastgele oluşturulan problemlere uygulanmıştır ve elde edilen sonuçlara yer verilmiştir.

4.1. Deneysel Problemlerin Oluşturulması

Deneysel problemler oluşturulurken aşağıda verilen 4 adet KTUNK'e ait istenilen girdi verisine ihtiyaç vardır. Bu veriler doğrultusunda rastgele problemler oluşturulabilir.

1. $|B| = (0, \approx 12500]$ arası kitap miktarı
2. $|DB| = (0, |B|]$ arası danışma masasında bulunan kitap miktarı
3. $|T| = \begin{cases} |B| - |DB| < 36 & (0, |B| - |DB|] \\ |B| - |DB| \geq 36 & (0, 36] \end{cases}$ kitap bulunan çalışma masası miktarı
4. $|R| = \begin{cases} |B| < 141 & (0, |R|] \\ |B| \geq 141 & (0, 141] \end{cases}$ belirlenen $|B|$ miktarına göre raf sayısının miktarı

Birinci girdi verisi, KTUNK'te yaklaşık 12500 adet kitap olmasından dolayı KTUNK kitapları arasından rastgele $|B|$ kadar seçilmesini sağlamak için kullanılır. İkinci girdi verisi ise birinci veri sayesinde rastgele seçilen $|B|$ miktarı kadar kitaplar arasından $|DB|$ miktarı kadar rastgele seçilen kitapların danışma masasında olması durumudur. Üçüncü girdi verisi, KTUNK'te bulunan 36 çalışma masası arasından en fazla $|B| - |DB|$ miktarı kadar çalışma masası rastgele seçilebilir olması için gereklidir. Çünkü rastgele seçilen $|B|$ miktarı kadar kitap içerisinde $|DB|$ kadar rastgele kitap danışma masasında bulunur ve geriye $|B| - |DB|$ miktarı kadar kitap kalır. Kalan kitaplarda maksimum $|B| - |DB|$ kadar masada bulunabilir. Bu durumda üçüncü girdi verisi ile seçilen çalışma masalarının miktarı, buldukları konumlar ve üzerlerinde bulunan kitapların yerleştirileceği raflarla öncelik ilişkilerinden kaynaklı oluşturulacak problem çalışma masalarının geçerli olduğu senaryolarda problemin zorluk derecesini belirleyecektir. Dördüncü girdi verisi sayesinde rastgele seçilen kitapların kaç farklı rafa ait olduğunu belirtir. KTUNK'de toplam 141 adet raf bulunduğu için eğer $|B| < 141$ ise maksimum raf sayısı $|B|$ kadar olabilir. Fakat, $|B| \geq 141$ ise 141 adet raf olduğu için maksimum 141 raf

seçilir. Bu veriler sayesinde rastgele KTYP problemlerinin oluşturulması ile ilgili *pseudo* kod Şekil 4.1’de verilmiştir.

Adım 0 : Başla

- $|B|$ belirle;
- $|DB|$ belirle;
- $|T|$ belirle;
- $|R|$ belirle;

Adım 1 :

- Veri tabanından $|R|$ kadar rafı rastgele seç;
- Seçilen raflara ait $|B|$ kadar kitabı veri tabanından rastgele seç;
- $|T|$ kadar çalışma masasını rastgele seç;
- Seçilen kitaplardan rastgele $|DB|$ kadarını danışma masasına olsun;
- Seçilen kitaplardan rastgele $|B| - |DB|$ kadarını seçilen çalışma masalarında rastgele bulunsun;
- Hangi kitap hangi masada ve yerleştirilmesi gereken rafların bilgilerinin tamamını veri tabanına kaydet.

Adım 2 : Senaryo-1 için işlemler

- Adım1’de ortaya çıkan çalışma masalarında bulunan kitapların tamamı danışma masasında olduğu varsayılır;
- Danışma masası ve kitapların ait oldukları rafların maliyet matrisini Senaryo.1 olarak kaydet;

Adım 3 : Senaryo-2-3 için işlemler

- Adım 1’de ortaya çıkan tüm düğümlerin koordinatları, birbirlerine olan uzaklıkları, düğüm numaraları gibi tüm bilgileri veri tabanından getir;
- Adım’1’de seçilen kitapların ait oldukları rafları ve şu an buldukları masaların tüm bilgilerini Senaryo-2-3 olarak veri tabanına kaydet;
- Düğümlerin maliyet matrisini Senaryo-2-3 veri olarak tabanına kaydet;

Adım 4 : Senaryo 4-5 için işlemler

- Adım1’de ortaya çıkan çalışma masalarında bulunan kitapların tamamı toplama masasında olduğu varsayılır; (M101 numaralı masa)
- Danışma masası, kitapların ait oldukları rafların ve toplama masasının olduğu maliyet matrisini Senaryo.4-5 olarak kaydet;

Adım 5 : BİTİR

Şekil 4.1. Rastsal KTYP Problemleri Oluşturma Mekanizmasının Pseudo Kodu

Senaryo-6-7-8 için maliyet matrisleri oluşturulmaz. Bu senaryolar dinamik olduğu için veriler anlık olarak veri tabanından çekilerek işlem yapılır. Geliştirilen algoritmaların ihtiyaç halinde maliyet matrisleri veri tabanında tutulan KTUNK’e ait tüm düğümlerin

birbirlerine olan uzaklığının bulunduğu matristen SQL komutları ile anlık olarak çekilerek oluşturulur. Bu oluşturulan dinamik maliyet matrisleri veri tabanında tutulmaz.

Girdi verileri ile rastgele oluşturulan KTYP problemlerinin çıktıları ise aşağıdaki gibidir.

1. Hangi kitap hangi masada ve ait olduğu raf neresi gibi bilgileri veren tablo. Bu tablo toplama ve yerleştirme verisi olarak adlandırılır.
2. Problemlerin senaryo bazında maliyet matrisleri

Rastgele oluşturulan KTYP problemleri belirli bir standartta göre isimlendirilir. Örnek olarak girdi verileri $|B| = 20$, $|DB| = 5$, $|T| = 10$, $|R| = 7$ bu şekilde olan bir KTYP problemi KTYP.B20.N18.T10.R7 olarak adlandırılır. Burada $N = |T| + |R| + 1$ olarak problemdeki toplam düğüm sayısını ifade eder. Düğüm sayıları her senaryoda farklı olabilir ancak aynı problem üzerinden senaryoların karşılaştırılması yapılacağı için problem ismi oluşturma standardı olarak bu şekilde belirlenmiştir.

Tablo 6. KTYP.B20.N18.T10.R7 adlı probleme ait kitapların toplama ve yerleştirme verisi

Kitap ID	Kitap İsmi	Bulunduğu Masa	Ait olduğu Raf
15440	Oxford'un Öküzü : Kelimelerin Şaşırtıcı Kökenleri	DM	R3
22905	Fluid mechanics	DM	R70
18257	Manken	DM	R22
21421	Feynman fizik dersleri	DM	R92
24771	Stoichiometry	DM	R49
18171	Beklenmedik anda : kendi mutlu sonunuzu yaşamamız dileğiyle...	M171	R22
21229	Kütleçekim : ağır bir konu üzerine kısa gezinti	M171	R89
15472	Ziyan balkonu	M11	R3
18231	Azerbaycan edebiyatı tarihinin etütleri	M11	R22
16523	Cennette iki yıl	M81	R9
16547	İçimdeki Şarkı	M82	R9
21175	Classical mechanics	M92	R89
18208	Sana güvenmiştim	M92	R22
21407	Fizik ilkeler ve pratik	M92	R92
24770	Process Calculations	M101	R49
18191	Mehir	M111	R22
15376	Rüzgâr bizi götürecektir	M32	R3
21189	Tarih boyunca Türklerde gökbilim	M132	R89
24758	Taşınma süreçleri ve ayırma süreci ilkeleri : Temel işlemleri içerir	M132	R49
24754	Engineering chemistry	M52	R49

Rastgele oluşturulan KTYP problemlerinde yerleştirilmesi gereken kitaplar ve ait oldukları raflar her senaryoda aynıdır. Senaryolar arasında iki fark vardır. Bu farklardan

birincisi kitapların buldukları masalar yani arz düğümleri bazı senaryolarda farklıdır. İkinci fark ise toplama ve yerleştirme kuralına göre oluşan maliyet matrisidir. Tablo 6’da verilen kitaplar Senaryo-2, Senaryo-3 ve Senaryo-6’da aynı masalardadır. Senaryo-6 dinamik olduğu için geliştirilen algoritmalar, hangi kitabın hangi masada olduğunu, ilgili masaların ziyareti gerçekleşince öğrenir.

Senaryo-1 için Tablo 6’da verilen kitapların tamamı danışma masasındadır. Senaryo-1’de maliyet matrisi bu durum dikkate alınarak oluşturulur. Senaryo-4, Senaryo-5, Senaryo-7 ve Senaryo-8’de ise bu tabloda danışma masasındaki kitaplar haricindeki tüm kitaplar toplama masasındadır. Senaryo-7 ve Senaryo-8 dinamik olduğu için geliştirilen algoritmalar, kullanılmış kitapları toplama masası ziyareti gerçekleşince öğrenir.

4.2. Geliştirilen Çözüm Yöntemlerinin KTYP Senaryoları ile Karşılaştırılması

Tezin bu bölümünde rastgele oluşturulan KTYP problemleri senaryolara uygulanmış ve KTYP için geliştirilen algoritmaların toplama ve yerleştirme sonucu ortaya çıkan rota uzunlukları ve çözüm süreleri karşılaştırılmıştır.

R-PSO için algoritmada ihtiyaç duyulan parametreler algoritmanın etkinliği konusunda önemlidir.. Bu nedenle rastgele oluşturulmuş KTYP problemleri ve TSPLib’de bulunan ASP problemleri üzerinde toplamda 100 deneme neticesinde $t=14$, $p=14$, $c2 = 0.5$ ve 0.8 arası rastgele seçilmesinin R-PSO’nun en etkin olduğu parametre değerleri olduğu gözlemlenmiştir. R-PSO ile çözülen problemlerde parametreler bu paragrafta belirtilen değerler üzerinden çözümler elde edilmiştir.

Geliştirilen çözüm yöntemleri iki versiyona (v_1 ve v_2) sahip olduğu için toplamda 10 algoritma senaryolara uygulanarak test edilmiştir. R-PSO hariç diğer çözüm yöntemlerinin her iki versiyonu Senaryo-1-3-5 ve dinamik olan Senaryo-6-7-8’lerde aynı sonucu verecektir. Çünkü Senaryo-1-3-5 önce kitaplar toplanıp daha sonra yerleştirme işlemi yapıldığı için ilk düğümden sonra ilgili arz düğümlerine uğradıktan sonra tüm talep düğümleri gidilebilir olduğu için versiyonlar aynı sonuç vermektedir. Ayrıca Senaryo-6-7-8’de ise problem dinamik olduğu için toplanan kitapların talep noktaları ve gidilmemiş arz noktaları ziyaret edileceği için herhangi bir öncelik ilişkisi kurulamaz. Dinamik rotalar arz noktasından kitap varsa alınıp sonra tekrar yeni rota oluşturduğu için mevcut rotada arz ve talep noktaları arasında herhangi bir öncelik ilişkisi kalmamaktadır. Dinamik senaryolarda sadece başlangıç ve bitiş düğümü için öncelik ilişkisi olduğu için

R-PSO hariç diğer tüm çözüm yöntemleri aynı sonucu vermektedir. R-PSO versiyonlarının her senaryoda farklı sonuçlar elde etmesinin nedeni R-PSO'nun rastsal yapısından kaynaklıdır.

Oluşturulan problemlerin zorluk derecesi Senaryo-2'nin düğüm sayısına bağlı olarak değişmektedir. Senaryo-2, çalışma masalarının olduğu ve kullanılmış kitapların hangi çalışma masasında bulunduğu ve hangi rafa yerleştirileceği daha önceden bilinen bir senaryodur. Dolayısı ile diğer senaryolara ait tüm veri setleri bu senaryo üzerinden oluşturulur.

Tablo 7. Rastgele oluşturulmuş KTYP.B4.N6.T2.R4.1 adlı probleme ait senaryoların, geliştirilen algoritmalar ile çözümlenerek elde edilen sonuçlar

PROBLEM	SENARYO	N	Mesafe D	EYKS		BEYKS		SÜP_A		SÜP_A-2		PSO		ORTALAMA	STANDART SAPMA		
				Süre T		v1	v2	v1	v2	v1	v2	v1	v2			v1	v2
				D (cm)	T (sn)	v1	v2	v1	v2	v1	v2	v1	v2			v1	v2
KTYP.B4.N6.T2.R4.1	S-1	4	D (cm)	9230		9230		9230		10670		9230	9230	9470	587,9		
			T (sn)	0		0		0		0		3	2	1	1,3		
	S-2	6	D (cm)	11474	10815	10951	10951	10951	10951	12782	12782	9239	10815	11171	1025,4		
			T (sn)	2	0	0	0	0	0	0	0	11	5	2	3,6		
	S-3	6	D (cm)	11474		9239		10925		12213		11373	9239	10744	1236,9		
			T (sn)	0		0		0		0		11	5	3	4,5		
	S-4	5	D (cm)	11825	10815	10951	10600	10600	10600	12229	12229	9959	9239	10905	961,0		
			T (sn)	0	0	0	0	0	0	0	0	10	5	2	3,4		
	S-5	5	D (cm)	11825		9239		9239		11133		10542	9239	10203	1131,2		
			T (sn)	0		0		0		0		9	5	2	3,8		
	S-6	40	D (cm)	19265		16969		31754		61376		48472	52292	38355	18395,2		
			T (sn)	0		0		0		0		41	38	13	20,4		
	S-7	5	D (cm)	11825		10951		10600		12229		11277	11277	11360	588,2		
			T (sn)	0		0		0		0		3	3	1	1,5		
	S-8	5	D (cm)	11825		9239		9239		11133		10815	10815	10511	1052,1		
			T (sn)	0		0		0		0		5	4	2	2,3		
ORTALAMA D				12343	10786	10846	9602	12817	11393	17971	15974	15113	15268	14090	3122		
ORTALAMA T				0,25	0,00	0,00	0,00	0,00	0,00	0,00	0,00	11,63	8,38	3,10	5,13		
Standart Sapma D				2932	3011,55	2615	2616,35	7690	7690,18	17553	17553	13504	14985,07				
Standart Sapma T				1	0	0	0	0	0	0	0	12	12,02				

Rastgele üretilen KTYP.B4.N6.T2.R4.1 adlı problem üzerinden algoritmaların ve senaryoların karşılaştırılması Tablo 7'de verilmiştir. Senaryo-2'deki düğüm sayısı 6'dır. Bu tabloda probleme ait senaryoların çözümlerinde elde edilen toplam mesafe ve algoritmaların çözüm süreleri yer almaktadır. Bu problemin tüm senaryoları, geliştirilen algoritmalar ile çözülmüş ve elde edilen mesafelerin ve çözüm sürelerinin; ortalaması ve standart sapması hesaplanmıştır. Bu problem bazında mesafe ortalamasına göre BEYKS v2, diğer algoritmalarla göre daha iyi sonuçlar vermiştir. Senaryo-1 ise en az mesafe ortalamasına ve standart sapmaya sahip senaryodur. SÜP_A-2 algoritması ise ortalama mesafe bazında diğer algoritmalarla nazaran daha kötü sonuçlar verdiği gözlemlenmiştir.

Tablo 8. Rastgele oluşturulmuş KTYP.B50.N13.T3.R10.3 adlı probleme ait senaryoların, geliştirilen algoritmalar ile çözümlenerek elde edilen sonuçlar

PROBLEM	SENARYO	N	Mesafe D		EYKS		BEYKS		SÜP_A		SÜP_A-2		PSO		ORTALAMA	STANDART SAPMA
			Süre T	v1	v2	v1	v2	v1	v2	v1	v2	v1	v2			
KTYP.B50.N13.T3.R10.3	S-1	10	D(cm)	13467		12875		14308		17341		13508	13627	14312	1667,4	
			T(sn)	0		0		0		0		5	8	1	2,8	
	S-2	13	D(cm)	18083	18083	21843	21843	23666	23666	22007	22007	18069	15465	20473	2811,4	
			T(sn)	5	5	5	5	4	5	4	5	77	78	19	30,7	
	S-3	13	D(cm)	16142		18616		20049		19887		18157	17940	18549	1498,2	
			T(sn)	4		6		7		5		70	73	18	28,1	
	S-4	11	D(cm)	13537	13537	14646	14646	15873	15873	20518	20518	14429	14432	15801	2607,4	
			T(sn)	3	3	3	5	3	3	3	3	52	56	13	21,4	
	S-5	11	D(cm)	13537		13540		14973		18773		14622	13473	14974	2094,6	
			T(sn)	3		3		3		3		52	50	13	20,1	
	S-6	51	D(cm)	23021		28530		41765		69070		61032	55121	44093	18377,4	
			T(sn)	0		0		0		0		64	70	14	28,2	
	S-7	42	D(cm)	13537		16394		19102		57865		44733	48592	30712	19033,3	
			T(sn)	0		0		0		0		41	35	8	16,1	
	S-8	41	D(cm)	13537		13540		14973		58307		49209	46929	29685	20528,3	
			T(sn)	0		0		0		0		30	22	5	11,1	
ORTALAMA D			15608	13873	17498	15554	20589	18301	35471	31530	29220	28197	23575	8577		
ORTALAMA T			1,88	1,88	2,13	2,38	2,13	2,25	1,88	2,00	48,88	49,00	11,41	19,83		
Standart Sapma D			3446	3445,56	5397	5396,96	9143	9143,02	22062	22062	19190	18429,84				
Standart Sapma T			2	2,1002	2	2,66927	3	2,71241	2	2,2678	23	25,39				

Rastgele üretilen KTYP.B50.N13.T3.R10.3 adlı problem üzerinden algoritmaların ve senaryoların karşılaştırılması Tablo 8'de verilmiştir. Senaryo-2'deki düğüm sayısı 13'tür. Bu problemde algoritmaların tüm senaryolarda elde ettiği ortalama mesafe değerlerine bakılırsa en iyi sonuç veren algoritma EYKS v2'dir. Bu tabloda algoritmaların ortalama çözüm sürelerine ve çözüm sürelerinin standart sapmalarına bakıldığında EYKS v1, EYKS v2 ve SÜP-A v1 hızlı sonuçlar üretmektedir. PSO hariç diğer çözümler de ortalama çözüm süreleri en düşük değere çok yakındır. Yine bu tabloda senaryolarda ortalama kat edilen mesafe verilerine bakarak en iyi senaryonun, Senaryo-1 olduğu görülmektedir.

Tablo 9. Rastgele oluşturulmuş KTYP.B20.N17.T7.R10.2 adlı probleme ait senaryoların, geliştirilen algoritmalar ile çözümlenerek elde edilen sonuçlar

PROBLEM	SENARYO	N	Mesafe D	EYKS		BEYKS		SÜP_A		SÜP_A-2		PSO		ORTALAMA	STANDART SAPMA
			Süre T	v1	v2	v1	v2	v1	v2	v1	v2	v1	v2		
KTYP.B20.N17.T7.R10.2	S-1	10	D (cm)	9982		11066		12295		25528		11533	13714	14020	5774,4
			T (sn)	0		0		0		0		5	5	2	2,6
	S-2	17	D (cm)	13821	16761	27155	21523	27304	22651	36869	30566	19423	22786	23886	6797,5
			T (sn)	1	0	1	0	1	0	1	0	26	14	4	8,7
	S-3	17	D (cm)	16479		20322		21551		34530		24311	23513	23451	6090,9
			T (sn)	1		1		1		1		29	11	7	11,3
	S-4	11	D (cm)	10856	15918	15376	18826	16665	19966	24964	24964	13321	15877	17673	4605,7
			T (sn)	1	0	1	0	1	0	1	0	20	8	3	6,4
	S-5	11	D (cm)	10856		11940		13169		26325		16435	14830	15593	5623,6
			T (sn)	1		0		0		1		17	8	5	6,8
	S-6	49	D (cm)	23522		37651		47802		75932		60576	63910	51566	19086,7
			T (sn)	0		0		0		0		127	85	35	56,3
	S-7	11	D (cm)	10856		16466		16870		22447		16872	15429	16490	3705,3
			T (sn)	0		0		0		0		8	8	3	4,1
	S-8	11	D (cm)	10856		11940		13169		26325		12357	17289	15323	5826,6
			T (sn)	0		0		0		0		9	9	3	4,6
ORTALAMA D			13404	12803	18990	16637	21103	18608	34115	29624	21854	23419	22250	7189	
ORTALAMA T			0,50	0,25	0,38	0,13	0,38	0,13	0,50	0,25	30,13	18,50	7,76	12,62	
Standart Sapma D			4624	4669,20	9245	8655,45	11908	11575,18	17613	17613	16191	16759,59			
Standart Sapma T			1	0,4629	1	0,3536	1	0,35355	1	0,4629	40	27,00			

Rastgele üretilen KTYP.B20.N17.T7.R10.2 adlı problem üzerinden algoritmaların ve senaryoların karşılaştırılması Tablo 9'de verilmiştir. Problemin Senaryo-2'de ki düğüm sayısı 17'dir. Bu problemde, EYKS v2 diğer algoritmalarla göre daha iyi sonuçlar vermiştir. Ortalama çözüm süresi ve çözüm sürelerinin standart sapması en düşük olan algoritmalar ise BEYKS v2 ve SÜP-A v2'dir. Bu algoritmalar diğer algoritmalarla göre ortalama daha hızlı sonuçlar vermiştir. Senaryo-1 ise algoritmaların sonuçlarına göre en düşük, mesafe ortalamasına ve ortalama çözüm süresine sahiptir. Senaryolar arasında algoritmaların en düşük, mesafe standart sapması ise Senaryo-7'ye aittir.

Tablo 10. Rastgele oluşturulmuş KTYP.B45.N25.T10.R15.3 adlı probleme ait senaryoların, geliştirilen algoritmalar ile çözümlenerek elde edilen sonuçlar

PROBLEM	SENARYO	N	Mesafe D		EYKS		BEYKS		SÜP_A		SÜP_A-2		PSO		ORTALAMA	STANDART SAPMA
			Süre T	v1	v2	v1	v2	v1	v2	v1	v2	v1	v2			
KTYP.B45.N25.T10.R15.3	S-1	15	D(cm)	11059		11361		11817		25194		17217	16898	15591	5463,4	
			T(sn)	0		0		0		0		8	8	3	4,1	
	S-2	25	D(cm)	19342	20442	29661	23865	31103	25154	41635	38019	30266	31801	29129	7171,0	
			T(sn)	5	1	4	1	4	1	4	0	88	21	13	27,1	
	S-3	25	D(cm)	20925		22719		22797		38817		30303	31541	27850	6924,3	
			T(sn)	5		8		4		4		82	20	21	30,7	
	S-4	16	D(cm)	13388	15494	14743	17973	15850	19740	25182	25182	16361	15061	17897	4218,9	
			T(sn)	4	1	3	0	4	0	3	0	55	19	9	17,1	
	S-5	16	D(cm)	13388		12937		14044		24179		16447	17813	16468	4221,4	
			T(sn)	3		3		2		3		55	17	14	21,0	
	S-6	64	D(cm)	27105		46484		66512		78231		72215	68075	59770	19254,3	
			T(sn)	0		1		0		0		211	231	74	114,2	
	S-7	16	D(cm)	13388		18019		19885		30304		26505	21550	21609	6050,2	
			T(sn)	0		0		0		0		12	21	6	9,0	
	S-8	16	D(cm)	13388		12937		14044		24179		17634	18106	16715	4265,4	
			T(sn)	0		0		0		0		15	19	6	8,9	
ORTALAMA D			16498	15021	21108	18477	24507	21555	35965	31567	28369	27606	25629	7196		
ORTALAMA T			2,13	1,25	2,38	1,63	1,75	0,88	1,75	0,88	65,75	44,50	17,98	29,01		
Standart Sapma D			5457	5424,61	11942	11346,60	18076	17693,03	18396	18281	18743	17589,32				
Standart Sapma T			2	1,8323	3	2,774244	2	1,45774	2	1,6421	66	75,48				

Rastgele üretilen KTYP.B45.N25.T10.R15.3 adlı problem üzerinden algoritmaların ve senaryoların karşılaştırılması Tablo 10'da verilmiştir. Senaryo-2'deki düğüm sayısı 25'tir. Bu problemde algoritmaların tüm senaryolarda elde ettiği ortalama mesafe değerlerine bakılırsa en iyi sonuç veren algoritma EYKS v2'dir. Bu tabloda algoritmaların ortalama çözüm sürelerine ve çözüm sürelerinin standart sapmalarına bakıldığında başta SÜP_A v2 ve SÜP_A-2 v2 hızlı sonuçlar üretmektedir. PSO hariç diğer algoritmaların ikinci versiyonları da ortalama çözüm süreleri en düşük değere çok yakındır. Yine bu tabloda senaryolarda ortalama kat edilen mesafe verilerine bakarak en iyi senaryonun, Senaryo-1 olduğu görülmektedir.

Tablo 11. Rastgele oluşturulmuş KTYP.B30.N35.T20.R15.2 adlı probleme ait senaryoların, geliştirilen algoritmalar ile çözümlenerek elde edilen sonuçlar

PROBLEM	SENARYO	N	Mesafe D	EYKS		BEYKS		SÜP_A		SÜP_A-2		PSO		ORTALAMA	STANDART SAPMA		
				Süre T		v1	v2	v1	v2	v1	v2	v1	v2			v1	v2
				v1	v2	v1	v2	v1	v2	v1	v2	v1	v2				
KTYP.B30.N35.T20.R15.2	S-1	15	D(cm)	13020		13406		14674		24022		19025	16597	16791	4182,7		
			T(sn)	0		0		0		0		8	7	3	3,9		
	S-2	35	D(cm)	23584	23108	33903	25548	44522	31080	56932	52677	43800	44616	37977	12216,1		
			T(sn)	4	0	3	0	3	0	3	0	3	0	72	24	11	22,6
	S-3	35	D(cm)	22858		25241		30640		53654		48327	46817	37923	13233,6		
			T(sn)	3		3		3		4		67	25	18	25,8		
	S-4	16	D(cm)	13634	14645	17511	17511	18563	18563	27065	27065	21152	19824	19553	4532,8		
			T(sn)	1	0	1	0	1	0	1	0	27	11	4	8,7		
	S-5	16	D(cm)	13814		14280		15548		22289		19255	17125	17052	3247,0		
			T(sn)	1		1		1		1		26	11	7	10,2		
	S-6	56	D(cm)	23754		53147		70354		82767		70900	65861	61131	20648,3		
			T(sn)	0		0		0		0		288	294	97	150,3		
	S-7	16	D(cm)	13814		18376		19428		30091		20865	21314	20648	5348,2		
			T(sn)	0		0		0		0		12	11	4	5,9		
	S-8	16	D(cm)	13814		14280		15548		22289		16983	18906	16970	3201,8		
			T(sn)	0		0		0		0		11	11	4	5,7		
ORTALAMA D				17287	15425	23768	20199	28660	23982	39889	34984	32538	31383	28506	8326		
ORTALAMA T				1,13	0,50	1,00	0,50	1,00	0,50	1,13	0,63	63,88	49,25	18,30	29,14		
Standart Sapma D				5074	4900,14	13751	13176,70	19714	18716,71	22201	21781	19693	18572,79				
Standart Sapma T				2	1,069	1	1,069045	1	1,06904	2	1,4079	94	99,11				

Rastgele üretilen KTYP.B30.N35.T20.R15.2 adlı problem üzerinden algoritmaların ve senaryoların karşılaştırılması Tablo 11’de verilmiştir. Senaryo-2’deki düğüm sayısı 35’tir. Bu problemde algoritmaların tüm senaryolarda elde ettiği ortalama mesafe değerlerine bakılırsa en iyi sonuç veren algoritma EYKS v2’dir. En hızlı çalışan algoritmalar, ortalama çözüm süreleri ve çözüm sürelerinin standart sapmalarını dikkate alındığında EYKS v2, BEYKS v2 ve SÜP-A v2 algoritmalarıdır. Senaryolarda algoritmalar sonucu ortalama katedilen mesafelere bakıldığı zaman, Senaryo-1 diğer senaryolara göre iyi sonuçlar verse de Senaryo-5 ve Senaryo-8’de başarılı sonuçlar verdiği gözlemlenmiştir. En hızlı sonuçların üretildiği senaryo ise Senaryo-1’dir.

Tablo 12. Rastgele oluşturulmuş KTYP.B30.N50.T20.R30.2 adlı probleme ait senaryoların, geliştirilen algoritmalar ile çözümlenerek elde edilen sonuçlar

PROBLEM	SENARYO	N	Mesafe D	EYKS		BEYKS		SÜP_A		SÜP_A-2		PSO		ORTALAMA	STANDART SAPMA
				v1	v2	v1	v2	v1	v2	v1	v2	v1	v2		
KTYP.B30.N50.T20.R30.2	S-1	30	D(cm)	16902	18623	27231	49889	41915	45099	31230	13997,1				
			T(sn)	0	0	0	0	16	15	3	6,5				
	S-2	50	D(cm)	29596	28427	49482	33437	62181	41616	78103	79751	75202	68908	54670	20614,0
			T(sn)	8	1	8	1	7	0	7	0	127	34	19	39,1
	S-3	50	D(cm)	31124	32379	43638	76623	71136	71202	50987	20326,9				
			T(sn)	7	6	6	5	110	33	19	33,1				
	S-4	31	D(cm)	18333	20538	21289	23311	27815	30335	49306	51880	38074	35746	31663	11893,9
			T(sn)	3	0	4	0	3	0	3	0	59	18	9	18,4
	S-5	31	D(cm)	16489	19497	28105	47404	41234	43675	30790	12934,3				
			T(sn)	3	4	3	2	54	22	10	16,6				
	S-6	66	D(cm)	27022	49983	73483	97690	76744	86142	65924	26232,4				
			T(sn)	0	0	0	0	318	471	79	170,2				
	S-7	31	D(cm)	18333	21289	27815	49306	40349	39525	31336	12288,1				
			T(sn)	0	0	0	0	18	26	4	9,5				
	S-8	31	D(cm)	16489	19497	28105	47404	41574	41340	30590	12726,9				
			T(sn)	0	0	0	0	17	27	4	9,6				
ORTALAMA D				21786	19480	29005	24224	39797	33370	61966	55550	53279	53955	40899	16377
ORTALAMA T				2,63	1,38	2,75	1,38	2,38	1,13	2,13	0,88	89,88	80,75	18,53	37,87
Standart Sapma D				6318	6000,40	13514	10869,80	18396	15930,65	19427	19409	17565	18670,42		
Standart Sapma T				3	2,5036	3	2,326094	3	2,23207	3	1,8077	101	157,82		

Rastgele üretilen KTYP.B30.N50.T20.R30.2 adlı problem üzerinden algoritmaların ve senaryoların karşılaştırılması Tablo 12’de verilmiştir. Senaryo-2’deki düğüm sayısı 50’dir. Bu problemde algoritmaların tüm senaryolarda elde ettiği ortalama mesafe değerlerine bakılırsa en iyi sonuç veren algoritma EYKS v2’dir. Bu tabloda algoritmaların ortalama çözüm sürelerine bakıldığında başta SÜP_A-2 v2 olmak üzere sırasıyla SÜP_A v2, BEYKS v2 ve EYKS v2 hızlı sonuçlar üretmektedir. Yine bu tabloda senaryolarda ortalama kat edilen mesafe verilerine bakarak en iyi senaryonun, Senaryo-8 olduğu görülmektedir.

Tablo 13. Rastgele oluşturulmuş KTYP.B45.N50.T20.R30.3 adlı probleme ait senaryoların, geliştirilen algoritmalar ile çözümlenerek elde edilen sonuçlar

PROBLEM	SENARYO	N	Mesafe D	EYKS		BEYKS		SÜP_A		SÜP_A-2		PSO		ORTALAMA	STANDART SAPMA		
				Süre T		v1	v2	v1	v2	v1	v2	v1	v2			v1	v2
				D(cm)	T(sn)												
KTYP.B45.N50.T20.R30.3	S-1	30	D(cm)	16026		14423		24025		44522		43336	40933	30544	14000,8		
			T(sn)	0		0		0		0		19	25	7	11,5		
	S-2	50	D(cm)	29002	26523	47140	31004	54942	41579	78715	69624	73463	71443	52344	20059,6		
			T(sn)	16	6	13	2	13	1	13	2	224	36	33	68,0		
	S-3	50	D(cm)	26614		30634		38959		69305		71769	74910	52032	22298,7		
			T(sn)	15		14		13		13		229	40	54	86,4		
	S-4	31	D(cm)	15288	19931	20521	23153	28539	31989	46621	46621	39061	46240	31796	12141,5		
			T(sn)	9	2	8	1	9	1	8	1	160	32	23	49,0		
	S-5	31	D(cm)	15288		17208		24899		43362		40818	37667	29874	12331,7		
			T(sn)	9		9		9		8		149	29	36	56,2		
	S-6	73	D(cm)	28681		56688		76313		128786		95889	82392	78125	34081,8		
			T(sn)	1		1		0		0		418	375	133	204,9		
	S-7	34	D(cm)	17677		21067		29655		52769		41782	43424	34396	13807,9		
			T(sn)	0		1		0		0		27	18	8	11,8		
	S-8	31	D(cm)	15288		17208		24899		43362		39113	42464	30389	12822,9		
			T(sn)	0		0		0		0		26	19	8	11,8		
ORTALAMA D				20483	18448	28111	23487	37779	32480	63430	55372	55654	54934	42437	17693		
ORTALAMA T				6,25	4,13	5,75	3,50	5,50	3,00	5,25	3,00	156,50	71,75	37,53	62,46		
Standart Sapma D				6393	5640,57	15664	13669,05	18720	17357,17	29517	29015	21742	18060,51				
Standart Sapma T				7	5,4886	6	5,154748	6	5,07093	6	4,8697	137	122,77				

Rastgele üretilen KTYP.B45.N50.T20.R30.3 adlı problem üzerinden algoritmaların ve senaryoların karşılaştırılması Tablo 13'te verilmiştir. Senaryo-2'deki düğüm sayısı 50'dir. Bu problemde algoritmaların tüm senaryolarda elde ettiği ortalama mesafe değerlerine bakılırsa en iyi sonuç veren algoritma EYKS v2'dir. Bu tabloda algoritmaların ortalama çözüm sürelerine bakıldığında SÜP_A v2 ve SÜP_A-2 v2 algoritmaları hızlı sonuçlar üretmektedir. Yine bu tabloda senaryolarda ortalama kat edilen mesafe verilerine bakarak en iyi senaryonun, Senaryo-5 olduğu görülmektedir.

Tablo 14. Rastgele oluşturulmuş KTYP.B50.N76.T36.R40.2 adlı probleme ait senaryoların, geliştirilen algoritmalar ile çözümlenerek elde edilen sonuçlar

PROBLEM	SENARYO	N	Mesafe D		EYKS		BEYKS		SÜP_A		SÜP_A-2		PSO		ORTALAMA	STANDART SAPMA
			Süre T		v1	v2	v1	v2	v1	v2	v1	v2	v1	v2		
			D(cm)	T(sn)												
KTYP.B50.N76.T36.R40.2	S-1	40	D(cm)		17160		17798		24265		58773		54406	49629	37005	19292,7
			T(sn)	0		0		0		0		19		20	7	10,1
	S-2	76	D(cm)	27191	27191	63019	63019	86434	86434	106079	106079	110665	111134	78725	32473,4	
			T(sn)	10	10	10	10	9	11	9	10	173	189	44	72,3	
	S-3	76	D(cm)		32165		32548		50501		106373		114238	111159	74497	40167,3
			T(sn)		9		9		9		9		161	163	60	79,0
	S-4	41	D(cm)	17785	17785	22372	22372	26380	26380	57344	57344	58152	53375	35929	18024,0	
			T(sn)	3	3	4	3	3	3	3	3	67	71	16	27,8	
	S-5	41	D(cm)		16957		18672		25139		56279		57015	52521	37764	19431,9
			T(sn)		3		3		3		3		71	74	26	35,9
	S-6	84	D(cm)		28325		69870		92508		112980		80124	104046	81309	30282,0
			T(sn)		0		1		0		0		627	969	266	425,9
	S-7	41	D(cm)		20297		22372		26380		57344		55690	56392	39746	18437,8
			T(sn)		0		0		0		0		24	38	10	16,6
	S-8	41	D(cm)		16957		18672		25139		56279		60946	54065	38676	20483,0
			T(sn)		0		0		0		0		26	41	11	17,9
	ORTALAMA D			22105	19649	33165	29480	44593	39638	76431	67939	73905	74040	52956	24824	
	ORTALAMA T			3,13	3,13	3,38	3,25	3,00	3,25	3,00	3,13	146,00	195,63	55,09	85,69	
	Standart Sapma D			6155	6154,53	21139	21139,39	29069	29068,86	26630	26630	25157	28910,22			
	Standart Sapma T			4	4,1555	4	4,062019	4	4,39968	4	4,1555	203	318,29			

Rastgele üretilen KTYP.B50.N76.T36.R40.2 adlı problem üzerinden algoritmaların ve senaryoların karşılaştırılması Tablo 14'te verilmiştir. Senaryo-2'deki düğüm sayısı 76'dır. Bu problemde algoritmaların tüm senaryolarda elde ettiği ortalama mesafe değerlerine bakılırsa en iyi sonuç veren algoritma EYKS v2'dir. Bu tabloda algoritmaların ortalama çözüm sürelerine ve çözüm sürelerinin standart sapmalarına bakıldığında başta SÜP_A v1 ve SÜP_A-2 v1 hızlı sonuçlar üretmektedir. PSO hariç diğer çözümler de ortalama çözüm süreleri en düşük değere çok yakındır. Yine bu tabloda senaryolarda ortalama kat edilen mesafe verilerine bakarak en iyi senaryonun, Senaryo-4 olduğu görülmektedir.

Tablo 15. Rastgele oluşturulmuş KTYP.B150.N76.T35.R41.3 adlı probleme ait senaryoların, geliştirilen algoritmalar ile çözümlenerek elde edilen sonuçlar

PROBLEM	SENARYO	N	Mesafe D		EYKS		BEYKS		SÜP_A		SÜP_A-2		PSO		ORTALAMA	STANDART SAPMA
			Süre T		v1	v2	v1	v2	v1	v2	v1	v2	v1	v2		
			D (cm)	T (sn)												
KTYP.B150.N76.T35.R41.3	S-1	41	D (cm)	18735		17200		29020		56703		51003	56415	38179	18666,7	
			T (sn)	0		0		0		0		22	21	7	11,1	
	S-2	76	D (cm)	31651	34129	95104	33065	105876	59711	108495	100577	103802	99803	77221	33451,2	
			T (sn)	17	2	17	2	16	1	24	1	311	58	45	95,1	
	S-3	76	D (cm)	30968		31952		54718		106037		106952	101846	72079	37032,5	
			T (sn)	15		15		15		15		247	61	61	92,8	
	S-4	42	D (cm)	18094	24317	20449	24629	32269	36487	56084	53515	52803	51727	37037	15171,1	
			T (sn)	5	1	6	1	5	0	5	0	99	32	15	30,9	
	S-5	42	D (cm)	18094		18074		29894		60706		53410	53075	38876	19155,6	
			T (sn)	5		5		4		4		84	39	24	32,7	
	S-6	122	D (cm)	36810		143404		168677		148622		134959	143755	129371	46729,3	
			T (sn)	1		2		1		1		798	821	271	417,4	
	S-7	53	D (cm)	18094		25181		46912		84896		76196	96082	57894	32549,0	
			T (sn)	0		0		0		0		50	51	17	26,1	
	S-8	51	D (cm)	18094		18074		29894		64867		70735	70359	45337	25986,6	
			T (sn)	0		0		0		0		47	60	18	27,9	
ORTALAMA D			23818	22138	46180	34620	62158	50590	85801	75103	81233	84133	61999	28593		
ORTALAMA T			5,38	3,00	5,63	3,13	5,13	2,63	6,13	2,63	207,25	142,88	57,20	91,75		
Standart Sapma D			7912	7938,65	47164	42650,74	50169	46689,17	33101	32773	30913	32137,25				
Standart Sapma T			7	5,127	7	5,083236	7	5,18066	9	5,1807	260	274,38				

Rastgele üretilen KTYP.B150.N76.T35.R41.3 adlı problem üzerinden algoritmaların ve senaryoların karşılaştırılması Tablo 15'te verilmiştir. Senaryo-2'deki düğüm sayısı 76'dır. Bu problemde algoritmaların tüm senaryolarda elde ettiği ortalama mesafe değerlerine bakılırsa en iyi sonuç veren algoritma EYKS v2'dir. Bu tabloda algoritmaların ortalama çözüm sürelerine ve çözüm sürelerinin standart sapmalarına bakıldığında başta SÜP_A v2 ve SÜP_A-2 v2 hızlı sonuçlar üretmektedir. PSO hariç diğer algoritmaların ikinci versiyonları da ortalama çözüm süreleri en düşük değere çok yakındır. Yine bu tabloda senaryolarda ortalama kat edilen mesafe verilerine bakarak en iyi senaryonun, Senaryo-4 olduğu görülmektedir.

Tablo 16. Rastgele oluşturulmuş KTYP.B250.N171.T36.R135.3 adlı probleme ait senaryoların, geliştirilen algoritmalar ile çözümlenerek elde edilen sonuçlar

PROBLEM	SENARYO	N	Mesafe D		EYKS		BEYKS		SÜP_A		SÜP_A-2		PSO		ORTALAMA	STANDART SAPMA
			Süre T	v1	v2	v1	v2	v1	v2	v1	v2	v1	v2			
KTYP.B250.N171.T36.R135.3	S-1	135	D(cm)	19226		24444		91017		201308		205564	205622	124530	90847,7	
			T(sn)	1		2		1		0		132	89	38	58,2	
	S-2	171	D(cm)	46435	37296	129881	40055	165822	116225	260777	246194	256561	267282	156653	96323,7	
			T(sn)	210	9	152	3	171	2	169	2	2485	139	334	760,3	
	S-3	171	D(cm)	35484		38329		117018		245560		267072	267505	161828	111783,5	
			T(sn)	72		65		63		64		1938	138	390	758,9	
	S-4	136	D(cm)	22931	33472	30009	36709	94730	97237	198142	202580	205261	206542	112761	81828,1	
			T(sn)	127	7	54	3	48	1	51	1	1764	108	216	545,6	
	S-5	136	D(cm)	22651		24453		91656		195487		210444	208367	125510	90459,3	
			T(sn)	43		42		39		40		846	111	187	324,1	
	S-6	221	D(cm)	47644		156756		217709		357162		299284	288071	227771	112323,1	
			T(sn)	2		4		1		1		2553	2503	844	1304,5	
	S-7	137	D(cm)	22651		31952		100640		228419		221999	236869	140422	100914,5	
			T(sn)	0		2		0		0		138	189	55	85,7	
	S-8	136	D(cm)	22651		24453		91656		195487		205570	210605	125070	89983,3	
			T(sn)	0		2		0		0		137	255	66	107,6	
ORTALAMA D			29959	26786	57535	41906	121281	102573	235293	208022	233969	236358	146818	96808		
ORTALAMA T			56,88	16,75	40,38	15,38	40,38	13,38	40,63	13,50	1249,13	441,50	266,18	493,12		
Standart Sapma D			11585	9889,60	53644	44765,24	46449	42661,91	55238	54123	35870	33554,5				
Standart Sapma T			77	26,521	52	24,33068	59	24,1006	58	24,623	1059	834,66				

Rastgele üretilen KTYP.B250.N171.T36.R135.3 adlı problem üzerinden algoritmaların ve senaryoların karşılaştırılması Tablo 16’da verilmiştir. Senaryo-2’deki düğüm sayısı 171’dir. Bu problemde algoritmaların tüm senaryolarda elde ettiği ortalama mesafe değerlerine bakılırsa en iyi sonuç veren algoritma EYKS v2’dir. Bu tabloda algoritmaların ortalama çözüm sürelerine ve çözüm sürelerinin standart sapmalarına bakıldığında başta SÜP_A v2 ve SÜP_A-2 v2 hızlı sonuçlar üretmektedir. PSO hariç diğer algoritmaların ikinci versiyonları da ortalama çözüm süreleri en düşük değere çok yakındır. Yine bu tabloda senaryolarda ortalama kat edilen mesafe verilerine bakarak en iyi senaryonun, Senaryo-4 olduğu görülmektedir.

Tablo 17. Deneysel Problemlerde Senaryoların Ortalama Mesafe Değerlerini Gösteren Tablo

PROBLEM	Senaryo-2 N	S-1	S-2	S-3	S-4	S-5	S-6	S-7	S-8
KTYP.B4.N6.T2.R4.1	6	9470	11171	10744	10905	10203	38355	11360	10511
KTYP.B50.N13.T3.R10.3	13	14312	20473	18549	15801	14974	44093	30712	29685
KTYP.B20.N17.T7.R10.2	17	14020	23886	23451	17673	15593	51566	16490	15323
KTYP.B45.N25.T10.R15.3	25	15591	29129	27850	17897	16468	59770	21609	16715
KTYP.B30.N35.T20.R15.2	35	16791	37977	37923	19553	17052	61131	20648	16970
KTYP.B30.N50.T20.R30.2	50	31230	31230	50987	31663	30790	65924	31336	30590
KTYP.B45.N50.T20.R30.3	50	30544	52344	52032	31796	29874	78125	34396	30389
KTYP.B50.N76.T36.R40.2	76	37005	78725	74497	35929	37764	81309	39746	38676
KTYP.B150.N76.T35.R41.3	76	38179	77221	72079	37037	38876	129371	57894	45337
KTYP.B250.N171.T36.R135.3	171	124530	156653	161828	112761	125510	227771	140422	125070
ORTALAMA		33167	51881	52994	33102	33710	83741	40461	35927
STANDART SAPMA		33761	43340	43945	29473	33830	56701	37537	33233

Rastgele oluşturulan deneysel problemler, geliştirilen algoritmalar ile çözülmüş ve problemlerin her senaryosu için elde edilen ortalama toplam mesafe değerleri Tablo 17’de verilmiştir. Bu değerlerin de senaryo bazında ortalaması tekrar alınarak görevlilerin en az mesafe kat edebileceği senaryolar tespit edilebilir hale getirilmiştir. Tablo 17’deki veriler doğrultusunda ortalama en az mesafe kat edilen senaryo ise Senaryo-4’tür. Senaryo-4’ün hemen ardından ortalama en az mesafe kat edilen senaryolar ise sırası ile Senaryo-1 ve Senaryo-5 ‘tir. Bu iki senaryonun ortalama mesafe değerleri, Senaryo-4 ile oldukça yakındır.

Çalışma masalarının olduğu statik senaryolar Senaryo-2 ve Senaryo-3, toplama ve yerleştirme prosedürüne göre tek bir farkla sahiptir. Senaryo-2 EZTY kuralına göre toplama ve yerleştirme işlemi yaparken, Senaryo-3 ise ÖTSY kuralına göre toplama yerleştirme işlemi yapar. Aynı durum toplama masası olan statik Senaryo-4 ve Senaryo-5 için de geçerlidir. Senaryo-2 ve Senaryo-3 değerleri karşılaştırıldığı zaman EZTY kuralına göre işlem yapan Senaryo-2 nispeten Senaryo-3’e göre daha iyi sonuç vermiştir.

Toplama masasının olduğu dinamik olan Senaryo-7 ve Senaryo-8 verileri karşılaştırıldığında ise ÖTSY kuralına sahip Senaryo-8 daha iyi sonuçlar vermektedir.

Senaryo-7 ve Senaryo-8 dinamik senaryolar olmasına rağmen, Senaryo-2 ve Senaryo-3 gibi statik senaryolara nazaran daha iyi sonuçlar verdiği gözlemlenmiştir. Çünkü Senaryo-7 ve Senaryo-8 problem ne olursa olsun her zaman iki adet arz düğümü bulunmaktadır. Senaryo-2 ve Senaryo-3’e ait arz düğümleri kütüphane içerisinde bulunan çalışma masaları miktarına göre değişmektedir. Arz düğümlerindeki azalma öncelik ilişkileri karmaşıklığını ve ziyaret edilecek düğüm sayısını düşürmektedir. Özet olarak toplama masası olan senaryolar, çalışma masaları olan senaryolara göre daha iyi sonuçlar vermektedir.

Senaryo-8 ve Senaryo-5 biri R-PSO hariç tüm deneysel problemlerde aynı sonucu vermektedir. Bu durum, üretilen deneysel problemlerin tamamında kullanılmış kitapların olmasından kaynaklanır. Kütüphane içerisinde kullanılmış kitap olmadığı bir problemde görevli Senaryo-8’e göre danışma masasından hemen sonra toplama masasını ziyaret etmesi gerekmektedir. Senaryo-5’te ise görevli toplama masasında kitap olmadığını önceden bildiği için toplama masasını ziyaret etmemektedir. Herhangi bir problemde kullanılmış kitapların olmaması durumunda tüm statik senaryolar, Senaryo-1’e benzeyecek ve R-PSO hariç diğer algoritmalar aynı sonuçları verecektir.

Dinamik senaryolar ve statik senaryolar karşılaştırılmış ve statik senaryoların genel olarak daha iyi sonuçlar verdiği gözlemlenmiştir.

Daha önce de bahsedilen Senaryo-6'nın EYKS ile çözümü KTUNK kütüphanesinde uygulanan mevcut çözüme eşittir. Senaryo-6 diğer senaryolar ile karşılaştırıldığında diğer senaryoların daha iyi çözümlere sahip olduğu görülmektedir. Senaryo-6 kötü bir senaryo olması, kütüphanede bulunan çalışma masalarının kitap olmasa bile ziyaret edilmesinden kaynaklıdır. KTUNK'te çalışma masaları bütün kütüphaneye yayılmış durumda olduğu için tüm kütüphaneyi görevlinin gezmesi gerekir. Dolayısı ile katedilen toplam mesafe diğer senaryolara göre daha fazladır. Ayrıca herhangi bir çalışma masasında kitap toplandığında yeniden rotalama işlemleri yapılması da zaman kayıplarına yol açmaktadır.

Tablo 18. Geliştirilen Algoritmaların Deneysel Problemlerde Elde Ettikleri Ortalama Mesafe Değerleri

SENARYO	Senaryo-2 N	EYKS		BEYKS		SÜP_A		SÜP_A-2		PSO	
		v1	v2	v1	v2	v1	v2	v1	v2	v1	v2
KTYP.B4.N6.T2.R4.1	6	12343	10786	10846	9602	12817	11393	17971	15974	15113	15268
KTYP.B50.N13.T3.R10.3	13	15608	13873	17498	15554	20589	18301	35471	31530	29220	28197
KTYP.B20.N17.T7.R10.2	17	13404	12803	18990	16637	21103	18608	34115	29624	21854	23419
KTYP.B45.N25.T10.R15.3	25	16498	15021	21108	18477	24507	21555	35965	31567	28369	27606
KTYP.B30.N35.T20.R15.2	35	17287	15425	23768	20199	28660	23982	39889	34984	32538	31383
KTYP.B30.N50.T20.R30.2	50	21786	19480	29005	24224	39797	33370	61966	55550	53279	53955
KTYP.B45.N50.T20.R30.3	50	20483	18448	28111	23487	37779	32480	63430	55372	55654	54934
KTYP.B50.N76.T36.R40.2	76	22105	19649	33165	29480	44593	39638	76431	67939	73905	74040
KTYP.B150.N76.T35.R41.3	76	23818	22138	46180	34620	62158	50590	85801	75103	81233	84133
KTYP.B250.N171.T36.R135.3	171	29959	26786	57535	41906	121281	102573	235293	208022	233969	236358
ORTALAMA		19329	17441	28621	23419	41328	35249	68633	60566	62513	62929
STANDART SAPMA		5356	4806	14055	9650	31530	26344	62320	55121	64189	65044

Geliştirilen algoritmaların rastgele oluşturulmuş 10 deneysel problemde verdikleri ortalama toplam mesafe değerleri Tablo 18'de verilmiştir. Bu değerlerin ortalamaları ve standart sapmaları hesaplanmıştır. Bu ortalamalara bakarak EYKS ikinci versiyonu (v2) başta olmak üzere sırasıyla EYKS v1, BEYKS v2 ve BEYKS v1 diğer algoritmalarla göre iyi sonuçlar vermiştir. EYKS v1 ve EYKS v2'nin ortalama mesafe değerleri arasında t anlamlı bir fark görülmemiştir. Aynı şekilde EYKS v2 ve BEYKS v1 ortalama mesafe verileri arasında da anlamlı bir fark yoktur. Ayrıca Tablo 18'de geliştirilen sezgisel algoritmaların ikinci versiyonlarının birinci versiyonlarına göre daha iyi sonuçlar verdiği de görülmektedir.

Tablo 19. Geliştirilen Algoritmaların Ortalama Çözüm Süresi Değerleri

SENARYO	Senaryo-2 N	EYKS		BEYKS		SÜP_A		SÜP_A-2		PSO	
		v1	v2	v1	v2	v1	v2	v1	v2	v1	v2
KTYP.B4.N6.T2.R4.1	6	0	0	0	0	0	0	0	0	12	8
KTYP.B50.N13.T3.R10.3	13	2	2	2	2	2	2	2	2	49	49
KTYP.B20.N17.T7.R10.2	17	1	0	0	0	0	0	1	0	30	19
KTYP.B45.N25.T10.R15.3	25	2	1	2	2	2	1	2	1	66	45
KTYP.B30.N35.T20.R15.2	35	1	1	1	1	1	1	1	1	64	49
KTYP.B30.N50.T20.R30.2	50	3	1	3	1	2	1	2	1	90	81
KTYP.B45.N50.T20.R30.3	50	6	4	6	4	6	3	5	3	157	72
KTYP.B50.N76.T36.R40.2	76	3	3	3	3	3	3	3	3	146	196
KTYP.B150.N76.T35.R41.3	76	5	3	6	3	5	3	6	3	207	143
KTYP.B250.N171.T36.R135.3	171	57	17	40	15	40	13	41	14	1249	442
ORTALAMA		8	3	6	3	6	3	6	3	207	110
STANDART SAPMA		17	5	12	4	12	4	12	4	371	130

Rastgele üretilen 10 deneysel problemde algoritmaların ortalama çözüm süresi değerleri Tablo 19’da verilmiştir. Bu değerler doğrultusunda sezgisel algoritmalar, R-PSO’ya göre daha hızlı sonuçlar ürettiği görülmektedir. Geliştirilen sezgisel algoritmaların ikinci versiyonları, hem hız olarak hem de toplam rota mesafesi olarak birinci versiyonlarına göre daha iyidir.

5. SONUÇLAR VE ÖNERİLER

Kütüphanelerde iade edilen ya da çalışma masalarında bırakılan kitapların ait oldukları raflara yerleştirilmesi; önemli zaman kayıplarına yol açan ve iş verimliliğini doğrudan etkileyen önemli bir problemdir.

Bu tez kapsamında bir kütüphane içerisinde kitap toplama ve yerleştirme işlemlerinin verimliliğini artırabilmek için ilgili problem bir optimizasyon problemi olarak ele alınmış ve bu problemi etkin bir şekilde çözebilmek için bir karar destek sistemi geliştirilmiştir. Söz konusu optimizasyon problemi Kitap Toplama ve Yerleştirme problemi (KTYP) olarak isimlendirilmiştir. KTYP ile ilgili literatürde herhangi bir çalışma bulunmamakta, dolayısıyla ilk defa bu tezde tanımlanmaktadır.

KTYP’de amaç; kütüphane çalışanlarının iade edilen ya da kütüphane içerisinde çalışma masalarında bırakılan kitapları ait oldukları raflara en kısa rotayı yürüyerek yerleştirmeleridir. Bu problemin çözümü için bir karar destek sistemi (KDS) geliştirilmiş. KDS; kütüphanelerde hâlihazırda kullanılan yazılımların uygulayabildiği veri tabanı işlevlerini yapmasının yansısı KTYP problemi için tezde önerilen farklı optimizasyon algoritmalarını da içermektedir. Bu anlamıyla geliştirilen KDS yine literatürde benzeri bulunmayan bir programdır.

Ayrıca KTYP için oluşturulan statik senaryolarda çalışma masalarına barkod okuyucular konulmasını ve masaya getirilen kitapların bu okuyucular ile okutulup veri tabanına aktarılması önerilmektedir. Bu öneri; mevcut durum için bir yeni bir metot geliştirme çalışması olarak dikkate alınmalıdır.

Geliştirilen KDS; herhangi bir kitap toplama ve yerleştirme problem örneğini önerilen çözüm algoritmaları ile kısa sürede çözüp, karşılaştırmasını yapıp, en iyi çözümü (en iyi yürüyüş rotasını) belirleyebilecektir. Kütüphane görevlisi de bu rotaya göre hareket edebilecektir. Ayrıca kütüphanelerde görülebilecek farklı kitap toplama ve yerleştirme senaryolarının test edilebilmesi sayesinde mevcut duruma göre daha verimli kitap toplama ve yerleştirme stratejileri belirlenebilir.

5.1 Sonuçlar

KTYP için geliştirilen optimizasyon algoritmaları ile çözülen deneysel problemler gösteriyor ki bir kütüphanede kullanılan kitaplar için toplama masası olan senaryolar dinamik veya statik fark etmeksizin diğer senaryolara göre daha iyi sonuçlar vermektedir.

Kütüphane içerisinde arz noktalarının sayısını azaltmanın daha avantajlı olduğu gözlemlenmiştir.

EYKS'nin bir modifikasyonu olarak bu teze özgü geliştirilen BEYKS ise EYKS'ye göre kısmen daha kötü sonuçlar verse de tezde kullanılan diğer algoritmalara göre oldukça başarılı sonuçlar vermiştir.

Dinamik senaryolar otomatik şekilde aynı makine üzerinden çözüldüğü için el terminali ile veri okuma, verileri server makineye gönderme, sonuçların server makineden elde edildikten sonra geri el terminaline gönderilmesi gibi işlemler bu tezde geliştirilen karar destek sistemi programında yoktur. Dolayısı ile bu işlemler esnasında oluşacak zaman kayıpları ve sistem arızaları gibi unsurlar göz önüne alınamamıştır.

5.2 Öneriler

Kütüphanelerde kitapların bir noktada toplanmasını sağlamak iş yükünü oldukça azaltacaktır. Ayrıca kullanılacak olan barkod okuyucu, el terminali gibi teknolojik masrafları minimuma düşürerek daha tasarruflu bir sistem kurulabilir. Tezin önceki bölümlerinde bahsedildiği üzere Senaryo-1 klasik GSP olarak da ele alınabilmektedir. Dolayısıyla klasik GSP için geliştirilen çözüm yöntemleri Senaryo-1 için kullanılabilir.

Bu tezde geliştirilen karar destek sistemi daha verimli şekilde tasarlanarak kod bütünlüğü ve etkinliği açısından daha iyi hale getirilebilir. Karar destek sistemi tarafından çözülen problemlerin daha kısa sürede daha iyi sonuçlar verebilmesi de kütüphanelerde kullanılan teknolojik altyapıyla paraleldir. KTYP operasyonel düzeyde çözülmesi gereken optimizasyon problemleri içerdiğinden çözüm sürelerinin düşürülmesi için veri okuma ve yazma hızına bağlı olarak kullanılan teknolojik altyapının iyi ve güncel olması gerekmektedir. İlerleyen zamanlarda bu tezde geliştirilen karar destek sisteminin geliştirilmesi planlanmaktadır.

KTYP'de çözülen optimizasyon problemleri için kullanılan tekniklerin yanı sıra daha etkili yöntemler geliştirilmesi gerekebilir. İleriye dönük yayınlar için KTYP'de ortaya çıkan optimizasyon problemlerinin çözümü için Gambardella ve Dorigo (2000) tarafından geliştirilen ASP-3 değişimi yerel arama sezgiseli ile bütünleşmiş optimizasyon teknikleri geliştirmek planlanmıştır.

KAYNAKLAR

- Alves, R. D. P. (2018). Ricardo David Pereira Alves Vehicle Routing and Tour Planning Problem : A Cement Industry Case Study Universidade do Minho Ricardo David Pereira Alves Vehicle Routing and Tour Planning Problem : A Cement Industry Case Study.
- Anghinolfi, D., Montemanni, R., Paolucci, M. ve Maria Gambardella, L. (2011). A hybrid particle swarm optimization approach for the sequential ordering problem. *Computers and Operations Research*, 38(7), 1076–1085. doi:10.1016/j.cor.2010.10.014
- Asani, E. O., Okeyinka, A. E. ve Adebisi, A. A. (2020). A Construction Tour Technique for Solving the Travelling Salesman Problem Based on Convex Hull and Nearest Neighbour Heuristics. *2020 International Conference in Mathematics, Computer Engineering and Computer Science, ICMCECS 2020*, 24–27. doi:10.1109/ICMCECS47690.2020.240847
- Ascheuer, N., Escudero, L. F., Grötschel, M. ve Stoer, M. (1993). A Cutting Plane Approach to the Sequential Ordering Problem (with Applications to Job Scheduling in Manufacturing). *SIAM Journal on Optimization*, 3(1), 25–42. doi:10.1137/0803002
- Balas, E., Fischetti, M. ve Pulleyblank, W. R. (1995). The precedence-constrained asymmetric traveling salesman polytope. *Mathematical Programming*, 68(1), 241–265. doi:10.1007/BF01585767
- Bianco, L., Mingozzi, A., Ricciardelli, S. ve Spadoni, M. (1994). Exact And Heuristic Procedures For The Traveling Salesman Problem With Precedence Constraints, Based On Dynamic Programming. *INFOR: Information Systems and Operational Research*, 32(1), 19–32. doi:10.1080/03155986.1994.11732235
- Clerc, M. (1999). The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization. *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*, 3, 1951–1957. doi:10.1109/CEC.1999.785513
- Clerc, M. (2004). Discrete Particle Swarm Optimization, illustrated by the Traveling Salesman Problem, 219–239. doi:10.1007/978-3-540-39930-8_8
- Dewil, R., Küçükoğlu, İ., Luteyn, C. ve Cattrysse, D. (2019). A Critical Review of Multi-hole Drilling Path Optimization. *Archives of Computational Methods in Engineering*, 26(2), 449–459. doi:10.1007/s11831-018-9251-x
- Dorigo, M. ve Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1), 53–66. doi:10.1109/4235.585892
- Escudero, L. F. (1988). An inexact algorithm for the sequential ordering problem. *European Journal of Operational Research*, 37(2), 236–249. doi:10.1016/0377-2217(88)90333-5
- Escudero, L. F., Guignard, M. ve Malik, K. (1994). A Lagrangian relax-and-cut approach for the sequential ordering problem with precedence relationships. *Annals of Operations Research*, 50(1), 219–237. doi:10.1007/BF02085641
- Fagerholt, K. ve Christiansen, M. (2000). A travelling salesman problem with allocation, time window and precedence constraints — an application to ship scheduling. *International Transactions in Operational Research*, 7(3), 231–244. doi:10.1111/j.1475-3995.2000.tb00196.x
- Fonseca, C. M., Paquete, L. ve López-Ibáñez, M. (2006). An improved dimension-sweep algorithm for the hypervolume indicator. *2006 IEEE Congress on Evolutionary Computation, CEC 2006*, 1157–1163. doi:10.1109/cec.2006.1688440
- Francis, P., Smilowitz, K. ve Tzur, M. (2006). The period vehicle routing problem with

- service choice. *Transportation Science*, 40(4), 439–454. doi:10.1287/trsc.1050.0140
- Gambardella, L. M. ve Dorigo, M. (2000). An Ant Colony System Hybridized with a New Local Search for the Sequential Ordering Problem. *INFORMS Journal on Computing*, 12(3), 237–255. doi:10.1287/ijoc.12.3.237.12636
- Gillett, B. E. ve Miller, L. R. (1974). A Heuristie Algorithm for the Vehicle-Dispatch Problem. *Operations Research Publication*, (August 2015), 357–360. doi:doi.org/10.1287/opre.22.2.340
- Hamta, N., Fatemi Ghomi, S. M. T., Jolai, F. ve Akbarpour Shirazi, M. (2013). A hybrid PSO algorithm for a multi-objective assembly line balancing problem with flexible operation times, sequence-dependent setup times and learning effect. *International Journal of Production Economics*, 141(1), 99–111. doi:10.1016/j.ijpe.2012.03.013
- Heppner, F. H. ve Grenander, U. (1990). A Stochastic Non-Linear Model for Bird Flocking. *The Ubiquity of Chaos*, (July), 233–238.
- Hernández, P. H. ve Salazar, G. J. J. (2009). The multi-commodity one-to-one pickup-and-delivery traveling salesman problem. *European Journal of Operational Research*, 196(3), 987–995. doi:10.1016/j.ejor.2008.05.009
- Ho, S. C., Szeto, W. Y., Kuo, Y. H., Leung, J. M. Y., Petering, M. ve Tou, T. W. H. (2018). A survey of dial-a-ride problems: Literature review and recent developments. *Transportation Research Part B: Methodological*, 111, 395–421. doi:10.1016/j.trb.2018.02.001
- Kennedy, J. ve Eberhart., R. (1995). Particle Swarm Optimisation. *Proceedings of ICNN'95-international conference on neural networks*, 4, 1942–1948. doi:10.1007/978-3-030-61111-8_2
- Khan, A. A. ve Agrawal, M. H. (2016). a Comparitive Study of Nearest Neighbour Algorithm and Genetic Algorithm in Solving Travelling Salesman Problem. *International Research Journal of Engineering and Technology*. www.irjet.net adresinden erişildi.
- Kizilateş, G. ve Nuriyeva, F. (2013). On the Nearest Neighbor Algorithms for the Traveling Salesman Problem. *Advances in Intelligent Systems and Computing*, 225(1), 111–118. doi:10.1007/978-3-319-00951-3_11
- Kusiak, A. ve Finke, G. (1987). Modeling and solving the flexible forging module scheduling problem. *Engineering Optimization*, 12(1), 1–12. doi:10.1080/03052158708941080
- Letchford, A. N. ve Salazar, G. J. J. (2016). Stronger multi-commodity flow formulations of the (capacitated) sequential ordering problem. *European Journal of Operational Research*, 251(1), 74–84. doi:10.1016/j.ejor.2015.11.001
- Lin, S. C. ve Hsueh, H. C. (1994). Nearest-neighbour heuristics in accelerated algorithms of optimisation problems. *Physica A: Statistical Mechanics and its Applications*, 203(3–4), 369–380. doi:10.1016/0378-4371(94)90005-1
- McKenney, M., Frye, R., Dellamano, M., Anderson, K. ve Harris, J. (2017). Multi-core parallelism for plane sweep algorithms as a foundation for GIS operations. *GeoInformatica*, 21(1), 151–174. doi:10.1007/s10707-016-0277-7
- McKenney, M. ve McGuire, T. (2009). A parallel plane sweep algorithm for multi-core systems. *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, 392–395. doi:10.1145/1653771.1653827
- Min, H. (1989). The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A: General*, 23(5), 377–386. doi:10.1016/0191-2607(89)90085-X
- Mingozzi, A., Bianco, L. ve Ricciardelli, S. (1997). Dynamic programming strategies for the traveling salesman problem with time window and precedence constraints.

- Operations Research*, 45(3), 365–377. doi:10.1287/opre.45.3.365
- Moon, C., Kim, J., Choi, G. ve Seo, Y. (2002). An efficient genetic algorithm for the traveling salesman problem with precedence constraints. *European Journal of Operational Research*, 140(3), 606–617. doi:10.1016/S0377-2217(01)00227-2
- Nurcahyo, G. W., Shamsuddin, S. M., Alias, R. A. ve Noor, M. M. S. (2003). Selection of Defuzzification Method To Obtain Crisp Values for Representing Uncertain Data in a Modified Sweep Algorithm. *Elektron : Jurnal Ilmiah*, 5(2), 13–28. doi:10.30630/eji.5.2.50
- Psaraftis, H. N. (1980). Dynamic Programming Solution To the Single Vehicle Many-To-Many Immediate Request Dial-a-Ride Problem. *Transportation Science*, 14(2), 130–154. doi:10.1287/trsc.14.2.130
- Reynolds, C. W. (1987). Flocks, herds, and schools: A distributed behavioral model. *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1987*, 21(4), 25–34. doi:10.1145/37401.37406
- Sherali, H. D. ve Driscoll, P. J. (2002). ON TIGHTENING THE RELAXATIONS OF MILLER-TUCKER-ZEMLIN FORMULATIONS FOR ASYMMETRIC TRAVELING SALESMAN PROBLEMS. *Operations Research*, 50(4), 656–669. doi:10.1287/opre.50.4.656.2865
- Shi, Y. ve Eberhart, R. (1998). A Modified Particle Swarm Optimizer. *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)* içinde (ss. 69–73). doi:10.1021/ci800374h
- Shobaki, G. ve Jamal, J. (2015). An exact algorithm for the sequential ordering problem and its application to switching energy minimization in compilers. *Computational Optimization and Applications*, 61(2), 343–372. doi:10.1007/s10589-015-9725-9
- Skinderowicz, R. (2015). Population-Based Ant Colony Optimization, 99–109. doi:10.1007/978-3-319-24306-1
- Stein, D. M. (1978). Scheduling Dial-a-Ride Transportation Systems. *Transp Sci*, 12(3), 232–249. doi:10.1287/trsc.12.3.232
- Tuna, G., Zogo, R. ve Demirelli, B. (2013). An Introduction to Digitization Projects Conducted by Public Libraries : Digitization and Optimization Techniques. *Journal of Balkan Libraries Union*, 1(1), 28–30. <http://bkbd.trakya.edu.tr> adresinden erişildi.
- Wang, D., Tan, D. ve Liu, L. (2018). Particle swarm optimization algorithm: an overview. *Soft Computing*, 22(2), 387–408. doi:10.1007/s00500-016-2474-6
- Wang, K.-P., Huang, L., Zhou, C.-G. ve Pang, W. (2003). Particle swarm optimization for Traveling Salesman Problem. *Proceedings - 27th European Conference on Modelling and Simulation, ECMS 2013* içinde . doi:10.7148/2013-0857
- While, L., Hingston, P., Barone, L. ve Huband, S. (2006). A faster algorithm for calculating hypervolume. *IEEE Transactions on Evolutionary Computation*, 10(1), 29–38. doi:10.1109/TEVC.2005.851275
- Yousefikhoshbakht, M. ve Sedighpour, M. (2012). A combination of sweep algorithm and elite ant colony optimization for solving the multiple traveling salesman problem. *Proceedings of the Romanian Academy Series A - Mathematics Physics Technical Sciences Information Science*, 13(4), 295–301.
- Yükseköğretim Bilgi Yönetim Sistemi, 2022, <https://istatistik.yok.gov.tr/> [Ziyaret Tarihi: 4 Kasım 2022].
- Zhao, Y. W., Wu, B., Wang, W. L., Ma, Y. L., Wang, W. A. ve Sun, H. (2004). Particle Swarm Optimization for Vehicle Routing Problem with Time Windows. *Materials Science Forum*, 471–472, 801–805. doi:10.4028/www.scientific.net/msf.471-472.801

Zhu, Q., Qian, L., Li, Y. ve Zhu, S. (2006). An improved particle swarm optimization algorithm for vehicle routing problem with simultaneous pickup and delivery. *Communications in Computer and Information Science* içinde . doi:10.1007/978-3-642-16336-4_57

