



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

**AFET ÖNCESİ ZARAR AZALTMA FAALİYETLERİNİN İNSANİ YARDIM
LOJİSTİĞİNDEKİ ETKİSİNİN EN İYİLEME KULLANILARAK
İNCELENMESİ**

Aslıhan Fatma KULA

YÜKSEK LİSANS TEZİ

Endüstri Mühendisliği Anabilim Dalı

Haziran-2022
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Aslıhan Fatma KULA tarafından hazırlanan “Afet Öncesi Zarar Azaltma Faaliyetlerinin İnsani Yardım Lojistiğindeki Etkisinin En İyileme Kullanılarak İncelenmesi” adlı tez çalışması 30/06/2022 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. İsmail KARAOĞLAN

.....

Danışman

Dr. Öğr. Üyesi Alper DÖYEN

.....

Üye

Doç. Dr. Muhammed Emre KESKİN

.....

Yukarıdaki sonucu onaylıyorum.

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.

Aslıhan Fatma KULA

Tarih: 03.06.2022

ÖZET

YÜKSEK LİSANS TEZİ

AFET ÖNCESİ ZARAR AZALTMA FAALİYETLERİNİN İNSANİ YARDIM LOJİSTİĞİNDEKİ ETKİSİNİN EN İYİLEME KULLANILARAK İNCELENMESİ

Aslıhan Fatma KULA

Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Endüstri Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Alper DÖYEN
İkinci Danışman: Prof. Dr. Yasemin ARDA

2022, 87 Sayfa

Jüri

Dr. Öğr. Üyesi Alper DÖYEN
Prof. Dr. İsmail KARAOĞLAN
Doç. Dr. Muhammed Emre KESKİN

Dünya nüfusunun artması, yeryüzünde insanların kullanımına açılan alanların çoğalması, kentleşme ve küresel ısınmanın etkileriyle birlikte her geçen yıl meydana gelen doğal afetler de artmaktadır. Doğal afetler, çok büyük insan kayıplarına neden olabileceği gibi ekonomik etkiler açısından da yıkıcı olabilmektedirler. Doğal afetlerin ne zaman ve hangi şiddette gerçekleşeceği önceden bilinemediğinden ve engellenemediğinden, gerçekleştiğinde çok büyük hasarların, can ve mal kayıplarının önüne geçebilmek için afet gerçekleşmeden ve gerçekleştikten sonra etkin afet yönetimi çalışmaları yapılmalıdır.

Etkin bir afet yönetimi çalışması, afet öncesi, afet esnası ve afet sonrasında tüm gereksinimleri kapsayacak şekilde olmalıdır. Afet öncesinde bina ve bağlantı yollarının güçlendirilmesi, afet olduktan hemen sonra afetzedelere insani yardım malzemelerinin iletilmesi, daha sonrasında ise hasar görmüş bina ve bağlantı yollarının iyileştirilmesi gerekmektedir. Literatürde, bu bahsedilen afet yönetimi kararlarının tamamını aynı anda içeren bir model bulunmamaktadır.

Bu tez çalışması kapsamında, afet öncesi ve sonrası tüm kararların birbirleriyle etkileşimlerinin bir bütün olarak dikkate alındığı iki kademeli tam sayılı stokastik programlama modeli geliştirilmiştir. Geliştirilen modelin ilk kademesinde deprem gerçekleştiğinde olası can, mal ve ekonomik kayıpları azaltmak amacıyla afetten önce güçlendirilecek bina ve bağlantı yollarının seçimine ve güçlendirme seviyesine karar verilmektedir. Modelin ikinci kademesinde ise deprem sonrasında oluşan insani yardım malzemesi talebine ait dağıtım kararları ile yıkılan bina ve bağlantı yollarını onarma kararları verilmektedir. Modelde depremin farklı zaman periyotlarında farklı olasılıklar ile gerçekleşebilme durumu da dikkate alınmaktadır.

Matematiksel modelin çözümü için iki kademeli tam sayılı stokastik programlama modellerinin çözümü için geliştirilmiş ayrıştırma yöntemlerinden biri olan Tam Sayılı L-Şekilli (Integer L-Shaped) yöntemi kullanılmıştır. Geliştirilen test problemleri iki saatlik çözüm süresi içerisinde hem önerilen Tam Sayılı L-Şekilli yöntemle hem de ticari bir çözücü olan CPLEX ile çözümlenerek sonuçlar karşılaştırılmış ve önerilen yöntemin etkinliği gösterilmiştir. CPLEX'e alınan toplam 840 problem örneğinin 228 tanesi için (problemlerin %28'i) uygun bir çözüm elde edebilirken, önerilen yöntem 600 problem için (problemlerin %71'i) uygun bir çözüm elde edebilmiştir. Özellikle, görece büyük boyutlu problemlerin çözümünde önerilen yöntem CPLEX'e göre çok ciddi avantaj sağlamaktadır.

Anahtar Kelimeler: Afet Yönetimi, İki Kademeli Stokastik Programlama, İnsani Yardım Lojistiği, Tam Sayılı L-Şekilli Yöntem

ABSTRACT

MS THESIS

USING OPTIMIZATION TO ANALYZE THE EFFECT OF MITIGATION DECISIONS IN HUMANITARIAN RELIEF LOGISTICS

Aslıhan Fatma KULA

Konya Technical University
Institute of Graduate Studies
Department of Industrial Engineering

Advisor: Asst. Prof. Dr. Alper DÖYEN
Secondary Advisor: Prof. Dr. Yasemin ARDA

2022, 87 Pages

Jury

Asst. Prof. Dr. Alper DÖYEN
Prof. Dr. İsmail KARAOĞLAN
Assoc. Prof. Dr. Muhammed Emre KESKİN

Natural disasters are gradually increasing due to population growth, expansion in the areas open to human usage on earth, urbanization and the effects of global warming. Natural disasters can result in massive human casualties as well as devastating economic consequences. Since natural disasters cannot be predicted when or how severe they will be, and because they cannot be prevented; effective disaster management studies should be conducted in order to prevent serious damage, civilian and property losses before and after they occur.

An effective disaster management study should cover all the requirements before, during and after the disaster. In advance of the disaster occurrence buildings and transportation networks must be strengthened, immediately after the disaster occurrence humanitarian aid must be delivered to catastrophe victims and much later the disaster occurrence damaged buildings and transportation network should be recovered. However, in the literature, there is no such a model that includes all of these disaster management decisions together.

Within the scope of this thesis, a two-stage stochastic integer programming model has been developed which takes into consideration the interactions of all pre-disaster and post-disaster decisions as a whole. In the first stage of the developed model, in order to reduce possible life, property and economic losses when an earthquake occurs, the selection of buildings and transportation roads to be strengthened before the disaster and the level of reinforcement are decided. In the second stage of the model, distribution decisions regarding the humanitarian aid material demand after the earthquake and decisions to repair the destroyed buildings and transportation networks are made. In the model, the possibility of the earthquake to occur with different probabilities in different time periods is also taken into account.

The Integer L-Shaped algorithm, which is one of the decomposition strategies that allows us to obtain effective results in the solution of two-stage stochastic programming models, is implemented to solve the given mathematical model. Developed test problems are solved in two hours by using both the proposed Integer L-Shaped method and the commercial solver CPLEX. The results are compared and the efficiency of the proposed methodology is shown. CPLEX was able to obtain a feasible solution for 228 (28% of problems) from the total 840 problem examples, while the proposed method obtained a feasible solution for 600 problem (71% of problems). In particular, the proposed method significantly outperforms CPLEX for tackling reasonably large problems.

Keywords: Disaster Management, Humanitarian Logistics, Integer L-Shaped, Two-stage Stochastic Programming

ÖNSÖZ

Dünya ve ülke topraklarının, başta deprem olmak üzere sel, fırtına, heyelan, çığ düşmesi gibi çeşitli afet risklerine sahip olması, bu topraklar üzerinde önemli sanayi kuruluşlarının bulunması ve nüfusun büyük bir kısmının yaşaması tehlikenin boyutunu ve afet yönetimi çalışmalarının önemini artırmaktadır. Her yıl giderek artan can ve mal kayıplarının yanında, milyarlarca dolar ekonomik kayıp meydana gelmektedir.

Ülkemizin deprem kuşağında bulunması sebebiyle günümüze kadar deprem kaynaklı can ve mal kayıpları da çok yüksek olmuştur. Depremlerden sonra özellikle konutların, işyerlerinin, okulların, hastanelerin vb. binaların zarar görmesi veya yıkılması çok büyük sosyal ve ekonomik kayıplara neden olmaktadır. Binaların yanı sıra depremden sonra insani yardım malzemelerinin afetzedelere ulaştırılması konusunda toplum açısından büyük önem taşıyan yollar, demiryolları, iletişim ve ulaşım ağları da zarar görebilmektedir. Bütün bunlar depremin şiddetini etkilemektedir.

Dolayısıyla, olası can ve olası mal kayıplarının azaltılabilmesi için deprem gerçekleşmeden önce depremin zararlarını, deprem gerçekleştikten sonra da depremin etkilerini azaltmaya yönelik çalışmalar yapılmalıdır.

Bu tez çalışmasının amacı da karar vericilere deprem öncesinde, anında ve sonrasında almaları gereken stratejik kararları vermelerine ve daha etkin afet yönetimi planları yapmalarına yardımcı olacak bir model geliştirmektir.

Yüksek lisans sürecimde engin bilgilerini benimle paylaşan, çalışmamızın her aşamasında bana kıymetli zamanını ayırıp önerileri ile beni yönlendiren, her sorun yaşadığımda yanına çekinmeden gidebildiğim, güler yüzünü ve samimiyetini benden esirgemeyen ve gelecekteki mesleki hayatımda da değerli bilgilerinden faydalanmak istediğim kıymetli danışmanım Sayın Dr. Öğr. Üyesi Alper DÖYEN' e, bilgi ve deneyimlerini bizimle paylaşarak tez çalışmamızda destek olan ikinci danışmanım Sayın Prof. Dr. Yasemin ARDA' ya ve son olarak tüm eğitim hayatım boyunca benden desteklerini esirgemeyen ve her daim yanımda olan sevgili aileme sonsuz teşekkürlerimi sunarım.

Aslıhan Fatma KULA
KONYA, 2022

İÇİNDEKİLER

Sayfa Nu

ÖZET	iii
ABSTRACT.....	iv
ÖNSÖZ	v
İÇİNDEKİLER	vi
1. GİRİŞ	1
2. KAYNAK ARAŞTIRMASI	5
2.1. Müdahale Aşamalarını Dikkate Alan Çalışmalar	7
2.2. Hafifletme ve Müdahale Aşamalarını Dikkate Alan Çalışmalar	17
2.3. Hafifletme ve İyileştirilmesi Aşamalarını Dikkate Alan Çalışmalar	19
2.4. Bütünleşik Afet Yönetimi Çalışmaları.....	21
2.5. L-Şekilli ve Tam Sayılı L-Şekilli Ayrıştırma Yöntemlerini Kullanan Çalışmalar	23
3. MATERYAL VE YÖNTEM.....	24
3.1. İki Kademeli Stokastik Programlama	24
3.2. Geliştirilen Afet Yönetimi Modeli	27
3.3. L-Şekilli Yöntemi.....	36
3.4. Önerilen Modelin Tam Sayılı L-Şekilli Yöntemi ile Çözülmesi	38
3.4.1. Optimalite kısıtının oluşturulması	39
4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA.....	41
4.1. Deney Tasarımı	41
4.2. Deney Sonuçları	42
4.3. Mükemmel Bilginin Beklenen Değeri ve Stokastik Çözümün Değeri	55
5. SONUÇLAR VE ÖNERİLER	58
KAYNAKLAR	60
EKLER	64
EK-1 C# Dilinde Yazılan Stokastik Programlama Modelinin Kodları.....	64

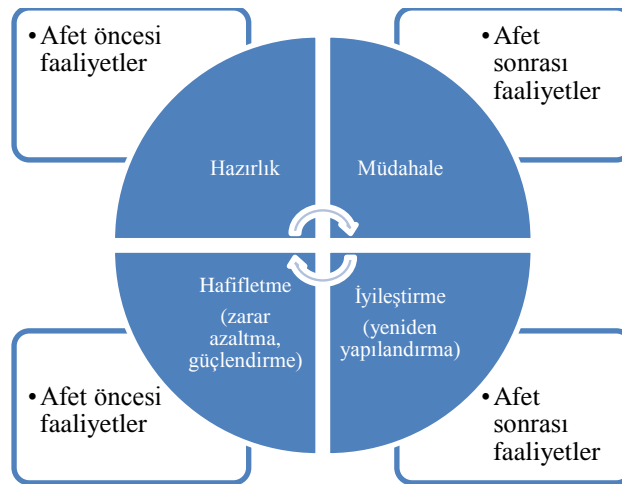
1. GİRİŞ

Afetler, insan yaşamını doğrudan tehdit eden, ekonomik ve sosyal kayıplara neden olan doğal, teknolojik veya insan kökenli olaylardır. Doğal afetler ise dünya nüfusunun artması, kentleşme ve küresel ısınmanın da etkileriyle birlikte tamamen doğal nedenlerle ortaya çıkan felaketlerdir.

Doğal afetler, insan kayıpları, çevre ve ekonomik etkiler açısından yıkıcı olabilmektedir. EM-DAT' tan (Acil Durum Veritabanı) elde edilen verilere göre, son 30 yılda dünya genelinde yaklaşık olarak 17 bin doğal afet ve bu doğal afetler nedeniyle toplamda 3 milyondan fazla ölüm meydana gelmiştir. 5,9 milyar insan etkilenmiş ve bu doğal afetler yaklaşık olarak 2,5 milyar dolar ekonomik hasara neden olmuştur (Moreno, 2015). Ayrıca veriler önümüzdeki yıllarda meydana gelen doğal afet sayısının dünyada ve ülkemizde artmaya devam edeceğini göstermektedir.

Bu küresel artış karşısında, dünya çapında hükümetler ve sivil toplum kuruluşları, özellikle afet riski bulunan bölgelerdeki afetlerin olumsuz etkilerini en aza indirmek ve kesintiye uğrayan hayatın bir an önce normale döndürülmesi amacıyla afet öncesi ve sonrası için ihtiyaç duyulan stratejileri geliştirmeli ve detaylı planlamalar yapmaları gerekmektedir. Bu strateji ve planlama faaliyetlerinin tümüne genel olarak afet yönetimi denir.

Afet yönetiminde temel olarak dört aşama bulunmaktadır: hazırlık, müdahale, hafifletme ve iyileştirme.



Şekil 1.1. Afet yönetimi aşamaları (Döyen, 2018)

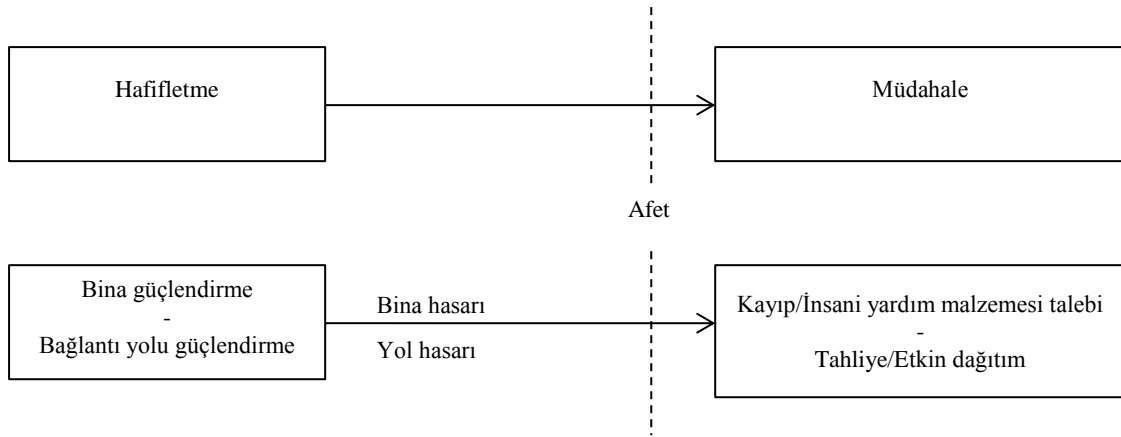
Hazırlık aşamasında, meydana gelebilecek bir afete karşı önlemler almak için çeşitli afet yönetimi politikaları ve stratejileri geliştirilmektedir. Hazırlık çalışmaları kapsamında örneğin, yardım (kurtarma) merkezlerinin konumları, sayıları, kapasiteleri ve stoklanması gereken insani yardım malzemesi miktarı, bağlantı yollarının ve araçların kapasiteleri, kullanılacak bağlantı yollarının yol bilgileri ve bütçe belirlenebilmektedir.

Müdahale aşamasında, yardım merkezlerinden yardım personeli ve/veya insani yardım malzemeleri afet bölgelerine ulaştırılmaktadır ya da afetzedeler afet bölgelerinden tahliye edilmektedir. Gönderilen yardım personeli ve/veya insani yardım malzemelerinin afet bölgelerine en kısa zamanda ve minimum maliyetle ulaştırılabilmesi amacıyla da müdahale aşamasında en kısa ve en uygun güzergâhlar belirlenmektedir.

Hafifletme (zarar azaltma) aşaması, doğal afetlerin etkilerini azaltmak amacıyla gerçekleştirilen bina (yapı) ve/veya bağlantı yolu güçlendirme kararlarının alındığı aşamadır. Hafifletme faaliyetlerinin zamanında yapılmamış olması, doğal afetler meydana geldiğinde hem can ve mal kayıplarının artmasına hem de afet sonrasında uzun vadede afet bölgelerinin yeniden inşası için yapılacak harcamaların artmasına neden olmaktadır.

İyileştirme (yeniden yapılandırma/onarma) aşamasında ise afette hasar görmüş binalar ve/veya bağlantı yolları onarılmaktadır veya yeniden inşa edilmektedir.

Yukarıdaki açıklamalardan da anlaşılacağı üzere, afet öncesi ve afet sonrası arasında önemli bir ilişki vardır. Afet gerçekleşmeden önce yapılan hazırlık ve hafifletme faaliyetleri, afetten sonra kayıpların azaltılmasında önemli bir rol oynamaktadır. Şekil 1.2’de görüldüğü gibi, yeterince güçlendirilmemiş binaların afette hasar görmesi veya yıkılması, yaralanan insan sayısının ve insan kayıplarının artmasına ve buna bağlı olarak da insani yardım malzemesi talebinin artmasına sebep olmaktadır. Yeterince güçlendirilmemiş bağlantı yollarının afette hasar görmesi veya yıkılması ise afetzedelerin afet bölgelerinden tahliyelerinin gecikmesine ya da insani yardım malzemelerinin etkin bir şekilde dağıtılamamasına sebep olmaktadır. Ayrıca hasar görmüş binaların ve/veya bağlantı yollarının onarılması veya yeniden inşası, iyileştirme aşamasında maliyetlerin çok fazla artmasına sebep olmaktadır.



Şekil 1.2. Afet öncesi ve sonrası arasındaki ilişki

Ancak afet yönetimi çalışmaları çoğunlukla binaların veya bağlantı yollarının güçlendirilmesi, afetzedelerin afet bölgelerinde tahliye edilmesi ya da yardım merkezlerinden afet bölgelerine insani yardım malzemelerinin ulaştırılması problemlerinden yalnızca birini ele almaktadır. Afet öncesinde binaların veya bağlantı yollarının güçlendirilmesi ile afet sonrasında iyileştirme çabalarına odaklanan planlama modellerinde, insani yardım malzemelerinin dağıtımı veya afetzedelerin etkin şekilde tahliye edilmesi göz ardı edilmektedir. Tersi şekilde; afetzedelerin tahliyesine ya da yardım malzemelerinin dağıtımına odaklanan problemlerde ise zarar azaltma ve iyileştirme faaliyetleri dikkate alınmamaktadır. Dolayısıyla bu çalışmalarda problem tek yönlü olarak ele alınmış olmaktadır.

Yapılan çalışmalar arasında, birden fazla aşamayı dikkate alan çalışmalar da bulunmaktadır. Örneğin hafifletme ve müdahale aşamalarının birlikte ele alındığı çalışmalar bulunmaktadır. Bu çalışmalarda, afet gerçekleşmeden önce bağlantı yollarının güçlendirildiği, afet gerçekleştikten sonra da afet bölgelerine insani yardım malzemelerinin dağıtıldığı veya afetzedelerin afet bölgelerinden tahliye edildiği problem ele alınmaktadır. Ancak bu çalışmalarda da, normal hayata en kısa sürede dönülmesi amacıyla yapılması gereken faaliyetler ve afet sonrasında hasar gören bağlantı yollarının yeniden inşası için yapılacak harcamalar göz önünde bulundurulmamaktadır. Ayrıca bu çalışmalarda binaların da yıkılabileceği durum göz ardı edilmektedir. Bu kapsamda yapılan çalışmalar Bölüm 2.2’te verilmiştir.

Hafifletme ve iyileştirme aşamalarının birlikte ele alındığı çalışmalar da bulunmaktadır. Hafifletme ve iyileştirme aşamalarının birlikte değerlendirildiği çalışmalarda alınan güçlendirme kararları genellikle yalnızca binaların

güçlendirilmesini ve iyileştirilmesini ele almaktadır. Bu çalışmalarda bağlantı yollarının da yıkılabileceği durum göz ardı edilmektedir ve afet bölgelerine insani yardım malzemelerinin dağıtımı veya afetzedelerin afet bölgelerinde tahliye edilmesi problemleri dikkate alınmamaktadır. Bu da ayrılan bütçenin yalnızca binaların güçlendirilmesi veya yeniden inşası için harcanmasına neden olmaktadır. Bu kapsamda yapılan çalışmalar bölüm 2.3'te verilmiştir.

Birden fazla aşamayı dikkate alan çalışmalar arasında hafifletme, müdahale ve iyileştirme aşamalarının birlikte ele alındığı çalışmalar da bulunmaktadır. Bu çalışmalarda, afet gerçekleşmeden bağlantı yollarının güçlendirilmesi, afet gerçekleştikten sonra insani yardım malzemelerinin dağıtımı veya afetzedelerin afet bölgelerinden tahliyesi ve hasar gören bağlantı yollarının yeniden inşası için yapılacak harcamalar dikkate alınmaktadır. Bu çalışmalarda binaların da yıkılabileceği durum göz ardı edilmektedir. Bu kapsamda yapılan çalışmalar bölüm 2.4'te verilmiştir.

Bu tez çalışması kapsamında ise afet öncesi ve sonrası kararların birbirleri arasındaki etkileşimlerini dikkate alan iki kademeli tam sayılı stokastik programlama modeli geliştirilmiştir. Önerilen modelde hafifletme, müdahale ve iyileştirme aşamaları bir bütün olarak ele alınmıştır. Hafifletme aşamasında, afet gerçekleşmeden önce, can ve mal kayıplarını azaltmak amacıyla hem binalara hem de bağlantı yollarına güçlendirme faaliyetleri uygulanarak afetin etkilerinin en aza indirilmesi hedeflenmiştir. Müdahale aşamasında, insani yardım malzemelerinin yardım merkezlerinden afet bölgelerine hangi bağlantı yollarının kullanılarak ulaştırılacağına karar verilerek karşılanamayan talep miktarlarını en aza indirmek amaçlanmıştır. İyileştirme aşamasında ise afette hasar görmüş binalar ve bağlantı yolları onarılmakta veya yeniden inşa edilmektedir. Sonuç olarak geliştirilen modelin temel amacı, belirli güçlendirme bütçeleri altında toplam yeniden inşa ve müdahale maliyetlerinin en küçüklenmesidir.

Bu çalışma ile karar vericilerin daha doğru kararlar verip daha etkin afet yönetimi planlamaları yapmalarına yardımcı olmak ve bu sayede de afetlerin insanlar ve ekonomi üzerindeki olumsuz etkilerinin en aza indirilmesi amaçlanmıştır.

2. KAYNAK ARAŞTIRMASI

Afet yönetimi çalışmalarının çoğunluğu hazırlık aşamasına aittir. Hazırlık aşaması, afette ortaya çıkabilecek kayıpların azaltılması ve acil yardım ihtiyaçlarının karşılanması amacıyla afet gerçekleşmeden arama-kurtarma faaliyetlerinin geliştirilmesi ile ilgilenmektedir (Şahin, 2009). Afet öncesi çalışmaları; planlama, satın alma, taşımacılık, depo yönetimi, raporlama ve insan kaynaklarının geliştirilmesi faaliyetlerini düzenlemektedir (Tanyaş, 2013a). Hazırlık aşaması aynı zamanda sağlık, barınma, ihtiyaçlar için stokların oluşturulmasını, lojistik ve yardım merkezlerinin kurulmasını da kapsamaktadır.

Literatürde afet yönetimi ile ilgili geliştirilen matematiksel modeller de hazırlık aşamasına aittir. Afet yönetimi problemlerinin çözümünde sıklıkla iki kademeli stokastik programlama yöntemi kullanılmaktadır. Afet öncesinde alınan planlama kararları stratejik kararlar iken, afet sonrasında alınan düzeltme kararları operasyonel kararlardır. İki kademeli stokastik programlama yönteminin, özellikle afet öncesi hazırlıklara uygun olması sebebiyle yapılan çalışmaların büyük bir çoğunluğu afet öncesi aşamasına aittir.

Afet yönetimi ile ilgili yapılan kaynak araştırmaları da hazırlık faaliyetlerindedir. Aşağıda sunulan çalışmalar afet yönetimi ile ilgili detaylı incelemeler yapmışlardır.

Örneğin Caunhye ve ark. (2012) acil durum lojistiğinde kullanılan optimizasyon modellerini incelemiştir. Acil durum lojistiğinde operasyonlar, afet meydana gelmeden önce veya sonra yapılabilir. Kısa süreli tahliye, tesis yerleşimi ve stokların hazırlanması afet öncesi, insani yardım malzemelerinin dağıtımı ve afetzedelerin yardım merkezlerine taşınması afet sonrası operasyonlar olarak sınıflandırılmaktadır. Caunhye ve ark. (2012) ise bu operasyonlara göre literatürdeki çalışmaları üç farklı bölüme ayırmış ve analiz etmiştir. Bu üç bölüm; tesis yerleşimi, insani yardım malzemelerinin dağıtımı ve afetzedelerin yardım merkezlerine taşınmasıdır.

Faturechi ve Miller-Hooks (2014) ise ulaştırma sistemlerinin afet yönetiminde performansının değerlendirilmesi ile ilgili ayrıntılı bir literatür incelemesi sunmaktadır. Yaklaşık 200 makale farklı kriterlere göre sınıflandırılmıştır.

Grass ve Fischer (2016) ve Hoyos ve ark. (2015), çalışmalarında afet yönetiminde iki aşamalı stokastik programlama ile ilgili bir literatür incelemesi sunmaktadır.

Gupta ve ark. (2016) ise, afet yönetimi ile ilgili çalışmalarını beş farklı kategoride sınıflandırmış ve incelemiştir. Afet yönetiminin işlevi bakımından (önleme ve hafifletme, tahliye, insani yardım lojistiği, afet sonrası iyileştirme, vb.); afetin gerçekleşme zamanı bakımından (afetten önce, afet sırasında ve afetten sonra); afetin tipi bakımından (kaza, deprem, sel, kasırga, toprak kayması, patlama, vb.); veri tipi bakımından (gerçek veriler, deneysel veriler) ve veri analiz tekniği bakımından incelenmiştir (matematiksel modelleme, hedef programlama, sezgisel, simülasyon, vb.).

Behl ve Dutta (2018)'ya göre literatürde tedarik zinciri yönetimi kapsamlı bir şekilde incelenirken, tedarik zinciri yönetiminin afet yardımı operasyonlarındaki rolüne yeterince önem verilmemiştir. Ancak son yıllarda insani yardım lojistiği ve tedarik zinciri yönetimine odaklanan çalışmaların, İnsani Operasyonlar ve Tedarik Zinciri Yönetimi (HOSCM - Humanitarian Operations and Supply Chain Management) dergilerinde popülaritesinin arttığı gözlemlenmiştir. Bu çalışmada ise, 2011 ve 2017 yılları arasında operasyonlar ve tedarik zinciri yönetimi dergilerinin yanı sıra popüler disiplinler arası dergilerde yayınlanan 362 makalenin kapsamlı bir incelemesi yapılmaktadır. Bu makaleler insani yardım lojistiği, teorik araştırmalar, vaka çalışmaları, matematiksel modeller, insani yardım tedarik zinciri özellikleri ve insani yardım operasyonlarının verimli ve etkili yönetimi için gereken kaynaklar olarak sınıflandırılmışlardır.

Son olarak Sandivari ve ark. (2021), insani yardım lojistiği ve afet müdahale operasyonları hakkında yapılan çalışmaların kapsamlı bir incelemesini yapmışlardır. Literatürdeki çalışmalar, optimizasyon modellerinin geliştirilmesi, verilerin kullanılabilirliği, afetlerin dinamik doğası, belirsizlik ve afet psikolojisi, insan davranışı, organizasyonel başarısızlıklar ve kurumların rolü olarak sınıflandırılmıştır.

Afet yönetimi çalışmalarında bir veya birden fazla aşamanın dikkate alındığı çalışmalar alt bölümlerde detaylı bir şekilde açıklanmıştır.

2.1. Müdahale Aşamalarını Dikkate Alan Çalışmalar

Müdahale aşaması ile ilgili çalışmalar, afet sırasında veya hemen sonrasında hayat kurtarma, yaralıların tedavisini sağlama, barınma, beslenme, güvenlik, psikolojik destek gibi yaşam gereksinimlerinin karşılandığı aşamadır. Burada faaliyetlerin kısa süre içerisinde en uygun yöntemlerle yapılması hedeflenmektedir (Şahin, 2009). Bu aşamada toplulukların ihtiyaçlarının karşılanması ve hayatın bir an önce normale döndürülmesi gerekmektedir. Normale dönüş süresinin mümkün olduğunca kısaltılması sürecin başlıca amacıdır (Kalkınma Bakanlığı, 2014). Temel hedef, olaya anında müdahale ederek mümkün olan en çok sayıda insan ve canlı hayatını kurtarmak ve can kayıplarına da insan onuruna yakışır şekilde müdahale etmektir. Bu kapsamda haberleşme ve ulaşım ağları işletilmeli, barınma ihtiyacı karşılanmalı, ilkyardım ve tedavi hizmetleri yerine getirilmeli, kamu-sivil-özel tüm imkanlar seferber edilmeli ve öncelikli olarak en yakın yerel birim müdahale etmelidir (Döyen, 2018).

Müdahale aşaması acil durum lojistiği alanındaki çalışmaları kapsamaktadır. Acil durum lojistiği alanındaki çalışmalar, afet gerçekleşmeden önce tesis yerleşimi, yardım merkezlerinde insani yardım malzemelerinin stoklarının belirlenmesi vb. faaliyetleri ve afet gerçekleştikten sonra da insani yardım malzemelerinin afetzedelere ulaştırılması, afetzedelerin afet bölgelerinden tahliyesi vb. faaliyetleri kapsamaktadır.

Serra ve Marianov 1998 yılında acil durum lojistiği alanında bir çalışma yapmışlardır. Serra ve Marianov (1998) çalışmalarında, belirsizlikler altında yeni tesislerin konumlandırılmasına yönelik P-medyan problemine dayanan iki farklı model önermişlerdir. Birinci modelde amaç, her bir senaryo için, hizmet alacağı tesise ulaşım süresi en büyük olan talep noktasının ulaşım süresinin veya uzaklığının minimize edilmesidir. İkinci modelde ise amaç, tesislerin en iyi şekilde konumlandırılması durumunda beklenen ulaşım süreleri ile gerçekleşen ortalama ulaşım süreleri arasındaki en büyük farkın minimize edilmesidir. Önerilen modellerde belirsizlikler; talepler, talep noktaları ve tesisler ile talep noktaları arasındaki ulaşım süreleri ya da mesafeleridir. Modellerin çözümü için Teitz ve Bart prosedürüne dayanan sezgisel bir yöntem geliştirilmişlerdir. Sezgisel yöntemin etkinliğini 10 farklı test problemi üzerinde ölçmüşlerdir. Elde edilen sonuçlara göre geliştirilen sezgisel yöntemin birinci model üzerinde oldukça iyi bir performans gösterdiği ancak ikinci modelde on test probleminde yalnızca dördünde optimal sonucu verdiği gözlemlenmiştir. Son olarak

Serra ve Marianov (1998) önerdikleri modelleri Barselona'daki itfaiye istasyonlarının konumlandırılmasında uygulamışlardır.

Barbarosoğlu ve Arda (2004), afet gerçekleşikten sonra acil müdahale aşamasında, insani yardım malzemelerinin afetten etkilenen bölgelere taşınmasını sağlamak için birden fazla insani yardım malzeme tipinin dikkate alındığı, iki kademeli stokastik programlama modeli geliştirmişler ve İstanbul iline uygulamışlardır. Herhangi bir afetin zamanlaması ve şiddeti önceden tahmin edilemediğinden, önerilen modelde talepler ve ulaşım ağının hasar görmesi durumunda taşınabilecek insani yardım malzemesi kapasitesi belirsizdir. Önerilen modelde amaç toplam taşıma, karşılanamayan talep ve stokta bulundurma maliyetlerini minimize etmektir.

Günneç ve Salman (2006), afetten sonra etkin bir müdahale için, İstanbul'daki acil müdahale ve dağıtım merkezlerinin afetten önce etkin bir şekilde konumlandırılmasını hedefleyen çok kriterli iki kademeli stokastik programlama modeli önermişlerdir. Önerilen model, hem acil müdahale ve dağıtım merkezlerinin yer seçimini, arama kurtarma ve sağlık ekiplerinin afetten sonra hazır bulunmasını, insani yardım malzemelerinin afetten önce acil müdahale ve dağıtım merkezlerinde depolanmasını hem de afetten sonra insani yardım malzemelerinin afet bölgelerine ulaştırılmasını dikkate almaktadır. Ele alınan modelde insani yardım malzemelerinin farklı önem seviyeleri bulunmaktadır. Modelde amaçlar, toplam müdahale süresini, her bir malzeme için maksimum hizmet süresini, her tesisin hasar görme durumunu, karşılanamayan talepleri ve tesislerin kurulum maliyetini minimize etmektir.

Görmez ve ark. (2011) çalışmalarında, İstanbul'da oluşabilecek bir depremde önceden konumlandırılacak insani yardım malzeme depolarının yerlerinin belirlenmesi amacıyla iki kademeli tam sayılı stokastik programlama modeli önermişlerdir. Modelin iki farklı amaç fonksiyonu vardır. Birinci amaç fonksiyonu, bir afetzedeye hizmet etmek için kat edilen ortalama mesafeyi minimize etmektir. İkinci amaç fonksiyonu ise kurulacak tesis sayısını minimize etmektir. Model aynı zamanda deprem sonrası meydana gelebilecek hasarları da dikkate almaktadır. Talepler belirsizdir ve belirlenen bölgelerdeki nüfus yoğunluğu ve oluşacak hasar ile ilişkilendirilmiştir. Model çeşitli test problemleri üzerinde uygulanmıştır ve çözümü için Epsilon Kısıt yönteminden faydalanılmıştır.

Salman ve Yücel (2014) çalışmalarında tesis yer seçimi için maksimum sayıda talep noktasını kapsamak amacıyla, tam sayılı stokastik doğrusal programlama modeli önermişlerdir. Modelde, bağlantı yollarının hasar görebileceği durumlar da dikkate

alınmıştır. Modelin çözümü için Tabu Arama algoritması önerilmiştir. Önerilen algoritmanın etkinliği değerlendirmek amacıyla bağlantı yollarının hasar görmediği durum, bağlantı yollarının birbirinden bağımsız ve bağımlı olduğu durumlar için İstanbul iline ait gerçek veriler kullanılarak çeşitli test problemleri üzerinde karşılaştırmalar yapılmıştır. Sonuçlar çözüm yönteminin bağlantı yollarının hasar görmediği ve birbirinden bağımsız olduğu durumlar için oldukça etkili olduğunu göstermektedir.

Chang ve ark. (2007) tarafından yapılan çalışmada, acil durum lojistiğinin planlanmasında devlet kurumları tarafından kullanılabilen bir karar verme aracı geliştirmek amaçlanmıştır. Bu amaçla çalışmalarında, kentsel sel felaketlerinde insani yardım malzemelerinin dağıtım planı için iki kademeli stokastik programlama modeli önermişlerdir. Modelde talepler belirsizdir ve belirlenen bölgelerdeki nüfus yoğunluğu ile ilişkilendirilmiştir. Bu çalışmada afetten önce tesislerin konumlandırılmasına ve stoklanacak malzemelere karar verilmekte, afetten sonra ise hangi tesisten hangi afet bölgesine yardım malzemesi gönderileceğine karar verilmektedir. Modelin amaç fonksiyonu ise taşıma ve karşılanamayan talep maliyetlerini minimizasyonudur. Son olarak, Tayvan'a ait veriler kullanılarak sel felaketleri için acil durum planı hazırlanmıştır.

Garrido ve ark. (2015) ise sel sonrası insani yardım malzemelerinin stok seviyelerinin belirlenmesi ve acil yardım malzemelerinin dağıtımını üzerine bir çalışma yapmışlardır. Talepler belirsizdir ve afetin şiddetine bağlı olarak hesaplanmaktadır. Çalışmalarında, stok maliyetlerinin ve taşıma sürelerinin minimizasyonunun hedeflendiği karışık tam sayılı programlama modeli önermişlerdir. Talepteki belirsizlik Monte – Carlo benzetim yöntemi ile elde edilmiş ve farklı parametre değerleri için elde edilen sonuçlar karşılaştırılmıştır.

Jia ve ark. (2007), büyük ölçekli acil durumlarda (deprem, terör saldırıları, vb.) hizmet verecek acil müdahale merkezlerinin konumlandırılmasına yönelik bir çalışma yapmıştır. Yapılan çalışmada, büyük ve küçük ölçekli afetlerin durumları ayrı ayrı analiz edilmiş ve kapsama modelleri, P-medyan modelleri ve P-merkez modelleri baz alınarak bir tesis yerleşim modeli geliştirilmiştir. Bu çalışmada diğer çalışmalardan farklı olarak her bir talep noktasının hizmet seviyesi de dikkate alınmıştır. Önerilen model, Los Angeles'taki tıbbi malzeme tesislerinin (sağlık merkezlerinin) konumlarını optimize etmek amacıyla gerçek veriler üzerinde uygulanmıştır. Ayrıca önerilen model ile geleneksel kapsama modelleri, P-medyan modelleri ve P-merkez modelleri ile elde

edilen çözümler karşılaştırılmış ve önerilen modelin can ve mal kayıplarını azaltmada oldukça etkili olduğu sonucuna varılmıştır.

Balcik ve Beamon (2008), afetlerden etkilenen insanların ihtiyaçlarını karşılamak amacıyla, insani yardım merkezlerinin sayısının, konumlarının ve bu yardım merkezlerinde stoklanacak malzeme miktarının belirlendiği, temeli kapsama modellerine dayanan, iki kademeli stokastik programlama modeli önermişlerdir. Önerilen modelde önem dereceleri birbirinden farklı olan birden fazla malzeme çeşidi dikkate alınmıştır. Ayrıca bir talep noktasının birden fazla yardım merkezinden hizmet alabileceği ve her bir yardım merkezinin birbirinden farklı kapasitelerinin olduğu varsayılmıştır. Bu çalışmadaki temel amaç, yardım merkezlerinin karşıladığı toplam talebi maksimize etmektir. Çalışmada ayrıca önerilen modelin uygulanabilirliğini ve afet öncesi faaliyetleri için ayrılan bütçe ile afet sonrası faaliyetleri için ayrılan bütçenin elde edilen sonuçlar üzerindeki etkisini göstermek amacıyla çeşitli test problemleri geliştirilmiştir. Elde edilen sonuçlar, afet öncesi faaliyetleri için ayrılan bütçenin, afet sonrası faaliyetleri için ayrılan bütçeye göre, müdahale süresi ve karşılanan talebin oranı bakımından daha iyi sonuçlar elde edilmesini sağladığı ve afet öncesi faaliyetlerine daha fazla önem verilmesi gerektiğini göstermiştir.

Mete ve Zabinsky (2010), farklı afet türleri ve büyüklüklerinin dikkate alındığı iki kademeli stokastik programlama modeli geliştirmişlerdir. Geliştirilen modelin amacı, afete hazırlık aşamasında insani yardım ve dağıtım merkezlerinin konumları, stoklanacak tıbbi malzeme miktarı, afetten sonra müdahale aşamasında ise taşınacak insani yardım malzemelerinin miktarları gibi kararlar almaya yardımcı olmaktır. Modelin amaç fonksiyonu açılacak insani yardım ve dağıtım merkezlerinin maliyetini, taşıma ve karşılanamayan taleplerin maliyetini minimize etmektedir. Mete ve Zabinsky (2010) tarafından önerilen model, karar vericilere depreme hazırlık planları yapmalarında yardımcı olmak amacıyla Seattle bölgesinde meydana gelebilecek farklı deprem senaryoları için uygulanmıştır.

Li ve ark. (2011), kasırganın etkilerini azaltmak amacıyla, afetten önce kurtarma merkezlerinin (sığınakların) kurulduğu, afetten sonra ise afetzedelerin ve yardım malzemelerinin kurtarma merkezlerine taşındığı iki kademeli stokastik programlama modeli önermişlerdir. Modelde tahliye edilenlerin sayısı belirsizdir ve amaç karşılanamayan taleplerin minimizasyonudur. Kurtarma merkezlerinin kasırgada zarar görmediği varsayımı altında 45 farklı senaryo geliştirilmiştir ve önerilen modelin çözümü için modeli senaryolara bağlı olarak alt problemlere ayırıştıran L-Şekilli

yöntemi kullanılmıştır. Modelin uygulanabilirliğini ve yöntemin etkinliğini göstermek amacıyla da model C++ dilinde kodlanıp CPLEX ile çözdürülmüştür. Elde edilen sonuçlar, L-Şekli yönteminin kısa bir sürede optimal sonuca yakın sonuçlar verdiğini göstermiştir.

Li ve ark. (2012), Li ve ark. (2011) gibi olası bir kasırmanın meydana gelmesi durumunda etkin bir müdahale aşamasını sağlamak için iki kademeli stokastik programlama modeli geliştirmişlerdir. Önerilen modelde, Li ve ark. (2011) tarafından geliştirilen modelde olduğu gibi ilk kademede kurtarma merkezlerinin (sığınakların) konumları ve bu kurtarma merkezlerinin hizmet vereceği afetzede sayısı ile ilgili kararlar verilmektedir, ikinci kademesinde ise ulaşım süreleri de dikkate alınarak tahliye yolları belirlenmektedir. Amaç fonksiyonu, karşılanamayan taleplerin ve ulaşım sürelerinin minimizasyonudur. Önerilen modelin daha hızlı ve kolay çözülmesi amacıyla Lagrange Gevşetmesi yöntemine ve senaryo ayrıştırmasına dayalı sezgisel bir yöntem geliştirilmiştir. Son olarak modelin büyük ölçekli problemlere uygulanabilirliğini göstermek amacıyla Kuzey Karolina eyaleti için bir vaka çalışması yapılmıştır.

Kasırğa etkilerini hafifletmek için yapılan bir diğer çalışma ise Hong ve ark. (2015) tarafından yapılmıştır. Hong ve ark. (2015), belirsiz talep ve taşıma kapasiteleri altında iki kademeli stokastik programlama modeli, bir de şans kısıtlı stokastik programlama modeli geliştirmişlerdir. Modellerin ilk kademesinde yardım ve dağıtım merkezlerinin konumlarına, boyutlarına ve stoklanacak insani yardım malzemelerinin seviyelerine, ikinci kademesinde ise insani yardım malzemelerinin afetzedelere ulaştırılması için kararlar verilmektedir. Modellerde tesislerin ve bağlantı yollarının hasar görmediği varsayılmıştır ve bağlantı yollarının kullanılabilirliğini temsil etmek için Gale-Hoffman eşitsizlikleri kullanılmıştır. Önerilen modellerin çözümünde öncelikle gereksiz Gale-Hoffman eşitsizliklerini ortadan kaldırmak amacıyla bir algoritma kullanılmıştır. Daha sonra modeller, son derece hızlı bir şekilde çözülebilen MIP problemleri olarak yeniden formüle edilmiştir. Geliştirilen modellerde ayrıca, insani yardım malzemesi taleplerinin yüksek bir olasılıkla karşılanmasını sağlamak amacıyla karşılanamayan her bir birim talep için ceza maliyeti verilmiştir. Son olarak, modellerin etkinliğini ölçmek için rassal test problemleri oluşturulup çözülmüştür.

Bozorgi-Amiri ve ark. (2012) çalışmalarında, taleplerin, yardım merkezlerinin konumlarının, stoklanacak insani yardım malzemesi miktarlarının ve taşıma maliyetlerinin belirsiz ve senaryolara bağlı olduğu iki kademeli stokastik programlama

modeli geliřtirmişlerdir. Afetten önce yardım merkezlerinin konumlarına ve stoklanacak insani yardım malzemesi miktarlarına, afetten sonra ise afetzedelerin yardım merkezlerine taşınması kararları verilmektedir. Bu çalışmada amaç beklenen toplam maliyeti (tesis yerleşimi, taşıma, elde bulundurma ve karşılanamayan talep maliyetleri) minimize etmektir. Önerilen matematiksel modelin çözümü için parçacık sürü optimizasyonu algoritmasına dayalı bir çözüm yöntemi önerilmiştir. Önerilen çözüm yönteminin etkinliğini ölçmek için küçük ve orta dereceli test problemleri geliştirilmiş ve önerilen yöntemin oldukça iyi sonuçlar verdiği gösterilmiştir.

Verma ve Gaukler (2014), depremden sonra sağlanan hizmetin kalitesini artırmak için afet müdahale tesislerinin yerlerinin seçilmesi ve bu tesislerde depolanacak insani yardım malzemesi miktarlarına karar verilebilmesi amacıyla iki farklı tesis yerleşim modeli geliřtirmişlerdir. Birinci model, afet müdahale tesisleri ve afetzedelerin buldukları bölgeler arasındaki mesafenin afetin etkisi üzerindeki öneminin dikkate alındığı deterministik bir modeldir. İkinci model ise, birinci modele ek olarak afetin şiddetinin rassal bir değişken olarak alındığı, iki kademeli stokastik bir programlama modelidir. İkinci modelin çözümünde, iki kademeli stokastik programlama modellerinin çözümü için kullanılan Benders Ayrıştırma yöntemine dayanan yeni bir çözüm yöntemi geliştirilmiştir. Kaliforniya’da meydana gelen bir depremden elde edilen veriler kullanılarak modellerin etkinliği ölçülmüştür. Elde edilen sonuçlar, geliştirilen iki kademeli stokastik programlama modelinin, deterministik modele göre beklenen maliyeti önemli ölçüde azalttığını göstermiştir.

Döyen ve ark. (2012), hem afetten önce (kalıcı) hem de afetten sonra (geçici) kurtarma merkezlerinin kurulması, kurtarma merkezlerinde stoklanacak insani yardım malzemesi miktarı, afetzedelere gönderilecek insani yardım malzemesi miktarı ve karşılanamayan taleplerin miktarı gibi kararların alındığı iki kademeli stokastik programlama modeli geliřtirmişlerdir. Geliřtirilen modelin amacı, toplam tesis kurulum maliyeti, insani yardım malzemelerini elde bulundurma maliyeti, taşıma ve karşılanamayan taleplerin maliyetini minimize etmektir. Modelin çözümü için Lagrange Gevşetmesi yöntemine dayanan yeni bir çözüm yöntemi önerilmiştir. Rassal oluşturulan test örnekleri ile önerilen çözüm yönteminin 25 senaryoya kadar iyi bir performans gösterdiği sonucuna varılmıştır.

Tofighi ve ark. (2016) tarafından merkezi depolardan dağıtım merkezlerine insani yardım malzemelerinin gönderildiği bulanık tabanlı çok amaçlı iki kademeli stokastik programlama modeli sunulmuştur. İlk kademede, merkezi depolarda

stoklanacak insani yardım malzemesi seviyeleri ile birlikte merkezi depoların ve dağıtım merkezlerinin konumları belirlenmektedir. Bu kademedeki talepler, taşıma süreleri ve deprem sonrası bağlantı yollarının kullanılabilirlik durumları belirsizdir. İkinci kademedeki ise, çeşitli afet senaryoları dikkate alınarak, toplam dağıtım süresi, kritik malzemeler için maksimum dağıtım süresi, kullanılmayan stokların toplam maliyeti ve karşılanmayan taleplerin maliyetini en aza indirmeyi amaçlayan bir dağıtım planı belirlenmektedir. Geliştirilen modelin çözümü için diferansiyel evrim algoritmasını kullanılmış ve Tahran için örnek bir vaka analizi yapılmıştır. Elde edilen sonuçlardan, önerilen dağıtım planının önceden planlanmış dağıtım planından daha iyi performans gösterdiği sonucuna varılmıştır.

Rennemo ve ark. (2014) afet müdahale planlaması için üç kademeli karma tam sayılı stokastik programlama modeli geliştirmiştir. İlk kademedeki yardım tesislerinin açılması ve merkez tesisten açılan yardım tesislerine insani yardım malzemelerinin dağıtımının gerçekleşmesi hedeflenmiştir. İkinci kademedeki, insani yardım malzemelerinin afetzedelere dağıtılması için araçların rotaları planlanmıştır. Son kademedeki ise modeldeki talep belirsizlikleri, araç filosunun kapasitesi ve alt yapı imkânları doğrultusunda değerlendirilmiştir. Gerçek veriler kullanılarak çeşitli test örnekleri oluşturulmuştur. Test örnekleri öncelikle stokastik programlama modeli olarak farklı senaryolara göre çözülmüştür. Daha sonra problem ortalama değerlere göre çözülmüş ve stokastik parametrelerin ortalama değerleri kullanılarak elde edilen çözüm ile stokastik programlamanın sonucu karşılaştırılmıştır. Sonuç olarak, stokastik programlama modeli ile elde edilen sonuçların, stokastik parametrelerin ortalama değerleri kullanılarak elde edilen sonuçlara göre önemli ölçüde daha iyi olduğunu göstermektedir.

Mohammadi ve ark. (2016), yardım malzemelerinin depolanması, optimal stok miktarı ve optimal malzeme akış miktarının belirlenmesi üzerine çalışmışlardır. Ele alınan problem için çok amaçlı stokastik programlama modeli önerilmiştir. Talep noktalarının kapsanmasının maksimize, maliyetin ve hizmet düzeyindeki farklılıkların ise minimize edilmesinin amaçlandığı modelde, çözüm için genotip-fenotip tabanlı parçacık sürü optimizasyonu algoritması geliştirilmiştir. Önerilen modelin ve çözüm yönteminin etkinliğini ölçmek için Tahran'daki deprem senaryolarına dayanan örnek bir vaka analizi yapılmıştır. Önerilen yöntem Deb ve ark. (2022), Zitzler ve ark. (2001) ve Laumanns ve ark. (2006) tarafından geliştirilen yöntemlerle karşılaştırılmıştır. Elde

edilen sonuçlar, önerilen yöntemin karşılaştırılan diğer yöntemlere göre üstün bir performans gösterdiği gözlemlenmiştir.

Irohara ve ark. (2016) çalışmalarında afet gerçekleşmeden, hazırlık aşaması için üç kademeli bir model önermişlerdir. İlk kademede, hangi potansiyel yardım merkezlerinin açılacağına ve stoklanacak insani yardım malzemelerine karar verilir ve amaç bu maliyetleri minimize etmektir. İkinci kademe, afetin neden olduğu hasarı ve maliyetini temsil etmektedir. Üçüncü kademede ise, afetten sonra müdahale ve kurtarma kararları belirlenmekte ve maliyetleri minimize edilmektedir. Üç kademeli matematiksel modelin formülasyonunda stokastik programlama veya şans kısıtlı stokastik programlama yerine yasaklama çerçeveli bütünleşik bir model önerilmiştir. Çeşitli test problemleri geliştirilerek model çözülmüştür ve önerilen modelin etkili sonuçlar verdiği gözlemlenmiştir.

Moreno ve ark. (2015) bu çalışmada, iki kademeli çok dönemli, birden fazla malzeme tipini, araç tipini ve sayısını dikkate alan iki karışık tam sayılı stokastik programlama modeli geliştirilmişlerdir. İlk kademede, hangi potansiyel yardım merkezlerinin açılacağı veya hangi mevcut yardım merkezlerin kullanılacağı ve depolara atanacak araçların seçildiği hazırlık aşaması kararları verilmektedir. İkinci kademede ise, kullanılacak rotalar (depo-yardım merkezi), ihtiyaç duyulan araçların tipi (model 1) ve sayısı (model 2) gibi afet müdahale kararları verilmektedir. Brezilya'nın Rio De Janeiro'nun dağlık kesimlerinde meydana gelen afetten elde edilen gerçek ve tahmini verileri kullanarak farklı senaryolara göre örnekler oluşturulmuştur. Bu örnekleri çözmek amacıyla Relax-and-Fix ve Fix-and-Optimize sezgisel yöntemleri geliştirilmiştir. Elde edilen sonuçlardan, birinci model için CPLEX' in, ikinci model için de sezgisel yöntemlerin daha iyi sonuçlar verdiği gözlemlenmiştir.

Bai (2016) çalışmasında, afetten önce bağlantı yollarının güçlendirildiği, dağıtım merkezlerinde insani yardım malzemelerinin stoklandığı, stoklanacak malzemelerin türüne ve miktarına ve kullanılacak araç sayısına karar verildiği, afetten sonra ise insani yardım malzemelerinin afet bölgelerine dağıtıldığı iki kademeli stokastik programlama modeli geliştirmiştir. İnsani yardım malzemelerinin afet bölgelerine dağıtılması aşamasında bağlantı yollarının kullanılabilirlik durumları dikkate alınmaktadır. Adil bir dağıtım, dağıtım sürelerinin, stokları elde bulundurma maliyetlerinin ve karşılanamayan taleplerin minimizasyonu için çok amaçlı bir model oluşturulmuştur. Geliştirilen modelin çözümü için hedef programlamadan faydalanılmıştır.

Alem ve ark. (2016), afetin etkilerinin azaltılması amacıyla afetzedelere insani yardım malzemelerinin nasıl hızlı bir şekilde ulaştırılacağına karar vermeye yardımcı olacak iki kademeli stokastik programlama modeli sunmuşlardır. Afetten önce stoklanacak malzeme miktarına ve kullanılacak araç türüne karar verilirken, afetten sonra insani yardım malzemelerinin afetzedelere hızlı bir şekilde dağıtılması amacıyla kararlar verilmektedir. Rio de Janeiro’da meydana gelen sel ve toprak kaymalarına dayanan çeşitli test problemleri geliştirilmiştir ve model birçok senaryoda afet planlamasında karar vericilere yardımcı olabileceği sonucuna varılmıştır.

Elçi ve Noyan (2018), afet sonrasında taleplerde ve ulaşım ağı koşullarındaki belirsizlikleri dikkate alan, kurtarma tesislerinin konumlarını, kapasitelerini ve insani yardım malzemelerinin envanter seviyelerini belirleyen stokastik bir afet öncesi yardım ağı tasarım problemini ele almışlardır. Geleneksel insani yardım lojistiği literatürünün aksine, ortalama riskli şans kısıtlı iki kademeli stokastik programlama modeli geliştirmişlerdir. Bu riskten kaçınan modelde, belirli bir risk değeri hesaplanıp, afet durumunda etkilenen bölgelere maruz kalınan risk değerine göre insani yardım malzemelerinin tedariki sağlanmaktadır. Modeli çözümü için Benders yöntemine dayanan dal-kesme algoritması önerilmiştir. Elçi ve Noyan (2018), Amerika Birleşik Devletleri’nin Güneydoğu bölgesindeki kasırgaları dikkate alarak örnek bir vaka çalışması yapmışlardır ve algoritmanın etkinliği incelenmişlerdir. Elde edilen sonuçlar önerilen algoritmanın CPLEX’ e göre daha iyi sonuçlar verdiği gözlemlenmiştir.

Mohaymany ve Nikoo (2020), halk tarafından kullanılmaması gereken afet müdahale yollarının tasarımı üzerinde bir çalışma yapmışlardır. Afet müdahale yolları, afetten sonraki 72 saat boyunca yaralı insanların ve insani yardım malzemelerinin taşınmasında çok önemli bir rol oynamaktadır. Bu yollar, büyük yardım merkezlerini birbirine bağlayan yollardır. Bir şehir için müdahale yollarının önceden belirlenmesi, müdahale ekiplerinin afet bölgelerine hızlı bir şekilde ulaşmasını, şehirden tahliye edilenlerin dışarı akışıyla engellenmeden, yardım ve kurtarma operasyonları gerçekleştirmesini sağlar. Bu yollar genellikle tahliye yollarından farklı olarak halk tarafından kullanılmamalıdır. Bu çalışmada, çok amaçlı bir stokastik afet müdahale yolları tasarım modeli önerilmiştir. Modelin çözümü için dal-kesme yöntemi önerilmiştir. Önerilen çözüm yönteminin etkinliğini ölçmek için orta ve büyük dereceli test problemleri geliştirilmiş ve önerilen yöntemin oldukça iyi sonuçlar verdiği gösterilmiştir.

Rath ve ark. (2015) yaptığı çalışma bulunmaktadır. Rath ve ark. (2015) insani değerlerin de dikkate alındığı iki kademeli stokastik programlama modeli sunmuşlardır. Önerilen modelde, talepler ve yardım merkezlerindeki stok miktarları deterministiktir. Ancak bağlantı yollarında meydana gelebilecek potansiyel hasarların da dikkate alınmasından dolayı ulaşım süreleri belirsiz olmaktadır. Modelin ilk kademesinde bütçeye, yardım merkezlerinin konumlarına ve araç tedarikine karar verilmektedir. İkinci kademesinde ise yardım merkezlerinden afet bölgelerine insani yardım malzemelerinin ulaştırılması ve araçların yardım merkezlerine atanması kararları verilmektedir. Ayrıca, önerilen modelin bütçenin afet gerçekleşmeden önce (her bir senaryo için bütçe miktarı) veya afet gerçekleştikten sonra belirlenmesi ya da araçların yardım merkezlerine afetten önce veya sonra atanması kararlarına göre farklı varyasyonları geliştirilmiştir ve bu farklılıkların çözümler üzerindeki etkileri analiz edilmiştir.

Günneç ve Salman (2011), bir bağlantı yolunun kullanılabilirliğini farklı performans ölçütlerine göre ve bu performans ölçütlerinin bir başlangıç ve bitiş düğümü arasındaki beklenen ulaşım süresi ya da mesafe üzerindeki etkilerini değerlendirmişlerdir. Bu çalışmada, diğer çalışmalardan farklı olarak, bağlantı yollarında meydana gelen hasarların diğer bağlantı yollarını da etkileyebileceği dikkate alınmaktadır. Performans ölçütlerinin hesaplanmasında Monte Carlo simülasyonundan faydalanılmış ve İstanbul için örnek bir vaka çalışması yapılmıştır.

Viswanath ve Peeta (2003) ele aldıkları problem için çok amaçlı tam sayılı programlama modeli geliştirmişlerdir. Model, depremden sonra kullanılabilir olması gereken bir başlangıç-bitiş (O-D) çifti için kritik rotaları bulmayı hedeflemektedir. Rotalar, toplam taşıma süresini en aza indirmek ve başlangıç-bitiş noktaları arasındaki maksimum popülasyonu kapsayacak şekilde oluşturulmuştur. Kısıtlı bir bütçe ile köprülerin güçlendirilmesiyle bağlantı yollarının kullanılabilirliği artırılmıştır. Bu çalışmada model deterministiktir, depreme bağlı belirsizlikler dikkate alınmamıştır.

Hu ve ark. (2019), insani yardım malzemelerinin dağıtımı için birden fazla araç türünün kullanıldığı, bu araçların afetten sonra kiralanmasında yaşanabilecek dalgalanmayı ve afetten sonra ulaşım ağlarının durumunu dikkate alan çok aşamalı bir stokastik programlama modeli geliştirilmişlerdir. Modelde, ulaşım ağlarının durumu (ulaşılabilir durumda olup olmadığı) belirsizdir ve her farklı afet senaryosu için belirli bir kapasiteyle sınırlandırılmıştır. Önerilen modeli büyük ölçekli bir problemde çözebilmek için PHA (Progressive Hedging Algorithm) algoritmasını geliştirmişlerdir.

Önerilen model Çin’ de gerçekleşen Yaan depremine uygulanmıştır ve çözüm yönteminin oldukça etkili sonuçlar verdiğini gözlemlemişlerdir.

Rawls ve Turnquist (2010), çalışmalarında kasırgalara karşı bir acil müdahale planı geliştirmişlerdir. İnsani yardım malzeme depolarının yerlerinin ve kapasitelerinin belirlenmesini amaçlayan karışık tam sayılı stokastik programlama modeli kurulmuştur. Maliyet minimizasyonu hedeflenmiştir. Hem binaların hem de bağlantı yollarının hasar görülebilirlik durumlarının dikkate alınması çalışmayı diğer çalışmalardan ayırmaktadır. Rawls ve Turnquist (2011) ise bir önceki çalışmalarında kurdukları modele ek olarak hizmet kalite kısıtları eklemişlerdir. Ek hizmet kalite kısıtları eklendiğinde, önceki çalışmada bulunan çözümler ile yeni çalışmada bulunan çözümlerin farklılık gösterdiği gözlemlenmiştir.

2.2. Hafifletme ve Müdahale Aşamalarını Dikkate Alan Çalışmalar

Hafifletme aşaması afet olaylarının etkilerinin önlenmesi veya büyük kayıplara sebep olmaması için gerekli tedbirleri içerir. Bu aşama iyileştirme ile yeni bir afet olana kadar olan süreci kapsar (Şahin, 2009). Bu aşamada insanları, mal, yapı, hizmet, doğal ve kültürel kaynakları korumak ile afet sonrası ve sırasındaki çalışmalarının maliyetini azaltmak amaçlanır (Kadıoğlu, 2011). Hafifletme faaliyetleri, doğal afetlerin olası etkilerini azaltmak amacıyla gerçekleştirilen bina veya bağlantı yolu güçlendirme faaliyetlerini kapsamaktadır. Müdahale aşaması ise afet sonrası acil durum malzemelerinin afet bölgelerine ulaştırıldığı ya da afetzedelerin afet bölgelerinden tahliye edildiği aşamadır. Bu bölümde incelenen çalışmalarda modelin ilk kademesinde bağlantı yolları için güçlendirme kararları alınmaktadır, ikinci kademesinde ise taşıma kararları alınmaktadır.

Afet sonrasında yardım merkezleri ve afet bölgeleri arasındaki ulaşılabilirliği artırmak, insani yardım malzemelerinin etkin bir şekilde dağıtılmasını ve yeniden yapılandırma maliyetlerinin en aza indirilmesini sağlamak amacıyla yapılan çalışmalar arasında, Peeta ve ark. (2010) yaptığı çalışma bulunmaktadır. Peeta ve ark. (2010), afet nedeniyle hasar görebilecek bağlantı yollarını afetten önce güçlendirmeyi amaçlayan iki kademeli stokastik programlama modeli geliştirmişlerdir. Afet gerçekleştikten sonra her bir bağlantı yolu ya kullanıma açık olmaktadır, ya da, hasara açık olmasına rağmen güçlendirilmemiş olması halinde kullanıma kapalı olmaktadır. Bağlantı yollarının hasar görme olasılıklarının önceden bilindiği varsayılmıştır. Alınan güçlendirme kararları ise

bu olasılıkları azaltmaktadır. Bu model, karar vericilere afet sonrasında dağıtım merkezleri ile afet bölgeleri arasındaki ulaşılabilirliği arttırmak ve toplam insani yardım malzemelerini taşıma maliyetlerini azaltmak amacıyla sınırlı bir bütçe ile hangi bağlantı yollarının güçlendirileceğine karar vermelerine yardımcı olmaktadır. Ayrıca, kurtarma merkezlerinden afet bölgelerine ulaşılabilirliği arttırmak amacıyla kurtarma merkezi ile belirlenen bir afet bölgesi arasında bağlantı yolunun kullanıma kapalı olması durumunda ceza maliyeti eklenmektedir. Son olarak, karar vericilere etkin afet planlamaları yapmalarına yardımcı olmak amacıyla İstanbul'daki karayollarının deprem riskine karşı güçlendirilmesine ilişkin çeşitli test problemleri geliştirilmiştir.

Daha sonra Du ve Peeta (2014), Peeta ve ark. (2010) tarafından önerilen modeli daha gerçekçi bir model olarak geliştirmişlerdir. Bir bağlantı yolunun önemini trafik akışı ve güçlendirme maliyetine göre değerlendirmişlerdir. Bir bağlantı yolunu güçlendirme konusunda kısmi yatırımları modellemek için ikili karar değişkenleri yerine sürekli karar değişkenleri kullanmışlardır. Du ve Peeta (2014), Peeta ve ark. (2010) tarafından önerilen modelin aksine, tek bir senaryo yerine birden fazla deprem senaryosu dikkate almışlardır.

Augusti ve ark. (1994), köprülerin hasara açık bağlantı yolları olduğu varsayımı altında, İtalya'da bulunan bir otoyol ağındaki köprüler için depremden önce alınabilecek güçlendirme kararlarının verilmesi amacıyla dinamik programlama modeli önermişlerdir. Bu çalışmanın temel amacı, ele alınan otoyolun güvenilirliğini/kullanılabilirliğini en üst düzeye çıkarmaktır.

Liu ve ark. (2009), sınırlı bir bütçeyle bir ulaşım ağının dayanıklılığını ve sağlamlığını artırmak ve bu ağ üzerinde bulunan köprüleri depremden önce güçlendirmek amacıyla iki kademeli stokastik programlama modeli önermişlerdir. Bir bağlantı yolunun depremden önce güçlendirilmesi durumunda, depremden sonra hasar görmeyeceği varsayılmıştır. Önerilen modelin çözümünde L-Şekilli yöntemi kullanmışlar ve oldukça etkili sonuçlar elde etmişlerdir.

Yücel ve ark. (2018) bir beklenen afet sonrasında ulaşılabilirliği iyileştirmek amacıyla bir ulaşım ağının afetlere karşı dayanıklılığını artırma konusunu ele almaktadır. Problemi bir tam sayılı stokastik programlama çerçevesinde benimseyen yazarlar, hangi bağlantıların güçlendirileceğini belirleyerek afet öncesi yatırım maliyetini en aza indirmek için bir model geliştirmişlerdir. Önerilen modeli tabu arama algoritmasını kullanarak çözmüşler ve İstanbul iline adapte etmişlerdir.

Monzon ve ark. (2020) özellikle gelişmekte olan ve az gelişmiş ülkelerde afetlerin etkilenen nüfus üzerinde yıkıcı etkilerini incelemişlerdir. İnsani Lojistik modellerinin, karar vericilere yardım malzemelerinin verimli ve etkili bir şekilde depolamasında, afetzedelere dağıtılmasında ve kayıpların azaltılmasında yardımcı olabileceğini belirtmişlerdir. Bununla birlikte zayıf planlama ve ulaşım ağlarındaki hasarların bu çabaları engelleyebileceğini ve sonuçta etkilenen talep merkezlerine ulaşmayı imkansız hale getirebileceğini vurgulamışlardır. Monzon ve ark. (2020) çalışmalarında, yardım merkezlerinin afetten önce konumlandırılmasını ve mümkün olduğunca etkilenen nüfusun verimli bir şekilde yardım alabilmesini sağlamak için ulaşım ağlarının güçlendirilmesini hedefleyen bir stokastik programlama modeli geliştirmişlerdir. Modelde talep ve ulaşım ağlarını temsil eden parametreler stokastiktir ve senaryolara göre değişmektedir. Geliştirilen model 2018 yılında Mozambik Nampula Eyaletini vuran fırtınaya adapte edilmiştir.

2.3. Hafifletme ve İyileştirilmesi Aşamalarını Dikkate Alan Çalışmalar

İyileştirme faaliyetlerin ana hedefi, afete uğramış toplulukların haberleşme, ulaşım, su, elektrik, kanalizasyon, eğitim, uzun süreli geçici iskan, ekonomik ve sosyal faaliyetler vb. gibi hayati aktivitelerinin minimum düzeyde karşılanabilmesi için gereken tüm çalışmaları yapmaktır (Döyen, 2018). Bu aşamada hasar görmüş yapıların onarımı ve yeniden inşası gerçekleştirilmektedir. Hafifletme ve iyileştirme faaliyetlerinin bir bütün olarak ele alınarak toplam maliyetlerin en aza indirilmesi gerekmektedir. Bu bölümde incelenen çalışmalarda yalnızca binalar için güçlendirme ve iyileştirme kararları alınmaktadır.

Hafifletme ve iyileştirme faaliyetlerinin bir bütün olarak ele alınarak toplam maliyetlerin en aza indirilmesinin hedeflendiği çalışmalar arasında, Dodo ve ark. (2005), binaların yeniden yapılandırma maliyetlerine katlanmak yerine, deprem gerçekleşmeden önce güçlendirme faaliyetlerine ne kadar harcama yapılacağına ve hangi güçlendirme faaliyetlerinin seçileceğine (binaların hangi seviyeye güçlendirileceğine) karar vermek amacıyla doğrusal programlama modeli geliştirmişlerdir. Amaç toplam güçlendirme ve yeniden yapılandırma maliyetlerini minimize etmektir. Model oluşturulurken bölgesel riskin büyüklüğü ve karakteri, tüm olası güçlendirme faaliyetleriyle ilişkili maliyetler, faydalar ve mevcut bütçe gibi

faktörler dikkate alınmıştır. Dodo ve ark. (2005) modeli Los Angeles County’de uygulamışlardır.

Dodo ve ark. (2007), depremden sonra kayıpları ve yeniden yapılandırma maliyetlerini azaltmak amacıyla kısıtlı bir bütçe ile depremden önce hangi binaların hangi sismik kod seviyelerine yükseltileceğine (güçlendirileceğine) karar vermeye yardımcı olan doğrusal programlama modeli geliştirmişlerdir. Her bir bina yapısal tipi (çelik, beton vb.), tasarım seviyesi (depreme dayanıklılık seviyesi), nüfus çeşidi ve sayısı bakımından farklılık göstermektedir. Önerilen modelin çözümü için Dantzig-Wolfe ayrıştırma yöntemi ve açgözlü sezgisel bir yöntem önerilmiştir. İki algoritma çözüm kalitesi ve çözüm süresi bakımından karşılaştırılmış ve sezgisel yöntemin Dantzig-Wolfe ayrıştırma yöntemine göre daha iyi sonuç verdiği gözlemlenmiştir.

Xu ve ark. (2007), yüksek yeniden yapılandırma maliyetlerini de dikkate alarak, Dodo ve ark. (2007) tarafından yapılan çalışmayı çok amaçlı bir modelle genişletmişlerdir. Bu çalışmada, belirli bir seviyenin üzerine çıkan yeniden yapılandırma maliyetleri cezalandırılarak afet sonrası kayıpların en aza indirilmesi hedeflenmiştir.

Bütçenin kısıtlı olmasından dolayı depremde hasar görmüş bazı binalar depremden hemen sonra onarılamamaktadır. Vaziri ve ark. (2010) ise, hasar görmüş binaların depremden hemen sonra onarılmadığı ve sonraki dönemlerde yeniden inşa edilmesine müsaade edildiği yeni bir model önermişlerdir.

Zolfaghari ve Peyghaleh (2015), ilk kademesinde güçlendirilecek binalara, ikinci kademesinde ise onarılacak ya da yeniden inşa edilecek binalara karar verildiği iki kademeli stokastik programlama modeli geliştirmişlerdir. Geliştirilen model karar vericilere, farklı konumlarda ve planlama ufuklarında, farklı yapısal türdeki binalar için güçlendirmeye yönelik bütçe belirlemeye yardımcı olmaktadır. Önerilen modelin pratikte uygulanabilirliğini göstermek için İran'ın başkenti Tahran'da bir uygulama yapılmıştır.

Legg ve ark. (2013) bölgesel kasırgalara karşı risk yönetimi ve olası bir kasırganın gerçekleşmesi durumunda, kasırganın etkilerini azaltmak için hükümetin ayırdığı kısıtlı bütçenin optimal bir şekilde kullanılmasını sağlayan doğrusal programlama modeli geliştirmişlerdir. Önerilen modelde amaç, afetten önce hangi binaların güçlendirilmesi gerektiği, nasıl ve ne zaman güçlendirileceği, afetten sonra binaların yıkılması durumunda oluşacak maliyet ve yapılan bu yatırımların getirilerinin neler olacağına karar verilmektedir. Model, Kuzey Carolina'nın doğusundaki ahşap

iskeletli konut binalar için uygulanmıştır. Uygulamada hem kasırgadan hem de sel felaketinden kaynaklanan hasarlar ele alınmıştır.

Üstün ve Anagün (2015) çalışmalarında, İstanbul'daki afet riskini azaltmak amacıyla kısıtlı bir bütçe ile bina güçlendirme ve yeniden inşasını dikkate alan çok amaçlı bir model geliştirmişlerdir. Geliştirilen modelde amaç, güçlendirme bütçesinin İstanbul'un ilçeleri arasında optimal bir şekilde nasıl dağıtılacağına, hangi bina türlerinin güçlendirileceğine ve nasıl güçlendirileceğine karar verilmektedir. Son olarak, modelin çözümü için kullanılan Epsilon Kısıt yönteminin etkili bir çözüm verdiği sonucuna varılmıştır.

2.4. Bütünleşik Afet Yönetimi Çalışmaları

Afet öncesi güçlendirme faaliyetleri ile afet sonrası ortaya çıkacak müdahale ve iyileştirme çabaları arasında kuvvetli bir ilişki söz konusudur. Etkin bir afet yönetimi çalışması için afet öncesi, afet esnası ve afet sonrasında alınacak tüm kararlar bir bütün olarak dikkate alınmalıdır. Ve çoğu zaman belirli bir afet yönetimi bütçesinin afet öncesi ve sonrası faaliyetler için dağıtılması gerekmektedir. Literatürde hafifletme, müdahale ve iyileştirme aşamalarını bir bütün olarak dikkate alan çalışmalar az da olsa bulunmaktadır.

Bu çalışmalar arasında Fan ve ark. (2010) yaptıkları çalışma bulunmaktadır. Fan ve ark. (2010) çalışmalarında depremden önce karayolu bağlantı yolları için optimal güçlendirme kararlarını belirlemeyi hedeflemişlerdir. Amaç, kısıtlı bir bütçeyle deprem gerçekleşmeden, deprem öncesi hafifletme faaliyetleri ve depremden sonra müdahale ve iyileştirme maliyetleri için ne kadar harcama yapılacağına karar vermek ve depremin etkisini en aza indirmektir. Fan ve ark. (2010) çalışmalarında iki kademeli stokastik programlama modeli geliştirmişlerdir. Model ilk kademedeki güçlendirilecek bağlantı yollarının, ikinci kademedeki ise iyileştirilecek bağlantı yollarının maliyetini minimize etmektedir.

Miller-Hooks ve ark. (2012) çalışmalarında kısıtlı bir bütçenin hafifletme ve müdahale aşamaları arasında optimal bir şekilde dağıtılması için iki kademeli stokastik programlama modeli geliştirmişlerdir. Geliştirilen modelde, depremden sonra hasar gören bağlantı yollarını iyileştirme maliyetleri de göz önünde bulundurulmuştur. Bu çalışmanın temel amacı bir bağlantı yolunun kullanılabilirliğini arttırmaktır.

Liao ve ark. (2018) yaptıkları çalışma kapsamında, afetler meydana geldiğinde ulaşım ağlarının dayanıklılığının nasıl artırılacağına karar vermeyi, ulaşım ağlarının dayanıklılığını ölçmeyi ve optimize etmeyi amaçlamaktadır. Bütçe ve süre kısıtlamaları altında ulaşım ağlarının güçlendirilmesi için bir optimizasyon modeli önerilmiştir. Önerilen modelde afet öncesi hafifletme ve afet sonrası müdahale aşamaları birlikte dikkate alınmıştır. Modelde depremden sonra hasar gören bağlantı yollarını iyileştirme maliyetleri de göz önünde bulundurulmuştur. Model gerçek afet verileri üzerinde uygulanmıştır. Afetlerden önceki ve sonraki trafik koşulları, DynaTAIWAN simülasyon-atama modeli ile değerlendirilmiştir.

Zhou ve ark. (2019), kentsel alanlarda afet risklerini azaltmak, sürdürülebilir ulaşım sistemleri oluşturmak, dayanıklılıklarını artırmak ve güvenilirliklerini sağlamak için iki kademeli stokastik programlama modeli geliştirmişlerdir. Model afetten sonra insani yardım malzemelerinin ulaşım süresini, ulaşım ağlarının kapasitesini, afet öncesi güçlendirme ve afet sonrası onarım maliyetlerini dikkate almaktadır. Önerilen modelin çözümü için Benzetimli Tavlama ve Frank-Wolfe tabanlı bir algoritma geliştirilmiştir. Yapılan uygulama sonucunda, hesaplama süresinin oldukça hızlı, sonucun makul ve dolayısıyla da geliştirilen algoritmanın etkili olduğunu göstermişlerdir.

Yukarıda açıklanan bütünleşik afet yönetimi çalışmalarında yalnızca bağlantı yollarının güçlendirilmesi dikkate alınmıştır. Binaların da yıkılabileceği durum göz ardı edilmiştir. Literatürde hem binaların hem de bağlantı yollarının güçlendirilmesini dikkate alan yalnızca Döyen ve Aras (2019) çalışması bulunmaktadır.

Döyen ve Aras (2019) çalışmalarında afet öncesi ve sonrası tüm kararların birbirleri ile etkileşimlerini dikkate alan ve kısıtlı afet yönetimi bütçesinin bu kararlar arasında optimal bir şekilde dağıtılmasını sağlayan iki kademeli stokastik programlama modeli geliştirmişlerdir. İlk kademe kararları deprem öncesinde alınan kararlardır ve bina ve bağlantı yollarını güçlendirme faaliyetlerini kapsamaktadır. İkinci kademe kararları ise müdahale aşamasında alınan kararlardır ve insanî yardım malzemelerinin kurtarma merkezlerinden afet bölgelerine hangi miktarda ve ne şekilde ulaştırılacağına dair kararları kapsamaktadır. Döyen ve Aras (2019) tarafından geliştirilen modelde insani yardım malzemesi talep miktarı, deprem senaryolarına ve deprem olduğunda binaların depreme dayanıklılık seviyesine bağlı karar değişkenleri olarak ele alınmıştır. Talep miktarının binaların depreme dayanıklılık seviyesine bağlı olması, talep miktarını azaltmak için, afetten önce etkin bir güçlendirme planının yapılmasının gerekli olduğu anlamına gelmektedir. Modelde, her bir bina için ayrı ayrı güçlendirme kararlarının

verilebiliyor olması oldukça önemlidir. Modelin amaç fonksiyonu; toplam bina ve bağlantı yolu güçlendirme, yardım malzemelerinin taşıma ve karşılanamayan talep maliyetlerinin minimizasyonudur. Geliştirilen model Lagrange Gevşetmesi sezgiseli ile çözülmüştür ve oldukça etkili sonuçlar elde edilmiştir.

2.5. L-Şekilli ve Tam Sayılı L-Şekilli Ayırıştırma Yöntemlerini Kullanan Çalışmalar

Van Slyke ve Wets (1969) yaptıkları çalışmada, stokastik programlama modellerinin çözümü için, sürekli değişkenlerin dikkate alındığı L-Şekilli yöntemi geliştirmişlerdir. Daha sonra Laporte ve Louveaux (1993), birinci kademe karar değişkenlerinin tam sayı, ikinci kademe karar değişkenlerinin ise tam sayılı veya sürekli olduğu Tam Sayılı L-Şekilli yöntemi geliştirmişlerdir. Laporte ve ark. (1994), Santoso ve ark. (2005) ve Listeş (2007) ise L-Şekilli yöntemi stokastik tesis yer seçimi problemlerine uygulamışlardır. Rawls ve Turnquist (2010), Li ve ark. (2011) ve Noyan (2012) L-Şekilli yöntemi insani yardım lojistiği problemlerine uygulamışlardır. Miller Hooks ve ark. (2012) ise yaptıkları çalışmada Tam Sayılı L-Şekilli yöntemi ilk defa insani yardım lojistiği problemlerine uygulamışlardır.

Parragh ve ark. (2021), yardım merkezlerinin en iyi şekilde konumlandırılmasını sağlamak için taleplerin belirsiz olduğu iki amaçlı stokastik programlama modeli geliştirmişlerdir. Geliştirilen model, maliyet minimizasyonu ile birlikte maksimum afetzedenin kapsanmasını amaçlamaktadır. Problemin çözümü için Tam Sayılı L-Şekilli yöntemi kullanılmıştır.

Ramezani ve Ghorbani (2020), afetten sonra insani yardım malzemelerinin dağıtımını ve afetzedelere ulaşabilmek için kullanılacak araçların planlamasını dikkate alan iki kademeli stokastik programlama modeli geliştirmişlerdir. Bir araç kiralama sözleşmesi oluşturarak, afetten sonra araç eksikliğinin ortaya çıkma olasılığını azaltmayı amaçlamışlardır. Afetten önce tedarikçilerle yapılan sözleşmeler çerçevesinde araç kiralama tutarının belirlenmesine, zaman ve maliyet açısından uygun bir kapsama mesafesinin belirlenmesi kararları verilmektedir. Afetten sonra ise, afete maruz kalan bölgelerden gelen taleplere hızlı ve etkin bir şekilde cevap verme kararları verilmektedir. Modelin etkinliğini göstermek için gerçek bir vaka çalışması yapılmıştır ve model L-Şekilli yöntemi ile çözülmüştür.

3. MATERYAL VE YÖNTEM

3.1. İki Kademeli Stokastik Programlama

Afet yönetiminde sıklıkla kullanılan yöntemlerden biri, problemde kullanılan parametrelerin bir kısmının veya tamamının belirsiz olmasından dolayı, iki kademeli stokastik programlama yöntemidir. İki kademeli stokastik programlama modelleri birinci ve ikinci kademe karar değişkenlerinden oluşmaktadır. Birinci kademe karar değişkenlerine rassal olay gerçekleşmeden önce karar verilirken, ikinci kademe karar değişkenlerine rassal olay gerçekleştikten sonra karar verilmektedir. İki kademeli stokastik programlamanın temel fikri, rassal olay gerçekleştikten sonra ikinci kademe karar değişkenlerini belirlerken, daha önce karar verilen birinci kademe karar değişkenlerine göre amaç fonksiyonunu maksimize ya da minimize edecek şekilde düzeltici yönde kararlar verebilmektir.

Bir stokastik programlama modelinin genel formülasyonu aşağıdaki şekildedir;

$$\text{Min } z = \mathbf{c}^T \mathbf{x} + E_{\xi} [\min \mathbf{q}(\omega)^T \mathbf{y}(\omega)] \quad (3.1)$$

$$\text{s.t. } \mathbf{Ax} = \mathbf{b} \quad (3.2)$$

$$\mathbf{T}(\omega)\mathbf{x} + \mathbf{Wy}(\omega) = \mathbf{h}(\omega) \quad (3.3)$$

$$\mathbf{x} \geq 0; \mathbf{y}(\omega) \geq 0 \quad (3.4)$$

Burada ω senaryo parametresinin $\omega=1, 2, \dots, |\Omega|$ değer aralığına sahip Ω kümesinin bir elemanı olduğunu ve rassal olayın gerçekleşme vektörünü temsil eden $\xi(\omega)$ ile ve ω senaryosunun gerçekleşme olasılığını gösteren $p(\omega)$ ile ilişkili olduğunu varsayalım. Öyle ki;

$$\sum_{\omega \in \Omega} p(\omega) = 1 \quad (3.5)$$

Rassal parametre vektörlerinin olasılık dağılımı kesikli olduğunda, belirsizlik, olası gerçekleşme sırasını (senaryoları) temsil eden bir olay ağacı biçiminde rastgele bir yapı tanımlamaktadır. Olay ağacı verildiğinde ise model, senaryo tabanlı düzeltme tipi stokastik programlama problemi olarak tanımlanmaktadır. Bu tip problemlerde ikinci karar aşamasında, rassal olaydan sonra ilave kararlar verilmektedir. Bu kararlar, ilk kademedeki verilen kararları, rassal olaya uygun hale getirmek için verilecek düzeltme

kararlarıdır. Yukarıdaki modelde x ve $y(\omega)$ sırasıyla birinci ve ikinci kademe karar değişkenlerini ifade etmektedir.

Birinci kademe kararları x , $n_1 \times 1$ vektörü ile gösterilmektedir. İlgili vektörler ve c , b ve A matrisleri sırasıyla $n_1 \times 1$, $m_1 \times 1$, $m_1 \times n_1$ boyutlarındadır.

İkinci kademede, bir dizi rassal $\omega \in \Omega$ olay gerçekleşebilmektedir. $y(\omega)$ gibi fonksiyonel formlar, ω senaryolarına bağlılığı göstermek için kullanılmaktadır. Gerçekleşen bir ω senaryosu için, problemin ikinci kademe verileri $q(\omega)$, $h(\omega)$ ve $T(\omega)$ bilinir hale gelmektedir. Dolayısıyla q , h ve T ' nin her bileşeninin olası bir rassal değişken olduğunu söyleyebiliriz. Burada $q(\omega)$ $n_2 \times 1$, $h(\omega)$ $m_2 \times 1$ ve $T(\omega)$ $m_2 \times n_1$ boyutlarındadır.

$T_i(\omega)$ ' nin, $T(\omega)$ ' nin i .satırı olduğunu varsayalım. O halde, ξ , $\xi^T(\omega) = (q(\omega)^T, h(\omega)^T, T_1(\omega), \dots, T_{m_2}(\omega))$ gerçekleşen deprem senaryosunu temsil eden rassal bir vektör olduğu söylenebilmektedir. İkinci kademe kararı $y(\omega)$ veya $y(\omega, x)$, ξ 'nin gerçek değeri bilindikten sonra tahmin edilebilmektedir.

Denklem (3.1)'de verilen amaç fonksiyonu, tüm olası senaryolar üzerinden birinci kademe kısıtlarını $Ax = b$ 'yi sağlayacak şekilde ilk kademe kararlarının maliyeti ($c^T x$) ile ikinci kademe kararlarının beklenen maliyetinin ($E_\xi [\min q(\omega)^T y(\omega)]$) toplamını minimize etmeyi amaçlamaktadır. Birinci kısıt (3.2), birinci kademe problemine ait kısıt ifade ederken, ikinci kısıt (3.3) ise senaryolara bağlı ikinci kademe problemini ifade etmektedir. Birinci kısıta ait karar değişkenleri deterministik, ikinci kısıta ait karar değişkenleri (ω) senaryosuna bağlı stokastik değişkenlerdir. Düzeltme (iyileştirme) maliyeti q , hem ilk kademe kararı olan x ' e hem de gerçekleşen rassal ω olayına bağlıdır. Üçüncü kısıt (3.4) ise karar değişkenlerine ait işaret kısıtıdır.

Gerçek hayatta, stokastik programlama modellerinin çoğu tam sayılı çözümlere ihtiyaç duyar bu da birinci veya ikinci kademede bazı değişkenlerin tam sayılı olmasını gerektirmektedir. İki kademeli ve tam sayılı stokastik programlama modelinin genel formülasyonu aşağıda verilmiştir. Formülasyonda c , b , ξ , A , W , T ve h yukarıda tanımlandığı şekildedir.

$$\text{Min } z = c^T x + E_\xi [\min\{q(\omega)^T y : Wy = h(\omega) - T(\omega)x, y \in Y\}] \quad (3.6)$$

$$\text{s.t. } Ax = b \quad (3.7)$$

Bir stokastik programlama modelinin deterministik eşdeğeri ise aşağıdaki şekilde ifade edilebilmektedir:

$$\text{Min } z = \mathbf{c}^T \mathbf{x} + p(1)\mathbf{q}(1)\mathbf{y}(1) + \dots + p(\Omega)\mathbf{q}(\Omega)\mathbf{y}(\Omega) \quad (3.8)$$

$$\text{s.t. } \mathbf{Ax} = \mathbf{b} \quad (3.9)$$

$$\mathbf{T}(1)\mathbf{x} + \mathbf{Wy}(1) = \mathbf{h}(1), \quad (3.10)$$

$$\vdots \quad (3.11)$$

$$\mathbf{T}(\Omega)\mathbf{x} + \mathbf{Wy}(\Omega) = \mathbf{h}(\Omega), \quad (3.12)$$

$$\mathbf{x} \geq 0; \mathbf{y}(1), \dots, \mathbf{y}(\Omega) \geq 0 \quad (3.13)$$

İki kademeli ve tam sayılı stokastik programlama modelinin deterministik eşdeğeri aşağıdaki şekilde de ifade edilebilmektedir;

$$\text{Min } z = \mathbf{c}^T \mathbf{x} + Q(\mathbf{x}) \quad (3.14)$$

$$\text{s.t. } \mathbf{A} \cdot \mathbf{x} = \mathbf{b} \quad \mathbf{x} \in \mathbf{X}$$

Burada $Q(\mathbf{x})$ ikinci kademe probleminin değerini ifade etmektedir;

$$Q(\mathbf{x}) = E_{\xi} Q(\mathbf{x}, \xi(\omega)) \quad (3.15)$$

$$Q(\mathbf{x}, \xi(\omega)) = \min\{\mathbf{q}(\omega)\mathbf{y} \mid \mathbf{Wy} = \mathbf{h}(\omega) - \mathbf{T}(\omega)\mathbf{x}, \mathbf{y} \in \mathbf{Y}\} \quad (3.16)$$

İki kademeli stokastik programlama modellerinde olası farklı durumları karakterize etmek ve iki kademeli programların hangi sınıfı temsil ettiğini göstermek amacıyla a/b/c gösterimi kullanılmaktadır. Burada ilk harf birinci kademe karar değişkenlerini ifade etmektedir. Birinci kademe karar değişkenlerinin tamamı sürekli ise C, ikili ise B veya hem sürekli hem de ikili karar değişkenleri varsa (karışık tam sayılı problemleri) M harfi kullanılmaktadır. İkinci harf ise ikinci kademe karar değişkenlerini ifade eder ve benzer şekilde karakterize edilmektedir. Üçüncü harf, ξ senaryo değişkeninin dağılımını temsil etmektedir. ξ sonlu (ayrık) bir dağılıma sahipse D, sürekli rasal bir değişken olduğunda ise C harfi kullanılmaktadır (Laporte ve Louveaux, 1992). Laporte ve Louveaux (1992), tarafından önerilen gösterimi kullanarak, bu tez çalışması kapsamında geliştirilen modelin B/C/D sınıfına ait olduğunu söyleyebiliriz.

sonra alınan ikinci kademe kararları daha çok müdahaleye ve iyileştirmeye yönelik operasyonel kararlardır.



Şekil 2.2. Önerilen modelde deprem anı (t) öncesi ve sonrası kararlar

Geliştirilen modelin birinci kademesinde; deprem gerçekleşmeden önce olası can ve olası mal kayıplarını azaltmak amacıyla hem binaları hem de bağlantı yollarını güçlendirme kararları (hafifletme) verilmektedir. Modelin ikinci kademesinde ise; afet bölgelerindeki talebi karşılamak ve karşılanamayan talepleri minimize etmek amacıyla yardım merkezlerinden afet bölgelerine gönderilecek insani yardım malzemesi miktarına karar verilmektedir (müdahale). Modelin ikinci kademesinde ayrıca, depremde hasar görmüş binalar ve bağlantı yolları için yeniden inşa kararları (iyileştirme) verilmektedir. Sonuç olarak geliştirilen modelin temel amacı, belirli güçlendirme bütçeleri altında toplam yeniden inşa ve müdahale maliyetlerini minimize etmektir.

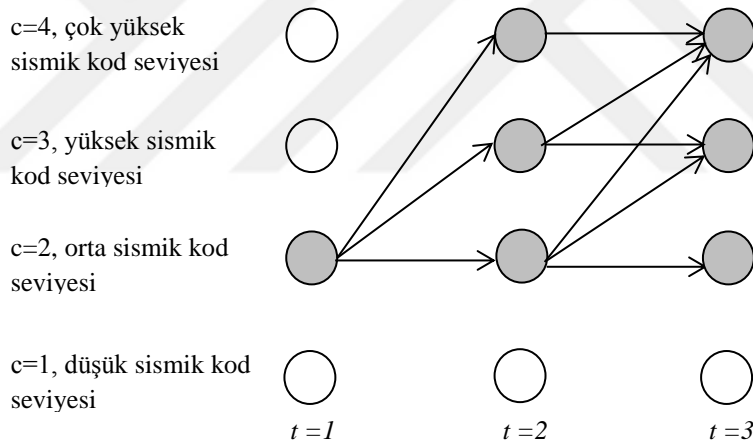
Gerçek hayatta, bir depremin bir t zamanında gerçekleşme olasılığı, şiddeti ve etkileri, $t+1$ zamanındakinden farklı olacaktır. Döyen ve Aras (2019) tarafından geliştirilen modelde, deprem öncesi ve sonrası kararları yer almasına rağmen depremin hangi zamanda gerçekleşeceği ele alınmamıştır. Geliştirilen modelde ise, Döyen ve Aras (2019) tarafından geliştirilen modelden farklı olarak, bu zamana bağlı olasılık değerleri de dikkate alınmıştır.

Ayrıca Döyen ve Aras (2019) tarafından geliştirilen modelde j bölgesindeki kurtarma merkezi ile k talep noktası arasındaki bağlantı yolu tek bir (j,k) hattı ile tanımlanmıştır. Geliştirilen modelde ise, gerçek hayatta olduğu gibi (j,k) arasındaki bütün yollar bir ulaşım ağı olarak nitelendirilerek hangi bağlantı yolunun hangi bölgesinde hasar olduğu da dikkate alınmıştır.

Döyen ve Aras (2019) tarafından geliştirilen modelde hafifletme ve müdahale aşamaları birlikte ele alınmıştır. Bu modelde deprem gerçekleşmeden önce güçlendirilecek bina ve bağlantı yollarının seçimi yapılmakta, deprem gerçekleştikten sonra da insani yardım malzemelerinin dağıtımı gerçekleştirilmektedir. Ancak modelin

amaç fonksiyonunda depremde hasar gören binaların ve bağlantı yollarının yeniden inşa maliyetini göz önünde bulunduran bir terim yer almamaktadır. Bu tez çalışmasında geliştirilen modelde ise hasar gören binaların ve bağlantı yollarının yeniden inşa maliyeti de göz önünde bulundurulmuştur. Böylece deprem ile ilgili kademelerin tümünde ortaya çıkabilecek tüm maliyetler dikkate alınarak bütünlük bir afet yönetimi modeli oluşturulması amaçlanmıştır.

Deprem gerçekleştiğinde olası can ve mal kayıplarını azaltmak amacıyla depremde hasar gören binaların ve bağlantı yollarının sayısını mümkün olduğunca azaltmak gerekir. Dolayısıyla deprem öncesinde etkili güçlendirme kararları alınmalıdır. Bu amaçla, geliştirilen modelde hangi binaların hangi sismik kod seviyesine yükseltileceğine (hangi derecede güçlendirileceğine) karar verilmektedir. Modelde kullanılan c sismik kod seviyeleri binaların dayanıklılık seviyelerini ifade etmektedir. Sismik kod seviyelerinin çalışma mekanizması aşağıda gösterilmiştir.



Şekil 3.3. Sismik kod seviyelerinin çalışma mekanizması

Bir binanın c sismik kod seviyesi arttıkça binanın dayanıklılık seviyesi artmaktadır. Örneğin $t=1$ döneminde k bölgesindeki m_k binasının $c=2$ orta dereceli sismik kod seviyesinde olduğunu varsayalım. Şekil 3.3, m_k binasının $t=1$ döneminde $c=2$ orta dereceli sismik kod seviyesinden, $t=2$ döneminde tek bir yüksek sismik kod seviyesine yükseltilebileceğini ya da aynı seviyede bırakılabileceğini göstermektedir. Şekil 3.3 ayrıca, $t=2$ döneminde m_k binasının dayanıklılık seviyesi, mevcut sismik kod seviyesinden daha aşağıda olacak şekilde değişmeyeceğini göstermektedir. Binanın dayanıklılık seviyesi ya artırılmalı ya da aynı seviyede bırakılmalıdır. Sonuç olarak $t=2$ döneminde m_k binasının sismik kod seviyesi $c=2$, $c=3$ veya $c=4$ olabilmektedir.

Modelde bina ve yolların güçlendirilmesi ve hasar görmesiyle ilgili yapılan varsayımlar:

- Hafifletme için ayrılan bütçe kısıtlıdır.
- Ulaşım ağı üzerindeki bazı hatlar hasara açık bağlantı yapılarıdır (köprü, tünel, viyadük vb. yapıları içeren yollar)
- Bir bağlantı güçlendirildiğinde herhangi bir deprem senaryosuna dayanıklı olacaktır.
- Bağlantı yollarındaki hasarlar birbirinden tamamen bağımsızdır.

Geliştirilen iki kademeli stokastik programlama modeline ait kümeler, indisler, parametreler ve karar değişkenlerine ait açıklamalar sırasıyla Çizelge 3.1, Çizelge 3.2, Çizelge 3.3 ve Çizelge 3.4’de verilmiştir.

Çizelge 3.1. Kümeler tablosu

N	Düğüm Kümeleri
A	Şebekede bulunan bağlantı yollarının kümesi
HY	Hasara açık bağlantı yollarının kümesi
TB	Talep bölgelerinin kümesi
KM	Kurtarma merkezlerinin kümesi
S	Deprem şiddeti senaryoları kümesi
W	Şebeke hasar durumları kümesi
C	Sismik kod seviyeleri kümesi
D	Binaların hasar seviyeleri kümesi
T	Dönemler kümesi
M_k	k bölgesindeki binalar kümesi

Çizelge 3.2. İndisler tablosu

j, k	Düğüm
s	Deprem şiddeti senaryoları
w	Şebeke hasar durumu
c	Sismik kod seviyeleri
d	Binaların hasar seviyeleri
t	Dönemler
m_k	k bölgesindeki binalar

Çizelge 3.3. Parametreler tablosu

v_{jk}^w	1, w şebeke hasar durumu altında hafifletme yapılmaksızın (j, k) bağlantı yolu hasara açık ise 0, diğer durumlar
$\lambda_{km_k}^d$	k bölgesindeki ve d hasar seviyesindeki m_k binasının talebi
$\gamma_{km_k}^{cs}$	s senaryosu altında ve c sismik kod seviyesinde olan k bölgesindeki m_k binasının d hasar seviyesinde olma olasılığı
n_{jk}^s	s senaryosu altında, kurtarma merkezi ile bir bölge arasında izin verilen maksimum taşıma kapasitesi
b_t	t dönemi için güçlendirme bütçesi
o_{jkt}^s	t döneminde ve s senaryosu altında hasar gören bağlantı yollarının yeniden inşasının (iyileştirme) maliyeti
$f_{km_k}^d$	k bölgesindeki ve d hasar seviyesindeki m_k binasının hasar görmesi durumunda yapının yeniden inşasının maliyeti
c_{jkt}^s	t döneminde ve s senaryosu altında j bölgesindeki kurtarma merkezinden k bölgesine gönderilen yardım malzemelerinin birim taşıma maliyeti
u_{jkt}	t döneminde (j, k) bağlantı yolunu güçlendirme maliyeti
$r_{km_k}^{cc'}$	t döneminde k bölgesindeki m_k binasını c sismik kod seviyesinden c' sismik kod seviyesine yükseltme maliyeti
a	Karşılanamayan yardım malzemesi talepleri için birim ceza maliyeti
p^w	w şebeke hasar durumunun gerçekleşme olasılığı
p_t^s	t döneminde s senaryosunun gerçekleşme olasılığı

Çizelge 3.4. Karar değişkenleri tablosu

$R_{km_k}^{cc'}$	1, t döneminde k bölgesindeki m_k binası c sismik kod seviyesinden c' sismik kod seviyesine yükseltildi ise 0, diğer durumlar
$J_{km_k}^c$	1, t döneminde k bölgesindeki m_k binası c sismik kod seviyesinde ise 0, diğer durumlar
U_{jkt}	1, t döneminde (j, k) bağlantı yolu güçlendirilirse 0, diğer durumlar
X_{jkt}^{sw}	t döneminde s deprem şiddeti senaryosu ve w şebeke hasar durumu altında, j bölgesinden k bölgesine gönderilen insani yardım malzemeleri miktarı
D_{kt}^s	t döneminde ve s senaryosu altında k bölgesindeki beklenen talep
Y_{kt}^{sw}	t döneminde, s senaryosu ve w şebeke hasar durumu altında, k bölgesinde ortaya çıkan karşılanamayan talep

Geliştirilen iki kademeli tam sayılı stokastik programlama modeli aşağıda bir bütün olarak verilmiştir. Modelin amaç fonksiyonuna ve her bir kısıtına ait açıklamalar da modelin altında detaylandırılmıştır.

$$\text{Min } \sum_t \sum_s \sum_w p_t^s [p^w (\sum_{(j,k) \in A} c_{jkt}^s \cdot X_{jkt}^{sw} + \sum_{k \in DP} a \cdot Y_{kt}^{sw})] + \sum_t \sum_s \sum_w p_t^s [p^w (\sum_{(j,k) \in VA} o_{jkt}^s \cdot v_{jk}^w \cdot (1 - \sum_{t'=1}^t U_{jkt'}))] + \sum_t \sum_s p_t^s [\sum_{k \in DP} \sum_{m_k} \sum_d \sum_c f_{km_k}^d \cdot \gamma_{km_k d}^{cs} \cdot J_{km_k t}^c] \quad (3.17)$$

$$J_{km_k(t-1)}^c = \sum_{c' \geq c} R_{km_k t}^{cc'} \quad \forall k, m_k, c, t \quad (3.18)$$

$$J_{km_k t}^c = \sum_{c' \leq c} R_{km_k t}^{c'c} \quad \forall k, m_k, c, t \quad (3.19)$$

$$\sum_{t=1}^T U_{jkt} \leq 1 \quad \forall (j,k) \in HY \quad (3.20)$$

$$\sum_k \sum_{m_k \in M_k} \sum_c \sum_{c' \geq c} r_{km_k t}^{cc'} \cdot R_{km_k t}^{cc'} + \sum_j \sum_k u_{jkt} \cdot U_{jkt} \leq B_t \quad \forall t \quad (3.21)$$

$$X_{jkt}^{sw} \leq (\sum_{t'=1}^t U_{jkt'} - v_{jk}^w + 1) \cdot n_{jk}^s \quad \forall (j,k) \in HY, s, t, w \quad (3.22)$$

$$X_{jkt}^{sw} \leq n_{jk}^s \quad \forall (j,k) \in A, s, t, w \quad (3.23)$$

$$D_{kt}^s = \sum_d \sum_{m_k \in M_k} \sum_c \lambda_{km_k}^d \cdot \gamma_{km_k d}^{cs} \cdot J_{km_k t}^c \quad \forall k \in TB, s, t \quad (3.24)$$

$$\sum_{j:(j,k) \in A} X_{jkt}^{sw} - \sum_{j:(k,j) \in A} X_{jkt}^{sw} = D_{kt}^s - Y_{kt}^{sw} \quad \forall k \in TB, s, t, w \quad (3.25)$$

$$\sum_{j:(j,k) \in A} X_{jkt}^{sw} - \sum_{j:(k,j) \in A} X_{jkt}^{sw} = 0 \quad \forall j \in N \setminus \{KM \cup TB\}, s, t, w \quad (3.26)$$

$$X_{jkt}^{sw}, Y_{kt}^{sw}, D_{kt}^s \geq 0 \quad \forall j, k, t, s, w \quad (3.27)$$

$$U_{jkt}, R_{km_k t}^{cc'}, J_{km_k t}^c \in \{0,1\} \quad \forall j, k, m_k, c, t \quad (3.28)$$

(3.17) numaralı denklemde verilen amaç fonksiyonu üç kısımdan oluşmaktadır. Birinci kısım yardım malzemelerinin dağıtımı ile karşılanamayan yardım malzemesi taleplerinin beklenen maliyetlerini minimize etmektedir. İkinci ve üçüncü kısım ise sırasıyla depremde hasar gören bağlantı yollarını ve binaları onarma maliyetlerinin toplamını minimize etmektedir.

$R_{km_k t}^{cc'}$ ikili karar değişkeni, t döneminde k bölgesindeki m_k binası c sismik kod seviyesinden c' sismik kod seviyesine yükseltirse 1, diğer durumda 0 değerini almaktadır. $J_{km_k t}^c$ ikili karar değişkeni ise t döneminde k bölgesindeki m_k binası c sismik

kod seviyesinde ise 1, diğer durumda 0 değerini almaktadır. (3.18) numaralı kısıt, $t-1$ döneminde k bölgesinde ve c sismik kod seviyesindeki bir m_k binasının, t döneminde tek bir sismik kod seviyesine yükseltilebileceğini ya da aynı seviyede bırakılabileceğini ifade etmektedir. Binanın dayanıklılık seviyesi ya artırılmalı ya da aynı seviyede bırakılmalıdır.

(3.19) numaralı kısıt, t döneminde bir m_k binanın dayanıklılık seviyesinin, mevcut sismik kod seviyesinden daha aşağıda olacak şekilde değişmeyeceğini ifade etmektedir. t döneminde c sismik kod seviyesinde bulunan bir bina ya daha düşük bir c' kod seviyesinden güçlendirilmiştir ya da güçlendirilmeden sabit kalmıştır.

(3.20) numaralı kısıt, bir (j,k) bağlantı yoluna yalnızca bir dönemde güçlendirme yapılmasını sağlamaktadır. Afet gerçekleşmeden önce, t döneminde (j,k) bağlantı yolu güçlendirilirse U_{jkt} karar değişkeni 1, diğer durumda 0 değerini almaktadır. HY ise hasara açık bağlantı yollarının (köprü, tünel, viyadük vb. yapıları içeren yollar) kümesini ifade etmektedir.

t döneminde k bölgesindeki m_k binasının c sismik kod seviyesinden c' sismik kod seviyesine güçlendirme maliyeti, $r_{km_k t}^{cc'}$ ve (j,k) bağlantı yolunun güçlendirme maliyeti u_{jkt} parametreleri ile gösterilmektedir. (3.21) numaralı kısıtta bu iki maliyet parametresi kullanarak, bina ve bağlantı yolu güçlendirme faaliyetlerinin toplam maliyetleri, her dönem için önceden tanımlanmış olan bir bütçe ile sınırlandırılmaktadır.

(3.22) numaralı kısıtta X_{jkt}^{SW} pozitif karar değişkeni, k düğümünün j düğümüne atanması durumunda, t döneminde s deprem şiddeti senaryosu ve w şebeke hasar durumu altında, j düğümünden k düğümüne gönderilen insani yardım malzemelerinin miktarını ifade etmektedir. w şebeke hasar durumu, bağlantı yollarının hasar görme durumunu ifade etmektedir. Şebekede m adet hasara açık bağlantı yolu varsa, şebeke hasar durumu 2^m şeklinde ifade edilmektedir. Örneğin şebekede üç tane hasara açık bağlantı yolu olduğunu varsayalım. Depremde bu bağlantı yolları yıkılabilir ya da yıkılmayabilir. Bu durumda 2^3 farklı şebeke hasar durumu ortaya çıkacaktır. Dolayısıyla alınacak taşıma kararları hem senaryoya hem de şebeke hasar durumuna bağlıdır. (3.22) numaralı kısıtın alabileceği dört farklı durum aşağıda açıklanmıştır;

- $X_{jkt}^{SW} \leq 0$; (j,k) bağlantı yolu hasar görebilir durumda ise ($v_{jk}^W = 1$) ve depremden önce güçlendirilmediyse ($U_{jkt} = 0$), j düğümünden k düğümüne hiçbir şekilde insani yardım malzemesi gönderilememektedir.

- $X_{jkt}^{sw} \leq n_{jk}^s$; (j,k) bağlantı yolu hasara açık durumda değilse ($v_{jk}^w = 0$), depremden önce güçlendirilmemiş olsa bile ($U_{jkt} = 0$), j düğümünden k düğümüne en fazla $(j-k)$ bağlantı yolunun kapasitesi kadar insani yardım malzemesi gönderilebilmektedir.
- $X_{jkt}^{sw} \leq n_{jk}^s$; (j,k) bağlantı yolu hasara açık durumda ise ($v_{jk}^w = 1$) ancak depremden önce güçlendirildiyse ($U_{jkt} = 1$), j düğümünden k düğümüne en fazla $(j-k)$ bağlantı yolunun kapasitesi kadar insani yardım malzemesi gönderilebilmektedir.
- $X_{jkt}^{sw} \leq 2 \cdot n_{jk}^s$; (j,k) bağlantı yolu hasara açık durumda olmamasına rağmen ($v_{jk}^w = 0$) depremden önce güçlendirildiyse ($U_{jkt} = 1$), j düğümünden k düğümüne en fazla (j,k) bağlantı yolunun kapasitesi kadar insani yardım malzemesi gönderilebilmektedir. Buradaki durumda, kapasitenin iki katı kadar gönderilebiliyor gibi görünse de aslında (3.23) numaralı kısıtta akış n_{jk}^s ile tekrar sınırlandırılmaktadır.

(3.23) numaralı kısıt, s senaryosu ve w şebeke hasar durumu altında j düğümünden k düğümüne akış olması durumunda, (j,k) bağlantı yoluna izin verilen maksimum taşıma kapasitesi n_{jk}^s 'nin aşılmamasını sağlamaktadır. A ise şebekede bulunan bağlantı yollarının kümesini ifade etmektedir.

Bir binanın hasar seviyesi, binanın afet öncesi sismik kod seviyesi ve gerçekleşen deprem senaryosuna göre belirlenebilmektedir. Dolayısıyla binaların hasar seviyesi sismik kod seviyelerinin yükseltilmesiyle azaltılabilmektedir. Binanın hasar seviyesi ve nüfusu ile de insani yardım malzemesi talebi yaklaşık olarak hesaplanabilmektedir. Sonuç olarak binaların güçlendirilmesiyle insani yardım malzemelerine olan talebin azalacağını söyleyebiliriz.

(3.24) numaralı kısıt, s senaryosunun gerçekleşmesi durumunda k bölgesindeki toplam yardım malzemesi talebini hesaplamaktadır. $Y_{km_k d}^{cs}$ parametresi s senaryosu altında ve c sismik kod seviyesinde olan k bölgesindeki m_k binasının d hasar seviyesinde olma olasılığını ifade etmektedir. $\lambda_{km_k}^d$ parametresi, k bölgesindeki m_k binasının d hasar seviyesinde olması durumunda, m_k binasında oluşan talebi ifade etmektedir. $J_{km_k t}^c$ ise, t döneminde k bölgesinde bulunan m_k binasının sismik kod seviyesini belirtmektedir. Sonuç olarak, k bölgesindeki beklenen toplam yardım malzemesi talebi D_{kt}^s , yukarıda açıklanan üç parametrenin çarpımına eşittir. TB talep bölgelerinin kümesini ifade etmektedir.

(3.25) numaralı kısıt, s senaryosu ve w şebeke hasar durumunun gerçekleşmesi durumunda, t döneminde j bölgesindeki toplam karşılanamayan Y_{kt}^{sw} yardım malzemelerinin talebini hesaplamaktadır.

(3.26) numaralı kısıt, şebeke denge kısıtıdır. t döneminde, s senaryosu ve w şebeke hasar durumu altında, (j,k) bağlantı yolu arasındaki geçiş düğümlerine gelen yardım malzemeleri miktarının o düğümden çıkan yardım malzemeleri miktarına eşit olmasını sağlamaktadır (gelen akış = çıkan akış). KM kurtarma merkezlerinin kümesini ifade etmektedir.

(3.27) ve (3.28) numaralı kısıtlar ise karar değişkenlerinin ikili ve pozitif değişkenler olduğunu ifade etmektedir.

Geliştirilen modelde şebeke hasar durumlarının dikkate alınması, senaryoların ve her bir bina için bina sismik kod seviyelerinin bulunmasından dolayı karar değişkeni sayısı oldukça fazladır ve buna bağlı olarak da problem oldukça büyük bir problem haline gelmektedir. Ticari çözücüler bu denli büyük ve karmaşık problemleri çözmekte zorlanmaktadır.

Stokastik programlama problemleri, özel yapılarından dolayı belirli ayrıştırma yöntemleri ile etkin bir şekilde çözülebilmektedir. L-Şekilli ve Tam Sayılı L-Şekilli yöntemleri, iki kademeli stokastik programlama problemlerinin çözümünde sıklıkla kullanılan ayrıştırma yöntemlerindedir. L-Şekilli yöntemi, problemi bir bütün olarak çözmek yerine, problemi ana problem ve alt problem olarak ayırıp, alt problemlerin çözümlerinden ana problemin çözümüne ulaşmayı amaçlamaktadır. Tam Sayılı L-Şekilli yöntemi ise L-Şekilli yönteminin bir dal-sınır mekanizması içerisinde uygulanmasıdır. Bir tür dal-kesme ve ayrıştırma yöntemi olan Tam Sayılı L-Şekilli yöntemi Laporte ve Louveaux (1992) tarafından önerilmiştir. Tam Sayılı L-Şekilli yöntemi, birinci kademe probleminin dal-sınır şemasını L-Şekilli yöntemin yinelemeli kesme düzlemleri prosedürü ile birleştirmektedir.

Bu tez çalışması kapsamında geliştirilen modeldeki tam sayılı karar değişkenlerinin varlığı nedeniyle de, modelin çözümünde Tam Sayılı L-Şekilli ayrıştırma yönteminden faydalanılmıştır.

Tam Sayılı L-Şekilli ayrıştırma yönteminde problem, yalnızca birinci kademe karar değişkenlerini içeren “ana problem (master problem)” ile birinci ve ikinci kademe karar değişkenlerini içerebilen “alt problemler (subproblem)” olarak ayrıştırılmaktadır. Ana problemin çözümünden elde edilen birinci kademe karar değişkenlerinin değerleri

sabitlenerek alt problemler oluşturulmaktadır. Dolayısıyla alt problemlerde karar değişkeni olarak sadece ikinci kademe karar değişkenleri bulunmaktadır ve alt problemlerin çözümünden ikinci kademe karar değişkenlerin değerleri elde edilmektedir. Ayrıca her senaryo için alt problemler ayrı ayrı oluşturulup çözülmektedir. Bu nedenle problem nispeten daha kolay çözülebilmektedir. Daha sonra, her adımda ana probleme çözüm uzayını kısıtlayan kısıtlar eklenir ve problem tekrar çözülür. Üst sınır ile alt sınır arasındaki göreceli farkın önceden belirlenmiş bir değerden daha küçük kalması sağlanana kadar adımlar tekrarlanır.

3.3. L-Şekilli Yöntemi

Aşağıdaki problem (Birge ve Louveaux,1999), iki kademeli bir stokastik programlama problemini temsil etmektedir. L-Şekilli yönteminde genellikle ilk aşamada x karar değişkeni gevşetilir ve ana problem çözülür. Ana problem birinci kademe karar değişkenlerinin değerlerini bulur ve bu bilgiyi sabitleyerek uygun bir çözüm bulması için alt probleme iletir. Alt problem de sabitlenmiş birinci kademe karar değerleri için uygun bir çözüm bulmaya çalışır. Alt problemin çözümü uygun değilse ana probleme ek kısıtlar (3.29) eklenir ve mevcut çözüm, çözüm uzayından çıkarılır. Bu kısıtlara fizibilite (olurluluk) kısıtları (kesmeleri) denir. Bu kısıtlar, ana problemde elde edilen çözümlerin alt problemler için uygun bir çözüm oluşturmadığı durumlarda kullanılır. Alt problemin uygun olduğu durumlarda ise, ikinci kademe probleminin amaç fonksiyonu değerini temsil eden θ değişkeni kullanılarak, optimal sonuca yaklaşmamızı sağlayan optimalite kısıtları (3.30) ana probleme eklenir. İteratif olarak devam eden bilgi alışverişi ve kısıtların eklenmesiyle, optimal sonuç bulunana kadar veya ana problemin çözüm uzayının tamamı taranana kadar algoritma devam eder.

$$\begin{aligned}
 \min \quad & cx + \theta \\
 \text{s. t.} \quad & Ax = b, \\
 & D_{\omega}x \geq d_{\omega}, \quad \omega = 1, \dots, r, \quad (3.29) \\
 & E_{\omega}x + \theta \geq e_{\omega}, \quad \omega = 1, \dots, s, \quad (3.30) \\
 & x \geq 0, \quad \theta \in R
 \end{aligned}$$

Bu tez kapsamında geliştirilen modelde ana problem, alt problemlere her zaman uygun çözümler vermektedir. Ana problemde elde edilen karar değişkenleri hiçbir

zaman alt problemleri etkilemeyeceği için fizibilite bozulmamaktadır, dolayısıyla Tam Sayılı L-Şekilli yönteminin modelimize uygulanmasında fizibilite kısıtları eklenmemiştir.

L-Şekilli yönteminin aşağıdaki probleme uygulama adımları aşağıda detaylı bir şekilde açıklanmıştır (Birge ve Louveaux, 1999).

$$\begin{aligned} \text{Min } z &= \mathbf{c}^T \mathbf{x} + E_{\xi} [\min \mathbf{q}(\omega)^T \mathbf{y}(\omega)] \\ \text{s.t. } \quad \mathbf{Ax} &= \mathbf{b} \\ \mathbf{T}(\omega)\mathbf{x} + \mathbf{Wy}(\omega) &= \mathbf{h}(\omega) \\ \mathbf{x} &\geq 0; \mathbf{y}(\omega) \geq 0 \end{aligned}$$

Başlangıç: $r = s = v = 0$ olarak tanımlayın.

Adım 1: $v = v + 1$ olarak güncelleyin ve aşağıdaki ana (master) problemi çözün.

$$\begin{aligned} \min \quad & cx + \theta \\ \text{s.t.} \quad & Ax = b, \\ & D_{\omega}x \geq d_{\omega}, \quad \omega = 1, \dots, r, \\ & E_{\omega}x + \theta \geq e_{\omega}, \quad \omega = 1, \dots, s, \\ & x \geq 0, \quad \theta \in R \end{aligned}$$

(x^v, θ^v) optimal çözüm olarak kabul edin. Çözümde (3.30) numaralı kısıt yoksa, $\theta^v = -\infty$ olarak kabul edilir ve x^v birinci kademe karar değişkeninin hesaplanmasında dikkate alınmaz.

Adım 2: x karar değişkeninin ikinci kademe karar değişkeni için de uygun bir çözüm olduğunu kontrol edin. Uygun bir çözüm değilse, ana probleme en az bir fizibilite kısıtı (3.29) ekleyin ve adım 1'e geri dönün. Aksi takdirde, adım 3'e gidin.

Adım 3: Her $\omega \in \Omega$ senaryosu için aşağıdaki alt problemi çözün.

$$\begin{aligned} \min w &= \mathbf{q}(\omega)\mathbf{y} \\ \text{s.t.} \quad \mathbf{Wy} &= \mathbf{h}(\omega) - \mathbf{T}(\omega) \mathbf{x}^v \\ \mathbf{y} &\geq \mathbf{0} \end{aligned} \quad (3.31)$$

π_{ω}^v değişkenini tanımlayın. Bu değişken (3.31) kısıtlarına karşılık gelen ikili (dual) değişkenini ifade etmektedir. Sırasıyla aşağıdaki (3.32) ve (3.33) numaralı eşitlikleri kullanarak (3.34) numaralı denkleme uygun olarak optimalite kısıtlarını oluşturun.

$$E_{s+1} = \sum_{\omega=1}^{\Omega} p_{\omega} \cdot (\pi_{\omega}^v) \cdot T(\omega) \quad (3.32)$$

$$e_{s+1} = \sum_{\omega=1}^{\Omega} p_{\omega} \cdot (\pi_{\omega}^v) \cdot h(\omega) \quad (3.33)$$

$$w^v = e_{s+1} + E_{s+1} \cdot x^v \quad (3.34)$$

$\theta^B \geq w^v$ ise çözümü optimal olarak kabul edin ve durun. Aksi takdirde $s = s+1$ olarak güncelleyin, ana probleme optimalite kısıtı (3.30) ekleyin ve adım 1'e geri dönün.

3.4. Önerilen Modelin Tam Sayılı L-Şekilli Yöntemi ile Çözülmesi

Tam Sayılı L-Şekilli yönteminin problemimize uygulama adımları aşağıda açıklanmıştır.

Başlangıç: Başlangıçta alt sınır $LB = -\infty$ ve üst sınır $UB(\bar{z}) = \infty$ olarak tanımlayın. $B=0$ iterasyon sayacı olarak tanımlayın

Adım 1: Düğüm listesinde bir düğüm seçin. Liste boş ise durun.

Tam Sayılı L-Şekilli yönteminde, listedeki her bir düğümün mevcut problemin bir formuna karşılık geldiği bir bekleyen düğüm listesi ile çalışılmaktadır (Listeş, 2005).

Adım 2: $B=B+1$ olarak güncelleyin ve ana problemi çözün.

Ana Problem:

$$\begin{aligned} \text{Min } \sum_t \sum_s \sum_w p_t^s [p^w \sum_{(j,k) \in VA} o_{jkt}^s \cdot v_{jk}^w \cdot (1 - \sum_{t'=1}^t U_{jkt'})] + \sum_t \sum_s p_t^s \\ [\sum_{k \in DP} \sum_{m_k} \sum_d \sum_c f_{km_k}^d \cdot \gamma_{km_k}^{cs} \cdot J_{km_k}^c] + \theta^B \end{aligned} \quad (3.35)$$

$$J_{km_k(t-1)}^c = \sum_{c' \geq c} R_{km_k}^{cc'} \quad \forall k, m_k, c, t$$

$$J_{km_k t}^c = \sum_{c' \leq c} R_{km_k}^{c'c} \quad \forall k, m_k, c, t$$

$$\sum_{t=1}^T U_{jkt} \leq 1 \quad \forall (j,k) \in HY$$

$$\sum_k \sum_{m_k \in M_k} \sum_c \sum_{c' \geq c} r_{km_k}^{cc'} \cdot R_{km_k}^{cc'} + \sum_j \sum_k u_{jkt} \cdot U_{jkt} \leq B_t \quad \forall t$$

$$U_{jkt}, R_{km_k}^{cc'}, J_{km_k}^c \in \{0,1\} \quad \forall j, k, m_k, c, t$$

θ^B değişkeni, ikinci kademe probleminin amaç fonksiyonu olan $Q(x)$ 'i temsil etmektedir.

Adım 3: İntegralite kısıtlarını kontrol edin. İntegralite bozulursa iki yeni düğüm ekleyin ve adım 1'e gidin

Adım 4: Her s senaryosu için $Q(J^B, R^B, U^B, s)$ alt problemleri çözün. Daha sonra (3.37)'da verilen amaç fonksiyonu $\hat{Q}(J^B, R^B, U^B)$ değerini hesaplayın.

Alt problem:

$$Q(J^B, R^B, U^B, s) = \mathbf{Min} \sum_{(j,k) \in A} c_{jkt}^s \cdot X_{jkt}^{sw} + \sum_{k \in DP} a \cdot Y_{kt}^{sw} \quad (3.36)$$

$$X_{jkt}^{sw} \leq \left(\sum_{t'=1}^t U_{jkt'} - v_{jk}^w + 1 \right) \cdot n_{jk}^s \quad \forall (j,k) \in HY, s, t, w$$

$$X_{jkt}^{sw} \leq n_{jk}^s \quad \forall (j,k) \in A, s, t, w$$

$$D_{kt}^s = \sum_d \sum_{m_k \in M_k} \sum_c \lambda_{km_k}^d \cdot Y_{km_k}^{cs} \cdot J_{km_k}^c \quad \forall k \in TB, s, t$$

$$\sum_{j:(j,k) \in A} X_{jkt}^{sw} - \sum_{j:(k,j) \in A} X_{jkt}^{sw} = D_{kt}^s - Y_{kt}^{sw} \quad \forall k \in TB, s, t, w$$

$$\sum_{j:(j,k) \in A} X_{jkt}^{sw} - \sum_{j:(k,j) \in A} X_{jkt}^{sw} = 0 \quad \forall j \in N \setminus \{KM \cup TB\}, s, t, w$$

$$X_{jkt}^{sw}, Y_{kt}^{sw}, D_{kt}^s \geq 0 \quad \forall j, k, t, s, w$$

$$\hat{Q}(J^B, R^B, U^B) = \sum_t \sum_s \sum_w p_t^s \cdot p^w \cdot Q(J^B, R^B, U^B, s) \quad (3.37)$$

Adım 5: $Q(J^B, R^B, U^B)$ ve $\mathbf{z}^B = \mathbf{c}^T \mathbf{x}^B + \hat{Q}(J^B, R^B, U^B)$ değerlerini hesaplayın. $\mathbf{z}^B < \bar{\mathbf{z}}$ ise $\bar{\mathbf{z}} \rightarrow \mathbf{z}^B$ olarak güncelleyin

Adım 6: $\theta^B \geq \hat{Q}(J^B, R^B, U^B)$ ise mevcut düğümü budayın ve adım 1'e geri dönün. Aksi takdirde, (3.38)'deki optimalite kısıtını ana probleme ekleyin ve adım 2'ye geri dönün.

3.4.1. Optimalite kısıtının oluşturulması

Alt problemin (3.31)'deki kısıtla aynı yapıda olan her bir kısıt için ikili (dual) değişkenleri tanımlanır. Önerilen modelde bu kısıtlar (3.22) - (3.24) kısıtlarıdır. Sırasıyla σ_{jkstw}^5 , σ_{jkstw}^6 ve σ_{kst}^7 bu kısıtlar için tanımlanan dual değişkenleri ifade etmektedir.

Alt problem:

$$\mathbf{Min} \sum_{(j,k) \in A} c_{jkt}^s \cdot X_{jkt}^{sw} + \sum_{k \in DP} a \cdot Y_{kt}^{sw}$$

$$X_{jkt}^{sw} \leq \left(\sum_{t'=1}^t U_{jkt'} - v_{jk}^w + 1 \right) \cdot n_{jk}^s \quad : \sigma_{jkstw}^5$$

$$X_{jkt}^{sw} \leq n_{jk}^s \quad : \sigma_{jkstw}^6$$

$$D_{kt}^s = \sum_d \sum_{m_k \in M_k} \sum_c \lambda_{km_k}^d \cdot \gamma_{km_k d}^{cs} \cdot J_{km_k t}^c \quad : \sigma_{kst}^7$$

$$\sum_{j:(j,k) \in A} X_{jkt}^{sw} - \sum_{j:(k,j) \in A} X_{jkt}^{sw} = D_{kt}^s - Y_{kt}^{sw}$$

$$\sum_{j:(j,k) \in A} X_{jkt}^{sw} - \sum_{j:(k,j) \in A} X_{jkt}^{sw} = 0$$

$$X_{jkt}^{sw}, Y_{kt}^{sw}, D_{kt}^s \geq 0$$

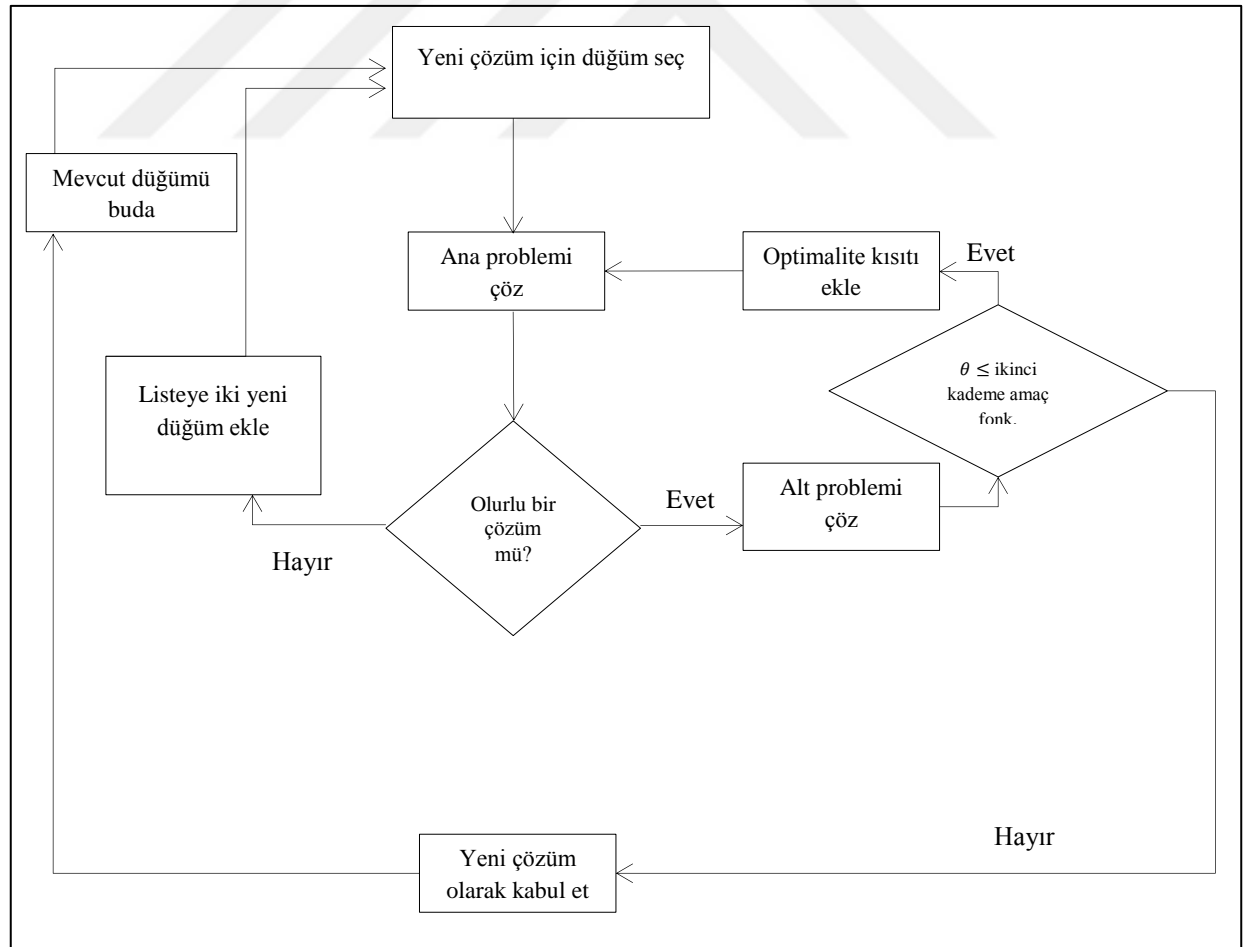
Dual deęişkenleri kullanılarak, (3.32)-(3.34) eşitlikleri doğrultusunda optimalite kısıtı (3.38) ařaęıdaki řekilde oluřturulur;

$$\theta \geq \sum_t \sum_s \sum_w \sum_{(j,k) \in VA} p_t^s \cdot p^w \cdot n_{jk}^s \cdot \sigma_{jktsw}^5 \cdot \left(\sum_{t'=1}^t U_{jkt'} \right)$$

$$+ \sum_t \sum_s \sum_{k \in DP} p_t^s \cdot \sigma_{kst}^7 \cdot \left(\sum_d \sum_{m_k \in M_k} \sum_c \lambda_{km_k}^d \cdot \gamma_{km_k d}^{cs} \cdot J_{km_k t}^c \right)$$

$$+ \sum_t \sum_s \sum_w p_t^s \left[p^w \left(\sum_{(j,k) \in VA} \sigma_{jktsw}^5 \cdot n_{jk}^s \cdot (1 - v_{jk}^w) + \sigma_{jktsw}^6 \cdot n_{jk}^s \right) \right] \quad (3.38)$$

Tam Sayılı L-řekilli ynteminin akıř řeması ařaęıda verilmiřtir;



řekil 3.3. Tam Sayılı L-řekilli ynteminin akıř řeması (Miller-Hooks ve ark., 2012)

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Bu bölümde deney tasarımı ve bu deney sonucunda elde edilen sonuçlara yer verilmiştir.

4.1. Deney Tasarımı

Bir önceki bölümde önerilen Tam Sayılı L-Şekilli yönteminin etkinliğini araştırmak amacıyla bir dizi rassal test problemleri geliştirilmiştir. Rastgele oluşturulan problemler talep noktaları sayısı $|K|$ ve düğüm sayısı $|N|$ bakımından birbirlerinden farklılık göstermektedir. Talep noktaları sayısı $|K|$ üç farklı değer ($|K| = 18, 20, 25$), düğüm sayısına ise yedi farklı değer atanmıştır ($|N| = 21, 23, 25, 28, 30, 35, 40$). Geliştirilen bütün problemler için kurtarma merkezlerinin sayısı $|J| = 3$, dönemler için ise $|T| = 5$ alınmıştır. Rassal olarak oluşturulan potansiyel kurtarma merkezleri ve talep noktalarının koordinatları $[0-100]$ aralığında düzgün dağılmaktadır. k bölgesindeki bina sayısı $|M_k|$, düzgün dağılan $[100-300]$ ve $[300-500]$ aralıklarına sahip iki farklı aralığa göre rassal oluşturulmaktadır. Her bir birim karşılanamayan yardım malzemesi talebi için birim ceza maliyeti $A = 1000$ olarak alınmıştır.

Binalar sismik kod seviyeleri bakımından *Sismik Kod-1 (SK-1)* ve *Sismik Kod-2 (SK-2)* olmak üzere iki grupta ele alınmıştır. *SK-1* grubunda, başlangıçta tüm binaların $c=1$ ya da $c=2$ sismik kod seviyelerinde olduğu durum incelenmiştir. *SK-2* grubunda, binaların $c=1, 2, 3, 4$ sismik kod seviyelerinde bulunabilecekleri durum incelenmiştir. Her iki grupta da herhangi bir k bölgesinde farklı sismik kod seviyelerinde olan binaların yüzdesi eşittir (*SK-1* grubu için %50 ve *SK-2* grubu için %25). Ayrıca herhangi bir binayı c seviyesinden $c + 1$ seviyesine güçlendirme maliyeti 10 birim olarak kabul edilmiştir.

Küçük, orta ve büyük olmak üzere üç farklı d hasar seviyesi olduğu varsayılmıştır ($d = 1, 2, 3$). Her bir bağlantı yolunun güçlendirme maliyeti, yolun yapısal özelliklerinden (bağlantı yolu üzerindeki köprü, viyadük, üstgeçit sayısı ve her köprü/viyadükün uzunluğu, yüksekliği, tasarımı, yaşı ve yük seviyeleri vb.) dolayı farklılık gösterdiğinden, u_{jk} parametreleri $[50-300]$ aralığında düzgün dağılıma göre üretilmektedir. Bağlantı yollarını yeniden inşasının ya da onarılmasının maliyeti o_{jkt}^s ise yine aynı mantıkla $[800-1000]$ aralığında düzgün dağılıma göre üretilmektedir.

Binaların yeniden inşasının ya da onarılmasının maliyeti $f_{km_k}^d$ ise d hasar seviyesine bağlı olarak değişmekte ve [40-80] aralığında rassal olarak üretilmektedir.

4 farklı deprem şiddeti senaryosu ele alınmıştır. Her bir senaryonun meydana gelme olasılıkları ve şiddetleri birbirinden farklıdır. Senaryoların meydana gelme olasılıkları $p = (0.50, 0.25, 0.15, 0.10)$ ve şiddetleri $\mu = (1, 1.1, 1.3, 1.8)$ olarak kabul edilmiştir. $\lambda_{km_k}^d$, k bölgesinde hasar seviyesi d olan m_k binasının talebini (etkilenen nüfusun oranını) ifade etmekteydi. Bu parametre $d = 1, 2$ ve 3 hasar seviyeleri için sırasıyla 10, 20 ve 40 olarak alınmıştır.

Geliştirilen matematiksel model C# dilinde ILOG CPLEX Concert Technology yardımıyla kodlanmıştır ve yazılan kodlar EK 1’de sunulmuştur. Önerilen model hem CPLEX ile ayrıştırılmadan (bütün olarak) hem de iki kademeli stokastik programlama modellerinin çözümü için kullanılan Tam Sayılı L-Şekilli yöntemi ile alt problemlere ayrıştırarak çözülmüştür. Karşılaştırma yapabilmek için farklı boyutlarda test problemleri geliştirilip çözülmüştür. Her bir problem Intel Core (TM) i7-10700 CPU 2.90 Ghz hızında 16 GB ara belleğe sahip, “Windows 10” işletim sistemi ile çalışan bilgisayarlarda gerçekleştirilmiştir. Bütün test problemleri 2 saat (7200 saniye) çözüm süresi ile sınırlandırılmıştır.

4.2. Deney Sonuçları

Performans ölçütü olarak Yüzde Sapma Değeri (YSD) kullanılmıştır. CPLEX’ in bulduğu en iyi alt sınır değerinden yüzde sapma değerleri hesaplanmıştır. CPLEX ve Tam Sayılı L-Şekilli yöntemi için YSD sırasıyla aşağıdaki şekilde hesaplanmaktadır.

$$YSD_{Cplex} = 100 * \left(\frac{Z_{üst}^{Cplex} - Z_{Alt}^{Cplex}}{Z_{Alt}^{Cplex}} \right) \quad (4.1)$$

$$YSD_{Tam\ Sayılı\ L-Şekilli} = 100 * \left(\frac{Z_{üst}^{Tam\ Sayılı\ L-Şekilli} - Z_{Alt}^{Cplex}}{Z_{Alt}^{Cplex}} \right) \quad (4.2)$$

Bu değerler sıfıra yaklaştıkça elde edilen sonuç da en iyi çözüme o kadar yaklaşmaktadır.

Geliştirilen test problemleri altı farklı grupta incelenmiştir. Bu gruplar talep noktası sayısı $|K|$ ve sismik kod seviyesi bakımından (*SK-1* ve *SK-2*) farklılık göstermektedir. $|K|$ değeri birinci ve dördüncü gruplar için 18, ikinci ve beşinci gruplar için 20, üçüncü ve altıncı gruplar için 25 alınmıştır. Her bir grupta geliştirilen test problemlerinde kurtarma merkezi sayısı ve dönem sayısı eşit alınmıştır. Bu problemlerde kurtarma merkezleri $|J| = 3$ ve dönemler de $|T| = 5$ alınmıştır.

Birinci grup test problemlerinde düğüm sayısı $|N| = 21, 25, 30, 35$ ve 40 olarak değişmektedir. Düğüm sayısı arttıkça problemin boyutu da artmaktadır. Ayrıca her bir talep bölgesindeki bina sayısı parametresinin CPLEX' in ve geliştirilen yöntemin performansı üzerindeki etkisini incelemek üzere [100-300] ve [300-500] aralıkları için ayrı ayrı test problemleri geliştirilmiştir.

Her dönem kullanılabilir güçlendirme bütçesi de performans üzerinde etkili olabilecek bir parametredir. Dolayısıyla bütçe parametresi (B_t) üç düzeyde (düşük, orta ve yüksek) incelenmiştir. Öncelikle, tüm hasar görebilir bağlantı yollarını ve tüm binaları en üst sismik kod seviyesine güçlendirmek için gerekli olan toplam bütçe dönem sayısına bölünerek b_{maks} değeri hesaplanmıştır. Düşük, orta ve yüksek bütçe değerlerini elde edebilmek için b_{maks} değeri sırasıyla 0,3, 0,6 ve 0,8 oranları ile çarpılmıştır.

Her bir problem için beş farklı problem örneği üretilmiş ve çözülmüştür. Böylece birinci, ikinci, dördüncü ve beşinci gruplar için 150'şer adet, üçüncü ve altıncı gruplar için 120'şer adet olmak üzere toplamda 840 örnek incelenmiştir.

Farklı boyuttaki test problemleri için CPLEX ve Tam Sayılı L-Şekilli yöntemi ile elde edilen YSD değerleri ve çözüm süreleri Çizelge 4.1 ve Çizelge 4.6 arasında sunulmuştur. Çizelgelerde ortalama YSD değerlerinin yanında parantez içerisinde belirtilen sayılar, çözülen beş farklı örnek içerisinde uygun çözüm veren örnek sayısını ifade etmektedir. Çözülen beş farklı örnek içerisinde herhangi bir uygun çözümün elde edilemediği durumlar için de uygun çözüm yok anlamına gelen "UÇY" ifadesi kullanılmıştır.

Birinci grup test problemlerinin sonuçları Çizelge 4.1'de verilmiştir.

Çizelge 4.1. Birinci grup test problemlerinin sonuçları ($|K|=18, SK-1$)

Problem Numarası	Düğüm Sayısı	Bina Sayısı Aralığı	Bütçe oranı	Ortalama YSD_{Cplex}	Süre (saniye)	Ortalama $YSD_{Tam Sayılı L-Şekilli}$	Süre (saniye)
1	21	[100, 300]	Yüksek	0 (5)	1500	12,4 (5)	7200
			Orta	0 (5)	3180	13,7 (5)	7200
			Düşük	8,6 (5)	7200	15,8 (5)	7200
2	21	[300, 500]	Yüksek	0 (5)	3840	13,6 (5)	7200
			Orta	0 (5)	7200	16,7 (5)	7200
			Düşük	7,9 (5)	7200	11,8 (5)	7200
3	25	[100, 300]	Yüksek	0 (5)	1860	5,8 (5)	7200
			Orta	0 (5)	4320	8,8 (5)	7200
			Düşük	10,8 (5)	7200	16,8 (5)	7200
4	25	[300, 500]	Yüksek	0 (5)	3600	12,9 (5)	7200
			Orta	3,9 (5)	7200	10,8 (5)	7200
			Düşük	16,9 (4)	7200	14,7 (5)	7200
5	30	[100, 300]	Yüksek	2,3 (5)	7200	14,9 (5)	7200
			Orta	14,4 (5)	7200	12,1 (5)	7200
			Düşük	UÇY	7200	8,6 (5)	7200
6	30	[300, 500]	Yüksek	UÇY	7200	15,7 (5)	7200
			Orta	UÇY	7200	13,9 (5)	7200
			Düşük	UÇY	7200	8,3 (5)	7200
7	35	[100, 300]	Yüksek	UÇY	7200	17,6 (4)	7200
			Orta	UÇY	7200	13,9 (5)	7200
			Düşük	UÇY	7200	9,4 (5)	7200
8	35	[300, 500]	Yüksek	UÇY	7200	17,2 (4)	7200
			Orta	UÇY	7200	15,6 (5)	7200
			Düşük	UÇY	7200	12,8 (5)	7200
9	40	[100, 300]	Yüksek	UÇY	7200	58,6 (2)	7200
			Orta	UÇY	7200	26,7 (3)	7200
			Düşük	UÇY	7200	18,7 (5)	7200
10	40	[300, 500]	Yüksek	UÇY	7200	UÇY	7200
			Orta	UÇY	7200	UÇY	7200
			Düşük	UÇY	7200	UÇY	7200

Birinci grup test problemlerinde toplam 150 örnek (10 adet problem, 3 farklı bütçe seviyesi için 5'er adet örnek) hem CPLEX hem de Tam Sayılı L-Şekilli yöntem ile çözülmüştür. İncelenen problem örnekleri içerisinde CPLEX 81, Tam Sayılı L-Şekilli yöntem ise 22 problemde uygun bir çözüm bulamamıştır.

Çizelge 4.1’de verilen sonuçlar incelendiğinde, özellikle $|N| = 25$ düğüme kadar CPLEX’ in optimal sonuç verdiği gözlemlenmiştir. Ancak düğüm sayısı arttıkça, CPLEX 7200 saniyede uygun bir çözüm veremezken Tam Sayılı L-Şekilli yöntemi uygun çözümler vermiştir. Dolayısıyla, problemin boyutu arttıkça geliştirilen iki kademeli stokastik programlama problemini alt problemlere ayrıştırarak çözmenin daha etkili sonuçlar verdiğini söyleyebiliriz.

Ayrıca, bina sayısının da çözümü zorlaştırdığı gözlemlenmiştir. Örneğin 5 ve 6 numaralı problem örneklerinde sadece bina sayısı bakımından farklılık göstermektedir. CPLEX [100-300] aralığında bina sayısı için 3600 saniyede optimal sonuca oldukça yakın bir çözüm verirken, [300-500] aralığında bina sayısı için 7200 saniyede çözüm verememektedir. 5 ve 6 numaralı problemler, Tam Sayılı L-Şekilli yöntemle her iki bina sayısı dağılımı için çözüm vermektedir ancak yine de [100-300] aralığında bina sayısı için daha iyi bir sonuç vermektedir. Problem 7-8 ve 9-10 için de aynı durumlar söz konusudur.

Çözümün etkileyen bir başka parametre ise bütçedir. Bütçe kısıtlandıkça CPLEX problemi çözmekte zorlanırken, Tam Sayılı L-Şekilli yönteminin daha kolay çözdüğü gözlemlenmiştir.

İkinci grup test problemlerinde talep noktası sayısı $|K| = 20$ ve düğüm sayısı $|N| = 23, 25, 30, 35$ ve 40 olarak değişmektedir. Talep noktası sayısı arttığı için bu gruptaki test problemleri, birinci gruptaki test problemlerine göre daha büyük problemlerdir. İkinci grup test problemlerinin sonuçları Çizelge 4.2’de verilmiştir.

Çizelge 4.2. İkinci grup test problemlerinin sonuçları ($|K|=20, SK-I$)

Problem Numarası	Düğüm Sayısı	Bina Sayısı Aralığı	Bütçe oranı	Ortalama YSD_{Cplex}	Süre (saniye)	Ortalama $YSD_{Tam\ Sayılı\ L-Şekilli}$	Süre (saniye)
11	23	[100, 300]	Yüksek	1,2 (5)	7200	12,3 (5)	7200
			Orta	5,7 (5)	7200	13,4 (5)	7200
			Düşük	7,3 (5)	7200	10,8 (5)	7200
12	23	[300, 500]	Yüksek	2,4 (5)	7200	10,7 (5)	7200
			Orta	8,3 (5)	7200	11,5 (5)	7200
			Düşük	UÇY	7200	9,7 (5)	7200
13	25	[100, 300]	Yüksek	2,7 (5)	7200	7,8 (5)	7200
			Orta	7,6 (5)	7200	10,1 (5)	7200
			Düşük	UÇY	7200	8,6 (5)	7200
14	25	[300, 500]	Yüksek	UÇY	7200	14,6 (5)	7200
			Orta	UÇY	7200	11,4 (5)	7200

			Düşük	UÇY	7200	12,7 (5)	7200
15	30	[100, 300]	Yüksek	0 (5)	4860	15,3 (5)	7200
			Orta	17,9 (4)	7200	12,7 (5)	7200
			Düşük	UÇY	7200	10,6 (5)	7200
16	30	[300, 500]	Yüksek	UÇY	7200	UÇY	7200
			Orta	UÇY	7200	13,4 (5)	7200
			Düşük	UÇY	7200	9,1 (5)	7200
17	35	[100, 300]	Yüksek	UÇY	7200	UÇY	7200
			Orta	UÇY	7200	14,3 (4)	7200
			Düşük	UÇY	7200	12,7 (5)	7200
18	35	[300, 500]	Yüksek	UÇY	7200	UÇY	7200
			Orta	UÇY	7200	13,8 (3)	7200
			Düşük	UÇY	7200	10,7 (3)	7200
19	40	[100, 300]	Yüksek	UÇY	7200	UÇY	7200
			Orta	UÇY	7200	UÇY	7200
			Düşük	UÇY	7200	UÇY	7200
20	40	[300, 500]	Yüksek	UÇY	7200	UÇY	7200
			Orta	UÇY	7200	UÇY	7200
			Düşük	UÇY	7200	UÇY	7200

İkinci grup test problemlerinde toplam 150 örnek (10 adet problem, 3 farklı bütçe seviyesi için 5'er adet örnek) hem CPLEX hem de Tam Sayılı L-Şekilli yöntem ile çözülmüştür. İncelenen örnekler içerisinde CPLEX 106, Tam Sayılı L-Şekilli yöntem ise 50 problemde uygun bir çözüm bulamamıştır.

İkinci grupta geliştirilen problemler talep noktası sayısı bakımından birinci grupta geliştirilen problemlerden farklılık göstermektedir. Birinci grupta $|K| = 18$ iken, ikinci grup test problemlerinde $|K| = 20$ alınmıştır. Bu da problemlerimizin boyutunu arttırmıştır. $|K| = 18$ ve düğüm sayısı $|N| = 30$ iken Tam Sayılı L-Şekilli yöntemi uygun bir çözüm verirken $|K| = 20$ olduğunda çözüm vermediği gözlemlenmiştir. Birinci grupta Tam Sayılı L-Şekilli yöntemi $|N| = 40$ düğüme kadar çözüm verirken, ikinci grupta $|N| = 40$ düğümde çözüm vermemektedir. Ayrıca Çizelge 4.2 'de verilen sonuçlar incelendiğinde, küçük problemlerde (11-13) problemi ayrıştırmadan çözümlerin daha iyi sonuçlar verdiğini söylenebilmektedir. Birinci grup test problemlerinde olduğu gibi bu grupta da bütçe kısıtlandıkça önerilen yöntem daha iyi sonuçlar vermektedir.

Üçüncü grup test problemlerinde talep noktası sayısı $|K| = 25$ ve düğüm sayısı $|N| = 28, 30, 35$ ve 40 olarak değişmektedir. Bu gruptaki test problemleri, ikinci

gruptaki test problemlerinden daha büyük problemlerdir. Üçüncü grup test problemlerinin sonuçları Çizelge 4.3 'te verilmiştir.

Çizelge 4.3. Üçüncü grup test problemlerinin sonuçları ($|K| = 25$, $SK-1$)

Problem Numarası	Düğüm Sayısı	Bina Sayısı Aralığı	Bütçe oranı	Ortalama YSD_{Cplex}	Süre (saniye)	Ortalama $YSD_{Tam Sayılı L-Şekilli}$	Süre (saniye)
21	28	[100, 300]	Yüksek	0 (5)	4980	5,4 (5)	7200
			Orta	9,7 (5)	7200	4,8 (5)	7200
			Düşük	17,9 (5)	7200	8,6 (5)	7200
22	28	[300, 500]	Yüksek	UÇY	7200	22,8 (5)	7200
			Orta	UÇY	7200	8,9 (5)	7200
			Düşük	UÇY	7200	5,8 (5)	7200
23	30	[100, 300]	Yüksek	1,8 (5)	7200	5,8 (5)	7200
			Orta	18,3 (3)	7200	5,3 (5)	7200
			Düşük	UÇY	7200	7,9 (5)	7200
24	30	[300, 500]	Yüksek	UÇY	7200	UÇY	7200
			Orta	UÇY	7200	9,3 (5)	7200
			Düşük	UÇY	7200	4,8 (5)	7200
25	35	[100, 300]	Yüksek	UÇY	7200	20,7 (4)	7200
			Orta	UÇY	7200	17,3 (5)	7200
			Düşük	UÇY	7200	10,8 (5)	7200
26	35	[300, 500]	Yüksek	UÇY	7200	UÇY	7200
			Orta	UÇY	7200	16,8 (4)	7200
			Düşük	UÇY	7200	13,7 (5)	7200
27	40	[100, 300]	Yüksek	UÇY	7200	32,2 (2)	7200
			Orta	UÇY	7200	24,7 (3)	7200
			Düşük	UÇY	7200	15,4 (4)	7200
28	40	[300, 500]	Yüksek	UÇY	7200	UÇY	7200
			Orta	UÇY	7200	36,8 (2)	7200
			Düşük	UÇY	7200	26,7 (2)	7200

Üçüncü grup test problemlerinde toplam 120 örnek (8 adet problem, 3 farklı bütçe seviyesi için 5'er adet örnek) hem CPLEX hem de Tam Sayılı L-Şekilli yöntem ile çözülmüştür. İncelenen problem örneklerinde CPLEX 97, Tam Sayılı L-Şekilli yöntem ise 30 problemde uygun bir çözüm bulamamıştır.

Üçüncü grupta geliştirilen test problemleri oldukça büyük problemlerdir. Çizelge 4.3 'te verilen sonuçlara göre CPLEX problemin boyutu büyüdükçe uygun çözüm bulmakta oldukça zorlanmaktadır. Tam Sayılı L-Şekilli yöntemi ise önemli bir

performans kaybı göstermemektedir. Öyleyse, problemin boyutu büyüdükçe Tam Sayılı L-Şekilli yöntem CPLEX' e oranla çok daha başarılı sonuçlar verdiğini söyleyebiliriz.

Bina envanterinin planlama periyodu başındaki sismik kod seviyeleri dağılımının çözüm üzerindeki etkisini incelemek adına birinci, ikinci ve üçüncü grup test problemlerinde *SK-1* yerine *SK-2* bina kod seviyesi dağılımı kullanılarak sırasıyla dördüncü, besinci ve altıncı grup test problemleri oluşturulmuştur. Birinci ve dördüncü, ikinci ve besinci, üçüncü ve altıncı grup test problemlerinde tek farklılık bina kod seviyesi dağılımıdır (*SK-1* ya da *SK-2*)

Dördüncü grup test problemlerinde talep noktası sayısı $|K| = 18$ ve düğüm sayısı $|N| = 21, 25, 30, 35$ ve 40 olarak değişmektedir. Dördüncü grup test problemlerinin sonuçları Çizelge 4.4 'te verilmiştir.

Çizelge 4.4. Dördüncü grup test problemlerinin sonuçları ($|K| = 18, SK-2$)

Problem Numarası	Düğüm Sayısı	Bina Sayısı Aralığı	Bütçe oranı	Ortalama YSD_{Cplex}	Süre (saniye)	Ortalama $YSD_{Tam Sayılı L-Şekilli}$	Süre (saniye)
29	21	[100, 300]	Yüksek	0 (5)	2700	12,3 (5)	7200
			Orta	0 (5)	4680	10,6 (5)	7200
			Düşük	10,8 (5)	7200	7,1 (5)	7200
30	21	[300, 500]	Yüksek	0 (5)	3840	13,7 (5)	7200
			Orta	3,3 (5)	7200	8,6 (5)	7200
			Düşük	12,4 (5)	7200	5,7 (5)	7200
31	25	[100, 300]	Yüksek	0 (5)	1860	7,9 (5)	7200
			Orta	5,8 (5)	7200	8,3 (5)	7200
			Düşük	UÇY	7200	8,6 (5)	7200
32	25	[300, 500]	Yüksek	4,6 (5)	3600	9,5 (5)	7200
			Orta	10,8 (4)	7200	7,6 (5)	7200
			Düşük	UÇY	7200	7,2 (5)	7200
33	30	[100, 300]	Yüksek	6,8 (5)	7200	13,8 (5)	7200
			Orta	UÇY	7200	10,5 (5)	7200
			Düşük	UÇY	7200	7,9 (5)	7200
34	30	[300, 500]	Yüksek	UÇY	7200	14,3 (4)	7200
			Orta	UÇY	7200	10,3 (5)	7200
			Düşük	UÇY	7200	7,8 (5)	7200
35	35	[100, 300]	Yüksek	UÇY	7200	14,8 (4)	7200
			Orta	UÇY	7200	13,1 (5)	7200
			Düşük	UÇY	7200	8,7 (5)	7200
			Yüksek	UÇY	7200	15,3 (4)	7200

36	35	[300, 500]	Orta	UÇY	7200	10,4 (5)	7200
			Düşük	UÇY	7200	9,6 (5)	7200
37	40	[100, 300]	Yüksek	UÇY	7200	45,6 (1)	7200
			Orta	UÇY	7200	36,8 (2)	7200
			Düşük	UÇY	7200	UÇY	7200
			Yüksek	UÇY	7200	UÇY	7200
38	40	[300, 500]	Orta	UÇY	7200	UÇY	7200
			Düşük	UÇY	7200	UÇY	7200

Dördüncü grup test problemlerinde toplam 150 örnek (10 adet problem, 3 farklı bütçe seviyesi için 5'er adet örnek) hem CPLEX hem de Tam Sayılı L-Şekli yöntem ile çözülmüştür. İncelenen problem örnekleri içerisinde CPLEX 96, Tam Sayılı L-Şekli yöntem ise 30 problemde uygun bir çözüm bulamamıştır.

Beşinci grup test problemlerinde talep noktası sayısı $|K| = 20$ ve düğüm sayısı $|N| = 23, 25, 30, 35$ ve 40 olarak değişmektedir. SK-2 grubu için beşinci grup test problemlerinin sonuçları Çizelge 4.5 'te verilmiştir.

Çizelge 4.5. Beşinci grup test problemlerinin sonuçları ($|K| = 20$, SK-2)

Problem Numarası	Düğüm Sayısı	Bina Sayısı Aralığı	Bütçe oranı	Ortalama YSD_{Cplex}	Süre (saniye)	Ortalama $YSD_{Tam\ Sayılı\ L-Şekli}$	Süre (saniye)
39	23	[100, 300]	Yüksek	4,3 (5)	7200	15,7 (5)	7200
			Orta	6,3 (5)	7200	13,7 (5)	7200
			Düşük	UÇY	7200	15,4 (5)	7200
40	23	[300, 500]	Yüksek	6,4 (5)	7200	11,5 (5)	7200
			Orta	11,8 (5)	7200	10,7 (5)	7200
			Düşük	UÇY	7200	12,8 (5)	7200
41	25	[100, 300]	Yüksek	6,8 (5)	7200	10,8 (5)	7200
			Orta	14,3 (5)	7200	9,4 (5)	7200
			Düşük	UÇY	7200	12,6 (5)	7200
42	25	[300, 500]	Yüksek	UÇY	7200	13,4 (5)	7200
			Orta	UÇY	7200	10,7 (5)	7200
			Düşük	UÇY	7200	12,8 (5)	7200
43	30	[100, 300]	Yüksek	UÇY	7200	UÇY	7200
			Orta	UÇY	7200	14,8 (4)	7200
			Düşük	UÇY	7200	9,7 (5)	7200
44	30	[300, 500]	Yüksek	UÇY	7200	UÇY	7200
			Orta	UÇY	7200	13,6 (5)	7200
			Düşük	UÇY	7200	8,8 (5)	7200

45	35	[100, 300]	Yüksek	UÇY	7200	UÇY	7200
			Orta	UÇY	7200	12,3 (4)	7200
			Düşük	UÇY	7200	10,8 (4)	7200
46	35	[300, 500]	Yüksek	UÇY	7200	UÇY	7200
			Orta	UÇY	7200	UÇY	7200
			Düşük	UÇY	7200	UÇY	7200
47	40	[100, 300]	Yüksek	UÇY	7200	UÇY	7200
			Orta	UÇY	7200	UÇY	7200
			Düşük	UÇY	7200	UÇY	7200
48	40	[300, 500]	Yüksek	UÇY	7200	UÇY	7200
			Orta	UÇY	7200	UÇY	7200
			Düşük	UÇY	7200	UÇY	7200

Beşinci grup test problemlerinde toplam 150 örnek (10 adet problem, 3 farklı bütçe seviyesi için 5'er adet örnek) hem CPLEX hem de Tam Sayılı L-Şekilli yöntem ile çözülmüştür. İncelenen problem örnekleri içerisinde CPLEX 120, Tam Sayılı L-Şekilli yöntem ise 61 problemde uygun bir çözüm bulamamıştır.

Altıncı grup test problemlerinde talep noktası sayısı $|K| = 25$ ve düğüm sayısı $|N| = 28, 30, 35$ ve 40 olarak değişmektedir. SK-2 için altıncı grup test problemlerinin sonuçları Çizelge 4.6'da verilmiştir.

Çizelge 4.6. Altıncı grup test problemlerinin sonuçları ($|K| = 25, SK-2$)

Problem Numarası	Düğüm Sayısı	Bina Sayısı Aralığı	Bütçe oranı	Ortalama YSD_{Cplex}	Süre (saniye)	Ortalama $YSD_{Tam\ Sayılı\ L-Şekilli}$	Süre (saniye)
49	28	[100, 300]	Yüksek	19,3 (3)	7200	13,7 (5)	7200
			Orta	UÇY	7200	11,1 (5)	7200
			Düşük	UÇY	7200	9,7 (5)	7200
50	28	[300, 500]	Yüksek	32,7 (3)	7200	14,2 (5)	7200
			Orta	UÇY	7200	13,6 (4)	7200
			Düşük	UÇY	7200	10,8 (5)	7200
51	30	[100, 300]	Yüksek	35,8 (2)	7200	14,8 (5)	7200
			Orta	UÇY	7200	7,6 (5)	7200
			Düşük	UÇY	7200	6,7 (5)	7200
52	30	[300, 500]	Yüksek	UÇY	7200	UÇY	7200
			Orta	UÇY	7200	15,7 (4)	7200
			Düşük	UÇY	7200	11,9 (5)	7200
53	35	[100, 300]	Yüksek	UÇY	7200	17,3 (5)	7200
			Orta	UÇY	7200	12,8 (5)	7200

			Düşük	UÇY	7200	10,7 (5)	7200
54	35	[300, 500]	Yüksek	UÇY	7200	UÇY	7200
			Orta	UÇY	7200	12,4 (3)	7200
			Düşük	UÇY	7200	UÇY	7200
55	40	[100, 300]	Yüksek	UÇY	7200	36,1 (2)	7200
			Orta	UÇY	7200	UÇY	7200
			Düşük	UÇY	7200	UÇY	7200
56	40	[300, 500]	Yüksek	UÇY	7200	UÇY	7200
			Orta	UÇY	7200	UÇY	7200
			Düşük	UÇY	7200	UÇY	7200

Altıncı grup test problemlerinde toplam 120 örnek (8 adet problem, 3 farklı bütçe seviyesi için 5'er adet örnek) hem CPLEX hem de Tam Sayılı L-Şekilli yöntem ile çözülmüştür. İncelenen problem örnekleri içerisinde CPLEX 112, Tam Sayılı L-Şekilli yöntem ise 47 problemde uygun bir çözüm bulamamıştır.

Sonuç olarak tüm gruplar incelendiğinde CPLEX ele alınan toplam 840 problem örneğinin 228 tanesi için (problemlerin %28'i) uygun bir çözüm elde edebilirken, önerilen yöntem 600 problem için (problemlerin %71'i) uygun bir çözüm elde edebilmiştir. Çizelge 4.4, 4.5 ve 4.6 'da verilen sonuçlar incelendiğinde, Tam Sayılı L-Şekilli yönteminin oldukça iyi ve hızlı sonuçlar verdiği gözlemlenmiştir. Özellikle düşük bütçeyle çözülen örneklerde Tam Sayılı L-Şekilli yöntemi CPLEX' e göre çok daha iyi sonuç vermiştir. Tüm test problemlerinde CPLEX' in uygun bir çözüm bulabildiği her örnek için Tam Sayılı L-Şekilli yöntem de uygun bir çözüm bulmuştur. Problem boyutu büyüdükçe CPLEX' e göre Tam Sayılı L-Şekilli yöntem görece daha avantajlı hale gelmektedir.

Aynı talep noktası sayısına sahip grupların sismik kod seviyesi bakımından (SK-1 ve SK-2) karşılaştırılabilirlikleri amacıyla, elde edilen sonuçlar Çizelge 4.7, Çizelge 4.8 ve Çizelge 4.9'da özetlenmiştir.

Çizelge 4.7. Birinci ve dördüncü grup test problemlerinin sonuçları

Düğüm Sayısı	Bina Sayısı Aralığı	Bütçe oranı	Sismik 1 Grubu		Sismik 2 Grubu	
			Ortalama YSD_{Cplex}	Ortalama $YSD_{Tam Sayılı L-Şekilli}$	Ortalama YSD_{Cplex}	Ortalama $YSD_{Tam Sayılı L-Şekilli}$
21	[100, 300]	Yüksek	0 (5)	12,4 (5)	0 (5)	12,3 (5)
		Orta	0 (5)	13,7 (5)	0 (5)	10,6 (5)
		Düşük	8,6 (5)	15,8 (5)	10,8 (5)	7,1 (5)
21	[300, 500]	Yüksek	0 (5)	13,6 (5)	0 (5)	13,7 (5)
		Orta	0 (5)	16,7 (5)	3,3 (5)	8,6 (5)
		Düşük	7,9 (5)	11,8 (5)	12,4 (5)	5,7 (5)
25	[100, 300]	Yüksek	0 (5)	5,8 (5)	0 (5)	7,9 (5)
		Orta	0 (5)	8,8 (5)	5,8 (5)	8,3 (5)
		Düşük	10,8 (5)	16,8 (5)	UÇY	8,6 (5)
25	[300, 500]	Yüksek	0 (5)	12,9 (5)	4,6 (5)	9,5 (5)
		Orta	3,9 (5)	10,8 (5)	10,8 (4)	7,6 (5)
		Düşük	16,9 (4)	14,7 (5)	UÇY	7,2 (5)
30	[100, 300]	Yüksek	2,3 (5)	14,9 (5)	6,8 (5)	13,8 (5)
		Orta	14,4 (5)	12,1 (5)	UÇY	10,5 (5)
		Düşük	UÇY	8,6 (5)	UÇY	7,9 (5)
30	[300, 500]	Yüksek	UÇY	15,7 (5)	UÇY	14,3 (4)
		Orta	UÇY	13,9 (5)	UÇY	10,3 (5)
		Düşük	UÇY	8,3 (5)	UÇY	7,8 (5)
35	[100, 300]	Yüksek	UÇY	17,6 (4)	UÇY	14,8 (4)
		Orta	UÇY	13,9 (5)	UÇY	13,1 (5)
		Düşük	UÇY	9,4 (5)	UÇY	8,7 (5)
35	[300, 500]	Yüksek	UÇY	17,2 (4)	UÇY	15,3 (4)
		Orta	UÇY	15,6 (5)	UÇY	10,4 (5)
		Düşük	UÇY	12,8 (5)	UÇY	9,6 (5)
40	[100, 300]	Yüksek	UÇY	58,6 (2)	UÇY	45,6 (1)
		Orta	UÇY	26,7 (3)	UÇY	36,8 (2)
		Düşük	UÇY	18,7 (5)	UÇY	UÇY
40	[300, 500]	Yüksek	UÇY	UÇY	UÇY	UÇY
		Orta	UÇY	UÇY	UÇY	UÇY
		Düşük	UÇY	UÇY	UÇY	UÇY

Çizelge 4.8. İkinci ve beşinci grup test problemlerinin sonuçları

Düğüm Sayısı	Bina Sayısı Aralığı	Bütçe oranı	Sismik 1 Grubu		Sismik 2 Grubu	
			Ortalama YSD_{Cplex}	Ortalama $YSD_{Tam Sayılı L-Şekilli}$	Ortalama YSD_{Cplex}	Ortalama $YSD_{Tam Sayılı L-Şekilli}$
23	[100, 300]	Yüksek	1,2 (5)	12,3 (5)	4,3 (5)	15,7 (5)
		Orta	5,7 (5)	13,4 (5)	6,3 (5)	13,7 (5)
		Düşük	7,3 (5)	10,8 (5)	UÇY	15,4 (5)
23	[300, 500]	Yüksek	2,4 (5)	10,7 (5)	6,4 (5)	11,5 (5)
		Orta	8,3 (5)	11,5 (5)	11,8 (5)	10,7 (5)
		Düşük	UÇY	9,7 (5)	UÇY	12,8 (5)
25	[100, 300]	Yüksek	2,7 (5)	7,8 (5)	6,8 (5)	10,8 (5)
		Orta	7,6 (5)	10,1 (5)	14,3 (5)	9,4 (5)
		Düşük	UÇY	8,6 (5)	UÇY	12,6 (5)
25	[300, 500]	Yüksek	UÇY	14,6 (5)	UÇY	13,4 (5)
		Orta	UÇY	11,4 (5)	UÇY	10,7 (5)
		Düşük	UÇY	12,7 (5)	UÇY	12,8 (5)
30	[100, 300]	Yüksek	0 (5)	15,3 (5)	UÇY	UÇY
		Orta	17,9 (4)	12,7 (5)	UÇY	14,8 (4)
		Düşük	UÇY	10,6 (5)	UÇY	9,7 (5)
30	[300, 500]	Yüksek	UÇY	UÇY	UÇY	UÇY
		Orta	UÇY	13,4 (5)	UÇY	13,6 (5)
		Düşük	UÇY	9,1 (5)	UÇY	8,8 (5)
35	[100, 300]	Yüksek	UÇY	UÇY	UÇY	UÇY
		Orta	UÇY	14,3 (4)	UÇY	12,3 (4)
		Düşük	UÇY	12,7 (5)	UÇY	10,8 (4)
35	[300, 500]	Yüksek	UÇY	UÇY	UÇY	UÇY
		Orta	UÇY	13,8 (3)	UÇY	UÇY
		Düşük	UÇY	10,7 (3)	UÇY	UÇY
40	[100, 300]	Yüksek	UÇY	UÇY	UÇY	34,6 (2)
		Orta	UÇY	UÇY	UÇY	UÇY
		Düşük	UÇY	UÇY	UÇY	UÇY
40	[300, 500]	Yüksek	UÇY	UÇY	UÇY	UÇY
		Orta	UÇY	UÇY	UÇY	UÇY
		Düşük	UÇY	UÇY	UÇY	UÇY

Çizelge 4.9. Üçüncü ve altıncı grup test problemlerinin sonuçları

Düğüm Sayısı	Bina Sayısı Aralığı	Bütçe oranı	Sismik 1 Grubu		Sismik 2 Grubu	
			Ortalama YSD_{Cplex}	Ortalama $YSD_{Tam Sayılı L-Şekilli}$	Ortalama YSD_{Cplex}	Ortalama $YSD_{Tam Sayılı L-Şekilli}$
21	[100, 300]	Yüksek	0 (5)	5,4 (5)	19,3 (3)	13,7 (5)
		Orta	9,7 (5)	4,8 (5)	UÇY	11,1 (5)
		Düşük	17,9 (5)	8,6 (5)	UÇY	9,7 (5)
22	[300, 500]	Yüksek	UÇY	22,8 (5)	32,7 (3)	14,2 (5)
		Orta	UÇY	8,9 (5)	UÇY	13,6 (4)
		Düşük	UÇY	5,8 (5)	UÇY	10,8 (5)
23	[100, 300]	Yüksek	1,8 (5)	5,8 (5)	35,8 (2)	14,8 (5)
		Orta	18,3 (3)	5,3 (5)	UÇY	7,6 (5)
		Düşük	UÇY	7,9 (5)	UÇY	6,7 (5)
24	[300, 500]	Yüksek	UÇY	UÇY	UÇY	UÇY
		Orta	UÇY	9,3 (5)	UÇY	15,7 (4)
		Düşük	UÇY	4,8 (5)	UÇY	11,9 (5)
25	[100, 300]	Yüksek	UÇY	20,7 (4)	UÇY	17,3 (5)
		Orta	UÇY	17,3 (5)	UÇY	12,8 (5)
		Düşük	UÇY	10,8 (5)	UÇY	10,7 (5)
26	[300, 500]	Yüksek	UÇY	UÇY	UÇY	UÇY
		Orta	UÇY	16,8 (4)	UÇY	12,4 (3)
		Düşük	UÇY	13,7 (5)	UÇY	UÇY
27	[100, 300]	Yüksek	UÇY	32,2 (2)	UÇY	36,1 (2)
		Orta	UÇY	24,7 (3)	UÇY	UÇY
		Düşük	UÇY	15,4 (4)	UÇY	UÇY
28	[300, 500]	Yüksek	UÇY	UÇY	UÇY	UÇY
		Orta	UÇY	36,8 (2)	UÇY	UÇY
		Düşük	UÇY	26,7 (2)	UÇY	UÇY

Çizelge 4.7, Çizelge 4.8 ve Çizelge 4.9 incelendiğinde, problemin büyüklüğü arttıkça SK-2 ile çözülen test problemlerinin daha iyi sonuçlar verdiği gözlemlenmiştir. SK-1 ile elde edemediğimiz bazı test problemlerinde bina envanteri SK-2 yapısıyla oluşturulduğunda uygun çözümler elde edilmiştir.

Sonuç olarak problemin boyutu arttıkça Tam Sayılı L-Şekilli yöntemin daha iyi sonuçlar verdiği gözlemlenmiştir. Yani karmaşıklık seviyesi arttıkça problemi alt problemlere ayrıştırarak çözümlerin daha etkili sonuçlar elde etmemizi sağlayacağını söyleyebiliriz.

4.3. Mükemmel Bilginin Beklenen Değeri ve Stokastik Çözümün Değeri

Bu tez çalışması kapsamında geliştirilen iki kademeli stokastik programlama modelinin çözülmesiyle elde edilen sonuçların etkinliğini değerlendirmek için mükemmel bilginin beklenen değeri (MBBD) ve stokastik çözümün değeri (SÇD) hesaplanmıştır. Literatürde bu terimler karşımıza sırasıyla “Expected Value of Perfect Information (EVPI)” ve “The value of the stochastic solution (VSS)” olarak çıkmaktadır.

Geliştirilen modelde her bir senaryo farklı bir şiddete ve farklı bir gerçekleşme olasılığına sahiptir. Bir t döneminde, iyileştirme faaliyetlerini gerçekleştirmeden önce karar vericilerin o t döneminde hangi senaryonun gerçekleşeceğini bildiğini varsayalım. O zaman karar vericiler, gerçekleşecek senaryoya göre iyileştirme kararları verecektir. Literatürde bu problem türü “*Wait and See (WS)*” yani “*Bekle ve Gör (BG)*” olarak ifade edilmektedir. Bekle ve Gör probleminde her senaryo için elde edilebilecek en iyi x değerleri üzerinden problemin beklenen değeri hesaplanmaktadır. BG probleminin formülasyonu aşağıdaki şekilde ifade edilebilmektedir.

$$BG = E_{\xi}[\min_x z(x, \xi)] \quad (4.3)$$

$$E_{\xi} z(\bar{x}(\xi), \xi) \quad (4.4)$$

Halbuki, bildiğimiz üzere karar vericiler hangi senaryonun gerçekleşeceği bilgisine önceden sahip değildir. Bu durumda yapılabilecek en iyi şey, iki kademeli stokastik programlama modelini (SP) çözerek beklenen maliyeti minimize etmektir. MBBD ise karar vericilerin mükemmel bilgiye sahip olmak için ödemek isteyeceği maksimum tutarı verir. MBBD aşağıdaki şekilde hesaplanmaktadır;

$$MBBD = SP - BG \quad (4.5)$$

Stokastik Çözümün Değeri (SÇD) ise problemi bir stokastik programlama modeli olarak tanımlamanın bize sağlayacağı getiriye gösterir. Problemi stokastik parametrelerin beklenen değerleri ile çözerek (beklenen değer problemi) ilk kademe karar değişkenleri elde edilir. Daha sonra ilk kademe karar değişkenlerinin değerlerini sabit tutarak tüm senaryolar ayrı ayrı çözülür. Her senaryo için elde edilen optimal amaç fonksiyonu değerleri ve senaryo olasılıkları kullanılarak optimal amaç fonksiyonu için

bir beklenen deęer elde edilir (EEV). EEV formülasyonu ařaęıda verilmiřtir. Burada $\bar{x}(\bar{\xi})$ beklenen deęer problemi ile elde edilen birinci kademe karar deęiřkenlerinin optimal çözümlerini ifade etmektedir.

$$EEV = E_{\xi}[z(\bar{x}(\bar{\xi}))\xi] \quad (4.6)$$

SÇD, stokastik parametrelerin beklenen deęerleri kullanılarak elde edilen çözümler (EEV) ile stokastik programlama (SP) sonucunu karřılařtırmaktadır. SÇD ařaęıdaki řekilde hesaplanmaktadır;

$$SÇD = EEV - SP \quad (4.7)$$

Önerilen stokastik programlama modelinin etkinlięini ölçmek için küçük bir problem üzerinde MBBD ve SÇD deęerleri hesaplanmıřtır. Rastgele oluřturulan problemde talep noktaları sayısı $|K| = 10$, kurtarma merkezleri $|J| = 3$, dönemler $|T| = 5$ ve düęüm sayısı $|N| = 20$ olarak alınmıřtır. k bölgesindeki bina sayısı $|M_k|$, düzgün daęılan $[100, 300]$ aralıęa göre rassal oluřturulmaktadır. Her bir karřılanamayan yardım malzemesi talepleri için birim ceza maliyeti $A = 1000$ olarak alınmıřtır. Problemde 4 farklı sismik kod seviyesi ele alınmıřtır ($c = 1, 2, 3, 4$) ve herhangi bir c seviyesinden $c + 1$ seviyesine güçlendirme maliyeti 10 birim olarak kabul edilmiřtir. Ve son olarak küçük, orta ve büyük olmak üzere üç farklı d hasar seviyesi olduęu varsayılmıřtır ($d = 1, 2, 3$).

Sonuçlara göre geliřtirilen problemin stokastik programlama ile çözümlerinden toplam maliyet $SP = 47212$ elde edilmiřtir. Hangi senaryonun gerçekleřeceęi bilindięi varsayıldıęında ise toplam maliyet $BG = 36676$ çıkmaktadır. Dolayısıyla karar vericiler mükemmel bilgiye ulařmak için $MBBD = SP - BG = 47212 - 36676 = 12535$ birim maliyete katlanmalıdır. Yani problemin deterministik deęil de rassal bir yapıya sahip olması 12535 birim maliyete sebep olmaktadır. Bu da yüzde olarak % 34 oranında bir artış demektir.

Stokastik parametrelerin beklenen deęerlerini kullanarak elde edilen çözümler yerine stokastik programlama çözümlerini kullanmak ise karar vericilere SÇD kadar kazanç saęlamaktadır. $SÇD = EEV - SP = 78523 - 47212 = 31311$ birim maliyet. Yani rassal bir problemi ilk akla geleceęi řekilde rassal parametrelerin beklenen deęerlerini

olarak çözmek yerine stokastik programlama ile modelleyerek çözmek 31311 birimlik bir avantaj sağlamaktadır. Bu da yüzde olarak yaklaşık % 40 oranında bir avantaj demektir.



5. SONUÇLAR VE ÖNERİLER

Bu tez çalışması kapsamında, afet öncesi ve sonrası kararların birbirleri arasındaki etkileşimlerini dikkate alan yeni bir iki kademeli tam sayılı stokastik programlama modeli önerilmiştir. Modelde hafifletme, müdahale ve iyileştirme aşamaları bir bütün olarak ele alınmıştır. Modelin ilk kademesinde, literatürdeki diğer çalışmalardan farklı olarak, olası can ve mal kayıplarını azaltmak amacıyla, depremden önce hem binalara hem de bağlantı yollarına ait güçlendirme kararları (hafifletme) verilmektedir. Modelin ikinci kademesinde ise depremden sonra talep edilen insani yardım malzemelerinin etkin bir şekilde dağıtımı ile ilgili kararlar (müdahale) ve hasar gören yol ve binaların yeniden inşa kararları (iyileştirme) verilmektedir. Modelin amaç fonksiyonu ise toplam bina ve bağlantı yollarının iyileştirilmesi, yardım malzemelerinin taşınması ve karşılanamayan ihtiyaç malzemesi talebi maliyetlerinin toplamının en aza indirilmesi olarak belirlenmiştir.

Geliştirilen bu iki kademeli tam sayılı stokastik programlama modeli; bina ve yol güçlendirme kararlarını içeren ve tüm afet yönetimi aşamalarını beraber ele alan ilk çalışmadır.

Önerilen modelde; özellikle, her bir bina için ayrı ayrı güçlendirme kararlarının verilebiliyor olması ayrıca deprem sonrası ulaşım şebekesinde ortaya çıkması muhtemel her bir şebeke durumunun modele dahil edilmesi modelin boyutunu oldukça büyütülmektedir. Modelin bir bütün olarak ticari bir çözücü de çözülmesi çok zor olabilmektedir. Bu nedenle modeli ana ve alt problem olmak üzere iki parçaya ayırıştırıp, bu problemleri yinelemeli şekilde art arda çözerek sonuca ulaşmayı amaçlayan bir yöntem olan Tam Sayılı L-Şekilli yöntem kullanılmıştır.

Rastgele oluşturulmuş test problemleri ile önerilen matematiksel model hem ayırıştırılmadan CPLEX ile hem de Tam Sayılı L-Şekilli yöntem ile ayırıştırılarak çözülmüştür. CPLEX ele alınan toplam 840 problem örneğinin 228 tanesi için (problemlerin %28'i) uygun bir çözüm elde edebilirken, önerilen yöntem 600 problem için (problemlerin %71'i) uygun bir çözüm elde edebilmiştir. Üstelik bulunan çözümler oldukça makul YSD değerlerine sahiptir yani optimale yakın çözümlerdir. Beklenildiği üzere, problemlerin boyutu arttıkça, CPLEX uygun çözüm bulmakta zorlanırken, Tam Sayılı L-Şekilli yöntem yine yüksek oranda uygun çözüm elde edebilmektedir. Yani, problem boyutu büyüdükçe önerilen yöntemin CPLEX'e karşı görece avantajı daha fazla olmaktadır.

Önerilen iki kademeli tam sayılı stokastik programlama modeli gerçek hayatta gerçek bir afet yönetimi problemini çözmek için kullanılabilir yapıdadır. Geliştirilen test problemlerinde kullanılan parametreler (düğüm sayıları, senaryolar, maliyetler vb.) İstanbul için gerçekçi değerler olarak seçilmiştir. Böylece, bu model dünyanın deprem tehlikesi en yüksek bölgelerinden biri olan İstanbul için afet yönetimi planlama aracı olarak kullanılabilir.

Geliştirilen model, bağlantı yollarının birbirinden bağımsız olmadığı durum ve kurtarma merkezlerinden afet bölgelerine ulaşım süresi de dikkate alınarak genişletilebilir. Binaların ve bağlantı yollarının güçlendirilmesi için kullanılacak bütçeler farklı kaynaklardan sağlanıyorsa bu durumda; binalar için ayrı bağlantı yolları için ayrı bütçeler belirlenip daha gerçekçi kararlar verilebilir. Son olarak, önerilen modelde birinci kademe problemine (ana problem) ait bütçe kısıtının çözümü zorlaştırdığı gözlemlenmiştir. Bu kısıt Lagrange Gevşetmesi yöntemi ile gevşetilerek ana problem çözülürse ana problem çok daha hızlı çözülebilir. Neticede daha fazla iterasyon yapılabileceği için daha iyi çözümler elde edilebileceği düşünülmektedir.

KAYNAKLAR

- Alem, D., Clark, A., Moreno, A., 2016, Stochastic network models for logistics planning in disaster relief, *European Journal of Operational Research*, 255, 187–206.
- Augusti, G., Borri, A., Ciampoli, M., 1994, Optimal allocation of resources in reduction of the seismic risk of highway networks, *Engineering Structures*, 16 (7), 485–97.
- Afet ve Acil Durum Yönetimi Başkanlığı (AFAD), 2018, Türkiye’de Afet Yönetimi ve Doğa Kaynaklı Afet İstatistikleri, 44-48, İstanbul.
- Bai, X., 2016, Two-stage multi objective optimization for emergency supplies allocation problem under integrated uncertainty, *Mathematical Problems in Engineering*.
- Balcik, B., Beamon, B. M., 2008, Facility location in humanitarian relief, *International Journal of Logistics Research and Applications*, 11 (2), 101–121.
- Barbarosoglu, G., Arda, Y., 2004, A two-stage stochastic programming framework for transportation planning in disaster response, *Journal of the Operational Research Society*, 55 (1), 43–53.
- Behl, A., Dutta, P., 2019, Humanitarian supply chain management: a thematic literature review and future directions of research, *Annals of Operations Research*, 283(1), 1001-1044.
- Birge, J.R., Louveaux, F., 1999, Introduction to Stochastic Programming, Second Edition, Mikosch, T., Resnick, S., Robinson, S.M., *Springer Series in Operations Research and Financial Engineering*, 181-262.
- Bozorgi-Amiri, A., Jabalameli, M. S., Alinaghian, M., Heydari, M., 2012, A modified particle swarm optimization for disaster relief logistics under uncertain environment, *International Journal of Advanced Manufacturing Technology*, 60, 357–371.
- Caunhye, A. M., Nie, X., Pokharel, S., 2012, Optimization models in emergency logistics: A literature review, *Socio-Economic Planning Sciences*, 46 (1), 4-13.
- Chang, M. S., Tseng, L., Chen, J. W., 2007, A scenario approach for the flood emergency logistics preparation problem under uncertainty, *Transportation Research Part E*, 43, 737–754.
- Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE transactions on evolutionary computation*, 6(2), 182-197.
- Dodo, A., Davidson, R. A., Xu N., Nozick, L. K., 2005, Optimizing regional earthquake mitigation investment strategies, *Earthquake Spectra*, 21 (2), 305–27.
- Dodo, A., Davidson, R. A., Xu, N., Nozick, L. K., 2007, Application of regional earthquake mitigation optimization, *Computers & Operations Research*, 34 (8), 2478-2494.
- Döyen, A., Aras, N., Barbarosoglu, G., 2012, A two-echelon stochastic facility location model for humanitarian relief logistics, *Optimization Letters*, 6, 1123–1145.
- Döyen, A., Aras, N., 2019, An integrated disaster preparedness model for retrofitting and relief item transportation. *Networks and Spatial Economics*, 19(4), 1031-1068.
- Döyen, A., 2018, *Konya Teknik Üniversitesi Fen Bilimleri Enstitüsü*, “Yüksek Lisans Ders Notları”, Konya.
- Du, L., Peeta, S., 2014, A stochastic optimization model to reduce expected post-disaster response time through pre-disaster investment decisions, *Networks and Spatial Economics*, 14 (2).

- Elçi, Ö., Noyan, N., 2018, A chance-constrained two-stage stochastic programming model for humanitarian relief network design, *Transportation research part B: methodological*, 108, 55-83.
- Fan, Y., Liu, C., Lee, R., Kiremidjian, A. S., 2010, Highway network retrofit under seismic hazard. *Journal of Infrastructure Systems*, 16(3), 181-187.
- Faturechi, R., Miller-Hooks, E., 2015, Measuring the performance of transportation infrastructure systems in disasters: A comprehensive review. *Journal of infrastructure systems*, 21(1), 04014025.
- Garrido, R. A., Lamas, P., Pino, F. J., 2015, A stochastic programming approach for floods emergency logistics. *Transportation research part E: logistics and transportation review*, 75, 18-31.
- Görmez, N., Köksalan, M., Salman, F. S., 2011, Locating disaster response facilities in Istanbul. *Journal of the Operational Research Society*, 62(7), 1239-1252.
- Grass, E., Fischer, K., 2016, Two-stage stochastic programming in disaster management: A literature survey. *Surveys in Operations Research and Management Science*, 21(2), 85-100.
- Gupta, S., Starr, M. K., Farahani, R. Z., Matinrad, N., 2016, Disaster management from a POM perspective: Mapping a new domain. *Production and Operations Management*, 25(10), 1611-1637.
- Günneç, D., Salman, F. S., 2006, A two-stage multi-criteria stochastic programming model for location of emergency response and distribution centers, *In: Euro Winter Institute on Location and Logistics*, Estoril, Portugal, 35, 209-227.
- Günneç, D., Salman, F. S., 2011, Assessing the reliability and the expected performance of a network under disaster risk, *OR Spectrum*, 33 (3), 499-523.
- Hong, X., Lejeune, M. A., Noyan, N., 2015, Stochastic network design for disaster preparedness, *IIE Transactions*, 47 (4), 329-357.
- Hu, S., Han, C., Dong, Z. S., Meng, L., 2019, A multi-stage stochastic programming model for relief distribution considering the state of road network, *Transportation Research Part B: Methodological*, 123, 64-87.
- Jia, H., Ordonez, F., Dessouky, M. M., 2007, A modeling framework for facility location of medical services for large-scale emergencies, *IIE Transactions*, 39, 41-55.
- Irohara, T., Kuo, Y. H., Leung, J. M., 2013, From preparedness to recovery: a tri-level programming model for disaster relief planning, *In International Conference on Computational Logistics*, Springer, Berlin, Heidelberg, 213-228.
- Kadioğlu, M., 2011, Afet Yönetimi. T.C. Marmara Belediyeler Birliği Yayını, İstanbul.
- Kalkınma Bakanlığı, 2014, "Afet Yönetiminde Etkinlik.", Özel İhtisas Komisyonu Raporu, Ankara.
- Laporte, G., Louveaux, F., 1992, The integer L-shaped method for stochastic integer programs with complete recourse, *Operations Research Letters*, 133-142.
- Laporte, G., Louveaux, F., Van Hamme, L., 1994, Exact solution to a location problem with stochastic demands, *Transportation Science*, 95- 103.
- Laumanns, M., Thiele, L., Zitzler, E., 2006, An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method, *European Journal of Operational Research*, 169(3), 932-942.
- Legg, M., Davidson, R. A., Nozick, L. K., 2013, Optimization-based regional hurricane mitigation planning, *Journal of infrastructure systems*, 19(1), 1-11.
- Li, A. C. Q., Nozick, L., Xu, N. ve Davidson, R., 2012, Shelter location and transportation planning under hurricane conditions, *Transportation Research Part E: Logistics and Transportation*, 48, 715-729.

- Li, L., Jin, M., Zhang, L., 2011, Sheltering network planning and management with a case in the Gulf Coast region, *International Journal of Production Economics*, 131, 431–440.
- Liao, T. Y., Hu, T. Y., Ko, Y. N., 2018, A resilience optimization model for transportation networks under disasters, *Natural hazards*, 93(1), 469-489.
- Listeş, O., 2005, A generic stochastic model for supply-and-return network design, *Computers & Operations Research*, 417-442.
- Liu, C., Fan, Y., Ordóñez, F., 2009, A two-stage stochastic programming model for transportation network protection, *Computers & Operations Research*, 36 (5), 1582–1590.
- Mete, H. O., Zabinsky, Z. B., 2010, Stochastic optimization of medical supply location and distribution in disaster management, *International Journal of Production Economics*, 126, 76–84.
- Miller-Hooks, E., Zhang, X., Fatourchi, R., 2012, Measuring and maximizing resilience of freight transportation networks, *Computers & Operations Research*, 39 (7), 1633–1643.
- Mohammadi, R., Ghomi, S. F., Jolai, F., 2016, Prepositioning emergency earthquake response supplies: A new multi-objective particle swarm optimization algorithm. *Applied Mathematical Modelling*, 40(9-10), 5183-5199.
- Mohaymany, A. S., Nikoo, N., 2020, Designing Large-Scale Disaster Response Routes Network in Mitigating Earthquake Risk Using a Multi-Objective Stochastic Approach. *KSCCE Journal of Civil Engineering*, 24(10), 3050-3063.
- Monzón, J., Liberatore, F., Vitoriano, B., 2020, A Mathematical Pre-Disaster Model with Uncertainty and Multiple Criteria for Facility Location and Network Fortification. *Mathematics*, 8(4), 529.
- Moreno, A., Alem, D., Ferreira, D., 2016, Heuristic approaches for the multiperiod location-transportation problem with reuse of vehicles in emergency logistics. *Computers & Operations Research*, 69, 79-96.
- Parragh, S.N., Tricoire, F., Gutjahr, W. J., 2021, A branch-and-Benders-cut algorithm for a bi-objective stochastic facility location problem, *OR Spectrum*, 1-41.
- Peeta, S., Salman, F., Gunec, D., Viswanath, K., 2010, Pre-disaster investment decisions for strengthening a highway network, *Computers & Operations Research*, 37 (10), 1708–1719.
- Ramezani, R., Ghorbani, M., 2020, Stochastic optimization for the carrier selection problem in humanitarian relief, *Scientia Iranica*.
- Rath, S., Gendreau, M., Gutjahr, W. J., 2015, Bi-Objective stochastic programming models for determining depot locations in disaster relief operations, *International Transactions in Operational Research*, 23 (6), 1–27.
- Rawls, C. G., Turnquist, M. A., 2010, Pre-positioning of emergency supplies for disaster response. *Transportation research part B: Methodological*, 44(4), 521-534.
- Rawls, C. G., Turnquist, M. A., 2011, Pre-positioning planning for emergency response with service quality constraints. *OR spectrum*, 33(3), 481-498.
- Rennemo, S. J., R., K. F., Hvattum, L. M., Tirado, G., 2014, A three-stage stochastic facility routing model for disaster response planning. *Transportation research part E: logistics and transportation review*, 62, 116-135.
- Salman, F. S., Yücel, E., 2015, Emergency facility location under random network damage: Insights from the Istanbul case. *Computers & Operations Research*, 62, 266-281.

- Santoso, T., Ahmed, S., Goetschalckx, M., Shapiro, A., 2005, A stochastic programming approach for supply chain network design under uncertainty, *European Journal of Operational Research*, 167, 96-115.
- Serra, D., Marianov, V., 1998, The p-median problem in a changing network: The case of Barcelona, *Location Science*, 6, 383–394.
- Şahin, N., 2009, “Afet Yönetimi ve Acil Yardım Planları.” TMMOB İzmir Kent Sempozyumu, 131–42.
- Tanyaş, M., Günalay Y., Aksoy L., Küçük B., 2013a, “Afet Lojistik Yönetiminde Rize İline Yönelik Yeni Model Önerisi.” In II. Rize Kalkınma Sempozyumu.
- Tofighi, S., Torabi, S. A., Mansouri, S. A., 2016, Humanitarian logistics network design under mixed uncertainty, *European Journal of Operational Research*, 250, 239–250.
- Üstün, A. K., Anagün, A. S., 2015, Multi-objective mitigation budget allocation problem and solution approaches: The case of İstanbul, *Computers & Industrial Engineering*, 81, 118-129.
- Van Slyke, R., M., Wets, R., 1969, L-Shaped linear programs with applications to optimal control and stochastic programming, *SIAM*, 638-663.
- Vaziri, P., Davidson, R., Nozick, L., Hosseini, M., 2010, Resource allocation for regional earthquake risk mitigation: a case study of Tehran, Iran, *Natural Hazards*, 53 (3), 527–546.
- Verma, A., Gaukler, G. M., 2014, Pre-positioning disaster response facilities at safe locations: An evaluation of deterministic and stochastic modeling approaches, *Computers & Operations Research*, 62, 197–209.
- Viswanath, K., Peeta, S., 2003, Multicommodity maximal covering network design problem for planning critical routes for earthquake response. *Transportation Research Record*, 1857(1), 1-10.
- Xu, N., Davidson, R., Nozick, L., Dodo, A., 2007, The risk-return tradeoff in optimizing regional earthquake mitigation investment, *Structure and Infrastructure Engineering: Maintenance, Management, Life-Cycle*, 3 (2), 133-146.
- Yáñez-Sandivari, L., Cortés, C. E., Rey, P. A., 2020, Humanitarian Logistics and Emergencies Management: New perspectives to a sociotechnical problem and its optimization approach management, *International Journal of Disaster Risk Reduction*, 101952.
- Yücel, E., Salman, F. S., Arsik, I., 2018, Improving post-disaster road network accessibility by strengthening links against failures. *European Journal of Operational Research*, 269(2), 406-422.
- Zhou, Y., Jiang, Q., Qin, J., 2019, Pre-disaster retrofit decisions for sustainable transportation systems in urban areas, *Sustainability*, 11.15 (2019): 4044.
- Zitzler, E., Laumanns, M., Thiele, L., 2001, SPEA2: Improving the strength Pareto evolutionary algorithm, *TIK-report*, 103.
- Zolfaghari, M. R., Peyghaleh, E., 2015, Implementation of equity in resource allocation for regional earthquake risk mitigation using two-stage stochastic programming, *Risk Analysis*, 35, 434–458.

EKLER

EK-1 C# Dilinde Yazılan Stokastik Programlama Modelinin Kodları

```

1247 public void Yeni_Benders()
1248 {
1249     Cplex cplex = new Cplex();
1250     Cplex sub = new Cplex();
1251     INumVar flowCost;
1252     INumVar[][][] cpl_R = new INumVar[noDEMs][][][];
1253     INumVar[][][] cpl_J = new INumVar[noDEMs][][][];
1254     INumVar[,] cpl_U = new INumVar[bridge_number,period_no];
1255     INumVar[, , ,] cpl_X = new INumVar[noNodes, noNodes, Itemlist.Count, period_no,
1256     scenariolist.Count,NetworkSenaryo_number];
1257     INumVar[, , ,] cpl_W = new INumVar[noDEMs, Itemlist.Count, period_no, scenariolist.Count,NetworkSenaryo_number];
1258     INumVar[, , ,] cpl_D = new INumVar[noDEMs, Itemlist.Count, period_no, scenariolist.Count];
1259     for (int k = 0; k < noDEMs; k++)
1260     {
1261         cpl_R[k] = new INumVar[dem_loc_List[k].no_of_buildings][][][];
1262         for (int m = 0; m < dem_loc_List[k].no_of_buildings; m++)
1263         {
1264             cpl_R[k][m] = new INumVar[scenariolist.Count][][];
1265             for (int c = 0; c < scenariolist.Count; c++)
1266             {
1267                 cpl_R[k][m][c] = new INumVar[scenariolist.Count][];
1268                 for (int c1 = 0; c1 < scenariolist.Count; c1++)
1269                 {
1270                     cpl_R[k][m][c][c1] = new INumVar[period_no];
1271                     for (int t = 0; t < period_no; t++)
1272                     {
1273                         cpl_R[k][m][c][c1][t] = cplex.NumVar(0, 1, NumVarType.Bool);
1274                     }
1275                 }
1276             }
1277         }
1278     }
1279     for (int k = 0; k < noDEMs; k++)
1280     {
1281         cpl_J[k] = new INumVar[dem_loc_List[k].no_of_buildings][][][];
1282         for (int m = 0; m < dem_loc_List[k].no_of_buildings; m++)
1283         {
1284             cpl_J[k][m] = new INumVar[scenariolist.Count][][];
1285             for (int c = 0; c < scenariolist.Count; c++)
1286             {
1287                 cpl_J[k][m][c] = new INumVar[period_no];
1288                 for (int t = 0; t < period_no; t++)
1289                 {
1290                     cpl_J[k][m][c][t] = cplex.NumVar(0, 1, NumVarType.Bool);
1291                 }
1292             }
1293         }
1294     }
1295 }

```

```

1295     }
1296     for (int j = 0; j < noNodes; j++)
1297     {
1298         for (int k = 0; k < noNodes; k++)
1299         {
1300             for (int t = 0; t < period_no; t++)
1301             {
1302                 for (int s = 0; s < scenariolist.Count; s++)
1303                 {
1304                     for (int l = 0; l < ItemList.Count; l++)
1305                     {
1306                         for (int nw = 0; nw < NetworkSenaryo_number; nw++)
1307                         {
1308                             cpl_X[j, k, l, t, s, nw] = cplex.NumVar(0, double.MaxValue, NumVarType.Float);
1309                         }
1310                     }
1311                 }
1312             }
1313         }
1314     }
1315 }
1316 for (int k = 0; k < bridge_number; k++)
1317 {
1318     for (int t = 0; t < period_no; t++)
1319     {
1320         cpl_U[k, t] = cplex.NumVar(0, 1, NumVarType.Bool);
1321     }

```

```

1315     }
1316     for (int k = 0; k < bridge_number; k++)
1317     {
1318         for (int t = 0; t < period_no; t++)
1319         {
1320             cpl_U[k, t] = cplex.NumVar(0, 1, NumVarType.Bool);
1321         }
1322     }
1323 }
1324 for (int k = 0; k < noDEMs; k++)
1325 {
1326     for (int l = 0; l < Itemlist.Count; l++)
1327     {
1328         for (int t = 0; t < period_no; t++)
1329         {
1330             for (int s = 0; s < scenariolist.Count; s++)
1331             {
1332                 cpl_D[k, l, t, s] = cplex.NumVar(0, double.MaxValue, NumVarType.Float);
1333             }
1334             for (int nw = 0; nw < NetworkSenaryo_number; nw++)
1335             {
1336                 cpl_W[k, l, t, s, nw] = cplex.NumVar(0, double.MaxValue, NumVarType.Float);
1337             }
1338         }
1339     }
1340 }
1341 }

```



```

1342     double summl;
1343     int zv;
1344     // Create objective function for master problem:
1345     IObjective master_objective;
1346     ILinearNumExpr master_objExpr = cplex.LinearNumExpr();
1347
1348     summl = 0;
1349     foreach (Bridge r in Bridge_List)
1350     {
1351         for (int t = 0; t < period_no; t++)
1352         {
1353             for (int s = 0; s < scenariolist.Count; s++)
1354             {
1355                 for (int nw = 0; nw < NetworkSenaryo_number; nw++)
1356                 {
1357                     if (NetworkSenaryo_List[nw].network_vulner[r.id] == '1')
1358                     {
1359                         summl += NetworkSenaryo_List[nw].network_sce_prob[s] *
1360                             recovery_cost[r.id, t, s] * NetworkSenaryo_List[nw].network_vulner[r.id] *
1361                             haz_rate[t] * scenariolist[s].probability;
1362                         for (int t1 = 0; t1 <= t; t1++)
1363                         {
1364                             master_objExpr.AddTerm(-1 * NetworkSenaryo_List[nw].network_sce_prob[s] *
1365                                 recovery_cost[r.id, t, s] * NetworkSenaryo_List[nw].network_vulner[r.id] *
1366                                 haz_rate[t] * scenariolist[s].probability, cpl_U[r.id, t1]);
1367                         }
1368                     }
1369                 }
1370             }
1371         }
1372     }
1373     for (int s = 0; s < scenariolist.Count; s++)
1374     {
1375         for (int t = 0; t < period_no; t++)
1376         {
1377             zv = 0;
1378             for (int k = 0; k < noDEMs; k++)
1379             {
1380                 for (int m = 0; m < dem_loc_List[k].no_of_buildings; m++)
1381                 {
1382                     for (int c = 0; c < scenariolist.Count; c++)
1383                     {
1384                         for (int d = 0; d < damage_count; d++)
1385                         {
1386                             master_objExpr.AddTerm(haz_rate[t] * scenariolist[s].probability *
1387                                 dem_loc_List[k].damage_ratio * gamma_param[s, c, d] *
1388                                 damage_intensity[d] * buildingList[zv + m].build_recovery_cost, cpl_J[k][m][c][t]);
1389                         }
1390                     }
1391                 }
1392                 zv += dem_loc_List[k].no_of_buildings;
1393             }
1394         }
1395     }
1396
1397     //add surrogate cost
1398     flowCost = cplex.NumVar(0.0, Double.MaxValue, "estFlowCost");

```

```

1396
1397 //add surrogate cost
1398 flowCost = cplex.NumVar(0.0, Double.MaxValue, "estFlowCost");
1399 master_objExpr.AddTerm(1, flowCost);
1400 master_objective = cplex.AddMinimize(master_objExpr, "master_objective");
1401 //Add constraints
1402 ILinearNumExpr cpl_exp1 = cplex.LinearNumExpr();
1403 //const2
1404 IRange[] cpl_cons2 = new IRange[buildingList.Count * period_no];
1405 int counterUB1 = 0;
1406 int cst = -1;
1407 zv = 0;
1408 for (int t = 0; t < period_no; t++)
1409 {
1410     zv = 0;
1411     for (int k = 0; k < noDEMs; k++)
1412     {
1413         for (int m = 0; m < dem_loc_List[k].no_of_buildings; m++)
1414         {
1415             cst = buildingList[zv + m].resistant;
1416             cpl_exp1 = cplex.LinearNumExpr();
1417             for (int c = cst; c < scenariolist.Count; c++)
1418             {
1419                 cpl_exp1.AddTerm(1, cpl_J[k][m][c][t]);
1420             }
1421             cpl_cons2[counterUB1] = cplex.AddEq(cpl_exp1, 1);
1422             counterUB1++;
1423         }
1424         zv += dem_loc_List[k].no_of_buildings;
1425     }
1426 }
1427 cpl_exp1.Clear();
1428 ////const3-Evolution constraint
1429 IRange[] cpl_cons3 = new IRange[buildingList.Count * scenariolist.Count * scenariolist.Count * period_no];
1430 counterUB1 = 0;
1431 for (int t = 0; t < period_no; t++)
1432 {
1433     if (t == 0)
1434     {
1435         zv = 0;
1436         for (int k = 0; k < noDEMs; k++)
1437         {
1438             for (int m = 0; m < dem_loc_List[k].no_of_buildings; m++)
1439             {
1440                 for (int c = 0; c < scenariolist.Count; c++)
1441                 {
1442                     if (buildingList[zv + m].resistant == c)
1443                     {
1444                         for (int c1 = c; c1 < scenariolist.Count; c1++)
1445                         {
1446                             cpl_exp1 = cplex.LinearNumExpr();
1447                             cpl_exp1.AddTerm(1, cpl_R[k][m][c][c1][t]);
1448                             cpl_exp1.AddTerm(-1, cpl_J[k][m][c1][t]);

```

```

1435     zv = 0;
1436     for (int k = 0; k < noDEMs; k++)
1437     {
1438         for (int m = 0; m < dem_loc_List[k].no_of_buildings; m++)
1439         {
1440             for (int c = 0; c < scenariolist.Count; c++)
1441             {
1442                 if (buildingList[zv + m].resistant == c)
1443                 {
1444                     for (int c1 = c; c1 < scenariolist.Count; c1++)
1445                     {
1446                         cpl_exp1= cplex.LinearNumExpr();
1447                         cpl_exp1.AddTerm(1, cpl_R[k][m][c][c1][t]);
1448                         cpl_exp1.AddTerm(-1, cpl_J[k][m][c1][t]);
1449                         cpl_cons3[counterUB1] = cplex.AddGe(cpl_exp1, 0);
1450                         counterUB1++;
1451                     }
1452                 }
1453                 else
1454                 {
1455                     for (int c1 = c; c1 < scenariolist.Count; c1++)
1456                     {
1457                         cpl_exp1 = cplex.LinearNumExpr();
1458                         cpl_exp1.AddTerm(-1, cpl_R[k][m][c][c1][t]);
1459                         cpl_exp1.AddTerm(1, cpl_J[k][m][c1][t]);
1460                         cpl_cons3[counterUB1] = cplex.AddLe(cpl_exp1, 1);
1461                         counterUB1++;

```

```

1462     },
1463     zv += dem_loc_List[k].no_of_buildings;
1464     }
1465     }
1466     else
1467     {
1468         for (int k = 0; k < noDEMs; k++)
1469         {
1470             for (int m = 0; m < dem_loc_List[k].no_of_buildings; m++)
1471             {
1472                 for (int c = 0; c < scenariolist.Count; c++)
1473                 {
1474                     for (int c1 = c; c1 < scenariolist.Count; c1++)
1475                     {
1476                         cpl_exp1= cplex.LinearNumExpr();
1477                         cpl_exp1.AddTerm(1, cpl_R[k][m][c][c1][t]);
1478                         cpl_exp1.AddTerm(-1, cpl_J[k][m][c][t - 1]);
1479                         cpl_exp1.AddTerm(-1, cpl_J[k][m][c1][t]);
1480                         cpl_cons3[counterUB1] = cplex.AddGe(cpl_exp1, -1);
1481                         counterUB1++;
1482                     }
1483                 }
1484             }
1485         }
1486     }
1487 }
1488

```

```

1491     cpl_exp1.Clear();
1492
1493     ///Single U Kisiti
1494     IRange[] master_cons_U_top = new IRange[bridge_number];
1495     counterUB1 = 0;
1496     for (int j = 0; j < bridge_number; j++)
1497     {
1498         cpl_exp1 = cplex.LinearNumExpr();
1499         for (int t1 = 0; t1 < period_no; t1++)
1500         {
1501             cpl_exp1.AddTerm(1, cpl_U[j, t1]);
1502         }
1503         master_cons_U_top[counterUB1] = cplex.AddLe(cpl_exp1, 1);
1504         counterUB1++;
1505     }
1506     cpl_exp1.Clear();
1507     //const-5 budget_constraint
1508     IRange[] master_cons_budget = new IRange[period_no];
1509     for (int t = 0; t < period_no; t++)
1510     {
1511         cpl_exp1 = cplex.LinearNumExpr();
1512         zv = 0;
1513         for (int k = 0; k < noDEMs; k++)
1514         {
1515             for (int m = 0; m < dem_loc_List[k].no_of_buildings; m++)
1516             {
1517                 for (int c = 0; c < scenariolist.Count; c++)
1518                 {
1519                     for (int c1 = c; c1 < scenariolist.Count; c1++)
1520                     {
1521                         cpl_exp1.AddTerm(buildingList[zv + m].mit_cost * (c1 - c), cpl_R[k][m][c][c1][t]);
1522                     }
1523                 }
1524             }
1525             zv += dem_loc_List[k].no_of_buildings;
1526         }
1527         for (int j = 0; j < bridge_number; j++)
1528         {
1529             cpl_exp1.AddTerm(road_mit_cost[j, t], cpl_U[j, t]);
1530         }
1531         master_cons_budget[t] = cplex.AddLe(cpl_exp1, budget[t]);
1532     }
1533     cpl_exp1.Clear();
1534     // END CreateMasterILP
1535

```

```

1537
1538 // set up the subproblem
1539 //subproblem constraints
1540 IRange[] cpl_cons_vulnerability = new IRange[bridge_number * scenariolist.Count *
1541     period_no * NetworkSenaryo_number];
1542 counterUB1 = 0;
1543 foreach (Bridge r in Bridge_List)
1544 {
1545     for (int t = 0; t < period_no; t++)
1546     {
1547         for (int l = 0; l < Itemlist.Count; l++)
1548         {
1549             for (int s = 0; s < scenariolist.Count; s++)
1550             {
1551                 for (int nw = 0; nw < NetworkSenaryo_number; nw++)
1552                 {
1553                     cpl_exp1 = cplex.LinearNumExpr();
1554                     cpl_exp1.AddTerm(1, cpl_X[r.bn, r.en, l, t, s, nw]);
1555                     if (NetworkSenaryo_List[nw].network_vulner[r.id] == '1')
1556                     {
1557                         cpl_cons_vulnerability[counterUB1] = sub.AddLe(cpl_exp1, 0);
1558                         counterUB1++;
1559                     }
1560                     else
1561                     {
1562                         cpl_cons_vulnerability[counterUB1] = sub.AddLe(cpl_exp1, link_capacity[r.bn, r.en, s]);
1563                         counterUB1++;
1564                     }
1565                 }
1566             }
1567         }
1568     }
1569 }
1570
1571 IRange[] cpl_cons_capacity = new IRange[noNodes * noNodes * scenariolist.Count * period_no * NetworkSenaryo_number];
1572 counterUB1 = 0;
1573 for (int t = 0; t < period_no; t++)
1574 {
1575     for (int j = 0; j < noNodes; j++)
1576     {
1577         for (int l = 0; l < Itemlist.Count; l++)
1578         {
1579             for (int s = 0; s < scenariolist.Count; s++)
1580             {
1581                 for (int k = 0; k < noNodes; k++)
1582                 {
1583                     for (int nw = 0; nw < NetworkSenaryo_number; nw++)
1584                     {
1585                         cpl_exp1 = cplex.LinearNumExpr();
1586                         cpl_exp1.AddTerm(1, cpl_X[j, k, l, t, s, nw]);
1587                         cpl_cons_capacity[counterUB1] = sub.AddLe(cpl_exp1, link_capacity[j, k, s]);
1588                         counterUB1++;
1589                     }
1590                 }
1591             }
1592         }
1593     }
1594 }
1595

```

```

1598
1599 //DEmand constraint
1600 double top_lambda = 0;
1601 IRange[] cpl_cons_demand2 = new IRange[noDEMs * scenariolist.Count * period_no];
1602 counterUB1 = 0;
1603 for (int j = 0; j < noDEMs; j++)
1604 {
1605     for (int s = 0; s < scenariolist.Count; s++)
1606     {
1607         for (int t = 0; t < period_no; t++)
1608         {
1609             for (int l = 0; l < Itemlist.Count; l++)
1610             {
1611                 cpl_exp1 = cplex.LinearNumExpr();
1612                 cpl_exp1.AddTerm(1, cpl_D[j, l, t, s]);
1613                 cpl_cons_demand2[counterUB1] = sub.AddEq(cpl_exp1, 0);
1614                 counterUB1++;
1615             }
1616         }
1617     }
1618 }
1619
1620 IRange[] cpl_cons_network2 = new IRange[noDEMs * scenariolist.Count * period_no * NetworkSenaryo_number];
1621 IRange[] cpl_cons_network3 = new IRange[(noNodes - (noRRCs + noDEMs)) * scenariolist.Count * period_no * NetworkSenar
1619
1620 IRange[] cpl_cons_network2 = new IRange[noDEMs * scenariolist.Count * period_no * NetworkSenaryo_number];
1621 IRange[] cpl_cons_network3 = new IRange[(noNodes - (noRRCs + noDEMs)) * scenariolist.Count *
1622     period_no * NetworkSenaryo_number];
1623 counterUB1 = 0;
1624 int counterUB2 = 0;
1625 int counterUB3 = 0;
1626 sum1 = 0;
1627 RC_link_capacity=new float[noRRCs,scenariolist.Count];
1628 for (int j = 0; j <noRRCs; j++)
1629 {
1630     for (int s = 0; s < scenariolist.Count; s++)
1631     {
1632         RC_link_capacity[j,s] = 0;
1633
1634         for (int i = 0; i < noNodes; i++)
1635         {
1636             RC_link_capacity[j,s]+=link_capacity[j, i, s];
1637         }
1638     }
1639 }
1640
1641 for (int j = 0; j < noNodes; j++)
1642 {
1643     for (int s = 0; s < scenariolist.Count; s++)
1644     {
1645         for (int t = 0; t < period_no; t++)

```

```

1646         {
1647             for (int l = 0; l < Itemlist.Count; l++)
1648             {
1649                 cpl_exp1.Clear();
1650                 if (j >= noRRCs && j < (noRRCs + noDEMs))
1651                 {
1652                     for (int nw = 0; nw < NetworkSenaryo_number; nw++)
1653                     {
1654                         cpl_exp1 = cplex.LinearNumExpr();
1655                         sum1 = 0;
1656                         for (int i = 0; i < noNodes; i++)
1657                         {
1658                             cpl_exp1.AddTerm(1, cpl_X[j, i, l, t, s, nw]);
1659                         }
1660
1661                         for (int i = 0; i < noNodes; i++)
1662                         {
1663                             cpl_exp1.AddTerm(-1, cpl_X[i, j, l, t, s, nw]);
1664                         }
1665                         cpl_exp1.AddTerm(-1, cpl_W[j - noRRCs, l, t, s, nw]);
1666                         cpl_exp1.AddTerm(1, cpl_D[j - noRRCs, l, t, s]);
1667                         cpl_cons_network2[counterUB2] = sub.AddEq(cpl_exp1, 0);
1668                         counterUB2++;
1669                     }
1670                 }
1671                 cpl_exp1.Clear();
1672
1673                 if (j >= (noRRCs + noDEMs))
1674                 {
1675                     for (int nw = 0; nw < NetworkSenaryo_number; nw++)
1676                     {
1677                         cpl_exp1 = cplex.LinearNumExpr();
1678                         for (int i = 0; i < noNodes; i++)
1679                         {
1680                             cpl_exp1.AddTerm(1, cpl_X[j, i, l, t, s, nw]);
1681                         }
1682                         for (int i = 0; i < noNodes; i++)
1683                         {
1684                             cpl_exp1.AddTerm(-1, cpl_X[i, j, l, t, s, nw]);
1685                         }
1686                         cpl_cons_network3[counterUB3] = sub.AddEq(cpl_exp1, 0);
1687                         counterUB3++;
1688                     }
1689                 }
1690             }
1691         }
1692     }
1693 }
1694
1695 // Create objective function for master problem:
1696 //IOjective sub_objective;
1697 ILinearNumExpr sub_objExpr = sub.LinearNumExpr();
1698 for (int l = 0; l < Itemlist.Count; l++)

```



```

1695 // Create objective function for master problem:
1696 //IObjective sub_objective;
1697 ILinearNumExpr sub_objExpr = sub.LinearNumExpr();
1698 for (int l = 0; l < Itemlist.Count; l++)
1699 {
1700     for (int t = 0; t < period_no; t++)
1701     {
1702         for (int s = 0; s < scenariolist.Count; s++)
1703         {
1704             for (int j = 0; j < noNodes; j++)
1705             {
1706                 for (int nw = 0; nw < NetworkSenaryo_number; nw++)
1707                 {
1708                     if (j >= noRRCs && j < (noRRCs + noDEMs))
1709                     {
1710                         sub_objExpr.AddTerm(NetworkSenaryo_List[nw].network_sce_prob[s] * haz_rate[t] * scenariolist
1711                     }
1712                     for (int k = 0; k < noNodes; k++)
1713                     {
1714                         sub_objExpr.AddTerm(NetworkSenaryo_List[nw].network_sce_prob[s] * haz_rate[t] * scenariolist
1715                     }
1716                 }
1717             }
1718         }
1719     }
1720 }
1721 sub.AddMinimize(sub_objExpr, "Sub_Problem_objective"); // minimize total flow cost
1722
1723
1724 // disable presolving of the subproblem (if the presolver realizes the
1725 // subproblem is infeasible, we do not get a dual ray)
1726 // sub.SetParam(Cplex.BooleanParam.PreInd, false);
1727
1728
1729 cplex.Use(new BendersLazyConsCallback(flowCost, cpl_J, cpl_U, cplex, sub, noNodes, noDEMs, period_no, noRRCs,
1730 scenariolist, dem_loc_List, cpl_cons_vulnerability,
1731 cpl_cons_network2, link_capacity, damage_count, Itemlist, gamma_param,
1732 cpl_cons_capacity, cpl_cons_demand2, cpl_cons_network3, haz_rate));
1733
1734 // Set the maximum number of threads to 1.
1735 // This instruction is redundant: If MIP control callbacks are registered,
1736 // then by default CPLEX uses 1 (one) thread only.
1737 // Note that the current example may not work properly if more than 1 threads
1738 // are used, because the callback functions modify shared global data.
1739 // We refer the user to the documentation to see how to deal with multi-thread
1740 // runs in presence of MIP control callbacks.
1741
1742 cplex.SetParam(Cplex.IntParam.Threads, 1);
1743 cplex.SetParam(Cplex.IntParam.MIPSearch, Cplex.MIPSearch.Traditional);
1744 cplex.SetParam(Cplex.IntParam.HeurFreq, 50); //cok faydali
1745 cplex.SetParam(Cplex.Param.TimeLimit, 120);
1746 cplex.Solve();

```



```

1745     cplex.SetParam(Cplex.Param.TimeLimit,120);
1746     cplex.Solve();
1747     System.Console.WriteLine();
1748     System.Console.WriteLine("Solution status: " + cplex.GetStatus());
1749     opt_value_l_shaped =cplex.GetObjValue();
1750     l_shaped_LB =cplex.GetBestObjValue();
1751     sub.End();
1752     cplex.End();
1753 }
1754
1755
1756
1757 // The class BendersLazyConsCallback
1758 // allows to add Benders' cuts as lazy constraints.
1759 //
1760 // 2 başvuru
1761 internal class BendersLazyConsCallback : Cplex.LazyConstraintCallback
1762 {
1763     INumVar[][][] cpl_J;
1764     INumVar[,] cpl_U;
1765     Cplex sub_cplex;
1766     Cplex sub_sub;
1767     INumVar sub_flowCost;
1768     int noNodes, noDEMs, period_no, noRRCs, damage_count;
1769     List<Scenario> scenariolist;
1770     List<Demand_Location> dem_loc_List;
1771     List<Item> Itemlist;
1772
1773     Cplex sub_cplex;
1774     Cplex sub_sub;
1775     INumVar sub_flowCost;
1776     int noNodes, noDEMs, period_no, noRRCs, damage_count;
1777     List<Scenario> scenariolist;
1778     List<Demand_Location> dem_loc_List;
1779     List<Item> Itemlist;
1780     IRange[] cpl_cons_vulnerability;
1781     IRange[] cpl_cons_network2;
1782     float[,] link_capacity;
1783     double[,] gamma_param;
1784     IRange[] cpl_cons_capacity;
1785     IRange[] cpl_cons_demand;
1786     IRange[] cpl_cons_network3;
1787     double[] haz_rate;
1788     double l_lowerbound;
1789     // 1 başvuru
1790     internal BendersLazyConsCallback(INumVar flowCost, INumVar[][][] cpl_J, INumVar[,] cpl_U, Cplex cplex, Cplex sub,
1791         int noNodes, int noDEMs, int period_no, int noRRCs, List<Scenario> scenariolist,
1792         List<Demand_Location> dem_loc_List, IRange[] cpl_cons_vulnerability,
1793         IRange[] cpl_cons_network2, float[,] link_capacity, int damage_count,
1794         List<Item> Itemlist, double[,] gamma_param, IRange[] cpl_cons_capacity, IRange[] cpl_cons_demand,
1795         IRange[] cpl_cons_network3, double[] haz_rate)
1796     {
1797         this.cpl_J = cpl_J;
1798         this.cpl_U = cpl_U;
1799         sub_flowCost = flowCost;

```

```

1787         this._cpl_J = cpl_J;
1788         this._cpl_U = cpl_U;
1789         sub_flowCost = flowCost;
1790         sub_cplex = cplex;
1791         sub_sub = sub;
1792         _noNodes = noNodes;
1793         _noDEMs = noDEMs;
1794         _period_no = period_no;
1795         _noRRCs = noRRCs;
1796         _scenariolist = scenariolist;
1797         _dem_loc_List = dem_loc_List;
1798         _cpl_cons_vulnerability = cpl_cons_vulnerability;
1799         _cpl_cons_network2 = cpl_cons_network2;
1800         _link_capacity = link_capacity;
1801         _Itemlist = Itemlist;
1802         _damage_count = damage_count;
1803         _gamma_param = gamma_param;
1804         _cpl_cons_capacity = cpl_cons_capacity;
1805         _cpl_cons_demand = cpl_cons_demand;
1806         _cpl_cons_network3 = cpl_cons_network3;
1807         _haz_rate = haz_rate;
1808     }
1809     public override void Main()
1810     {

```

```

1809     public override void Main()
1810     {
1811         // Get Master Problem Flow cost estimate
1812         double zMaster = GetValue(sub_flowCost);
1813         double[][][] sub_cpl_J;
1814         int[,] sub_cpl_U;
1815         sub_cpl_J = new double[_noDEMs][][];
1816         sub_cpl_U = new int[bridge_number, _period_no];
1817         for (int k = 0; k < _noDEMs; k++)
1818         {
1819             sub_cpl_J[k] = new double[_dem_loc_List[k].no_of_buildings][];
1820             for (int m = 0; m < _dem_loc_List[k].no_of_buildings; m++)
1821             {
1822                 sub_cpl_J[k][m] = new double[_scenariolist.Count];
1823                 for (int c = 0; c < _scenariolist.Count; c++)
1824                 {
1825                     sub_cpl_J[k][m][c] = new double[_period_no];
1826                     for (int t = 0; t < _period_no; t++)
1827                     {
1828                         sub_cpl_J[k][m][c][t] = 0;
1829                     }
1830                 }
1831             }
1832         }

```

```

1832     ,
1833     int p;
1834     for (int k = 0; k < _noDEMs; k++)
1835     {
1836         p = 0;
1837         for (int m = 0; m < _dem_loc_List[k].no_of_buildings; m++)
1838         {
1839             for (int c = 0; c < _scenariolist.Count; c++)
1840             {
1841                 sub_cpl_J[k][m][c] = GetValues(_cpl_J[k][m][c]);
1842             }
1843         }
1844         p += _dem_loc_List[k].no_of_buildings;
1845     }
1846     for (int j = 0; j < bridge_number; j++)
1847     {
1848         for (int t = 0; t < _period_no; t++)
1849         {
1850             sub_cpl_U[Bridge_List[j].id,t] = -1;
1851             if (GetValue(_cpl_U[Bridge_List[j].id, t]) > 0.5)
1852             {
1853                 sub_cpl_U[Bridge_List[j].id, t] = 1;
1854             }
1855             else
1856             {
1857                 sub_cpl_U[Bridge_List[j].id, t] = 0;

```

```

1861     List<double> dual_RHS1;
1862     dual_RHS1 = new List<double>();
1863     int counter_callbck = 0;
1864     p = 0;
1865     for (int j = 0; j < bridge_number; j++)
1866     {
1867         for (int t = 0; t < _period_no; t++)
1868         {
1869             for (int s = 0; s < _scenariolist.Count; s++)
1870             {
1871                 for (int nw = 0; nw < NetworkSenaryo_number; nw++)
1872                 {
1873                     p = 0;
1874                     for (int t1 = 0; t1 <= t; t1++)
1875                     {
1876                         p += sub_cpl_U[Bridge_List[j].id, t1];
1877                     }
1878                     dual_RHS1.Add(_cpl_cons_vulnerability[counter_callbck].UB);
1879
1880                     if (NetworkSenaryo_List[nw].network_vulner[Bridge_List[j].id] == '1')
1881                     {
1882                         _cpl_cons_vulnerability[counter_callbck].UB = p*_link_capacity[Bridge_List[j].bn,
1883                             Bridge_List[j].en, s];
1884                         counter_callbck++;
1885                     }
1886                     else

```

```

1894     ,
1895     double aaa = 0;
1896     double top_lambda = 0;
1897     double yeni_top_lambda = 0;
1898     counter_callbck = 0;
1899     List<double> dual_RHS2;
1900     dual_RHS2 = new List<double>();
1901     int ts=0;
1902     int cst2 = 0;
1903     int zv2=0;
1904     for (int j = _noRRCs; j < (_noRRCs + _noDEMs); j++)
1905     {
1906         for (int s = 0; s < _scenariolist.Count; s++)
1907         {
1908             for (int t = 0; t < _period_no; t++)
1909             {
1910                 for (int l = 0; l < _Itemlist.Count; l++)
1911                 {
1912                     aaa = 0;
1913                     for (int m = 0; m < _dem_loc_List[j - _noRRCs].no_of_buildings; m++)
1914                     {
1915                         for (int c = 0; c < _scenariolist.Count; c++)
1916                         {
1917                             if (sub_cpl_J[j - _noRRCs][m][c][t] == 1)
1918                             {
1919                                 top_lambda = 0;
1920                                 for (int d = 0; d < _damage_count; d++)
1921                                 {
1922                                     if (sub_cpl_J[j - _noRRCs][m][c][t] == 1)
1923                                     {
1924                                         top_lambda = 0;
1925                                         for (int d = 0; d < _damage_count; d++)
1926                                         {
1927                                             top_lambda += _Itemlist[l].dam_item_need[d] * _dem_loc_List[j - _noRRCs].
1928                                                 damage_ratio * _gamma_param[s, c, d];
1929                                         }
1930                                         aaa = aaa + top_lambda;
1931                                     }
1932                                 }
1933                             }
1934                         }
1935                     }
1936                 }
1937             }
1938             _cpl_cons_demand[counter_callbck].SetBounds(aaa,aaa);
1939             counter_callbck++;
1940         }
1941     }
1942     ts+=_dem_loc_List[j - _noRRCs].no_of_buildings;
1943 }
1944 sub_sub.SetParam(Cplex.IntParam.Threads, 1);
1945 sub_sub.Solve();
1946 Cplex.Status status = sub_sub.GetStatus();
1947 INumExpr expr = sub_cplex.NumExpr();
1948 double q_s;
1949 if (status == Cplex.Status.Optimal)
1950 {

```

```

1935     }
1936     sub_sub.SetParam(Cplex.IntParam.Threads, 1);
1937     sub_sub.Solve();
1938     Cplex.Status status = sub_sub.GetStatus();
1939     INumExpr expr = sub_cplex.NumExpr();
1940     double q_s;
1941     if (status == Cplex.Status.Optimal)
1942     {
1943         q_s = sub_sub.ObjValue;
1944         if (zMaster < sub_sub.GetObjValue() - FUZZ) //add optimality cut
1945         {
1946             double[] d1 = sub_sub.GetDuals(_cpl_cons_vulnerability);
1947             double[] d2 = sub_sub.GetDuals(_cpl_cons_capacity);
1948             double[] d3 = sub_sub.GetDuals(_cpl_cons_demand);
1949
1950             counter_callbck = 0;
1951             for (int j = 0; j < bridge_number; j++)
1952             {
1953                 for (int t = 0; t < _period_no; t++)
1954                 {
1955                     for (int l = 0; l < _ItemList.Count; l++)
1956                     {
1957                         for (int s = 0; s < _scenariolist.Count; s++)
1958                         {
1959                             for (int nw = 0; nw < NetworkSenaryo_number; nw++)
1960                             {

```

```

1949
1950         counter_callbck = 0;
1951         for (int j = 0; j < bridge_number; j++)
1952         {
1953             for (int t = 0; t < _period_no; t++)
1954             {
1955                 for (int l = 0; l < _ItemList.Count; l++)
1956                 {
1957                     for (int s = 0; s < _scenariolist.Count; s++)
1958                     {
1959                         for (int nw = 0; nw < NetworkSenaryo_number; nw++)
1960                         {
1961                             if (d1[counter_callbck] != 0)
1962                             {
1963                                 expr = sub_cplex.Sum(expr, (d1[counter_callbck] * dual_RHS1[counter_callbck]));
1964                                 for (int t1 = 0; t1 <= t; t1++)
1965                                 {
1966                                     expr = sub_cplex.Sum(expr, sub_cplex.Prod(d1[counter_callbck] *
1967                                     _link_capacity[Bridge_List[j].bn, Bridge_List[j].en, s],
1968                                     _cpl_U[Bridge_List[j].id, t1]));
1969                                 }
1970                             }
1971                             counter_callbck++;
1972                         }
1973                     }
1974                 }

```

```

1977 counter_callbck = 0;
1978 for (int t = 0; t < _period_no; t++)
1979 {
1980     for (int j = 0; j < _noNodes; j++)
1981     {
1982         for (int l = 0; l < _ItemList.Count; l++)
1983         {
1984             for (int s = 0; s < _scenariolist.Count; s++)
1985             {
1986                 for (int k = 0; k < _noNodes; k++)
1987                 {
1988                     for (int nw = 0; nw < NetworkSenaryo_number; nw++)
1989                     {
1990                         if (d2[counter_callbck] != 0)
1991                         {
1992                             expr = sub_cplex.Sum(expr, (d2[counter_callbck] * link_capacity[j, k,
1993                             counter_callbck++];
1994                         }
1995                     }
1996                 }
1997             }
1998         }
1999     }
2000 }

```

```

2002 ts = 0;
2003 zv2 = 0;
2004 for (int j = 0; j < _noDEMs; j++)
2005 {
2006     for (int s = 0; s < _scenariolist.Count; s++)
2007     {
2008         for (int t = 0; t < _period_no; t++)
2009         {
2010             for (int l = 0; l < _ItemList.Count; l++)
2011             {
2012                 for (int m = 0; m < _dem_loc_List[j].no_of_buildings; m++)
2013                 {
2014                     cst2 = buildingList[zv2 + m].resistant;
2015                     for (int c = cst2; c < _scenariolist.Count; c++)
2016                     {
2017                         yeni_top_lambda = 0;
2018                         if (d3[counter_callbck] != 0)
2019                         {
2020                             for (int d = 0; d < _damage_count; d++)
2021                             {
2022                                 yeni_top_lambda += _ItemList[l].dam_item_need[d] *
2023                                 _dem_loc_List[j].damage_ratio * _gamma_param[s, c, d];
2024                             }
2025                             expr = sub_cplex.Sum(expr, sub_cplex.Prod(d3[counter_callbck]*
2026                             yeni_top_lambda, _cpl_J[j][m][c][t]));
2027                         }

```

```

2029         }
2030         counter_callbck++;
2031     }
2032 }
2033 }
2034     zv2 += _dem_loc_List[j].no_of_buildings;
2035 }
2036 long mem6 = GC.GetTotalMemory(false);
2037 // add the optimality cut
2038 IConstraint r = AddLocal((IRange)sub_cplex.Ge(sub_flowCost, expr));
2039 System.Console.WriteLine(">>> Adding optimality cut: " + r);
2040
2041 long mem7 = GC.GetTotalMemory(false);
2042 dual_RHS1.Clear();
2043 dual_RHS2.Clear();
2044 }
2045 else
2046 {
2047     //long mem9 = GC.GetTotalMemory(false);
2048     System.Console.WriteLine(">>> Accepting new incumbent with value " + sub_sub.ObjValue);
2049     //Console.WriteLine(mem9);
2050     // the master and subproblem flow costs match
2051     // -- record the subproblem flows in case this proves to be the
2052     // winner (saving us from having to solve the LP one more time
2053     // once the master terminates)
2054 }
2055
2056 else
2057 {
2058     // unexpected status -- report but do nothing
2059     System.Console.WriteLine("!!! Unexpected subproblem solution status: " + status);
2060 }
2061 }
2062 }
2063 // END BendersLazyConsCallback
2064
2065

```