

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**KALABALIK TAHLİYESİ İÇİN ETMEN TABANLI
SİMÜLASYON İLE ELDE EDİLEN VERİLER ÜZERİNDE
YAPAY ARI KOLONİ PROGRAMLAMA TABANLI
SEMBOİK REGRESYON ÇALIŞMASI**

**Hazırlayan
İbrahim ŞAHAN**

**Danışman
Dr. Öğr. Üyesi Beyza GÖRKEMLİ**

Yüksek Lisans Tezi

**Temmuz 2022
KAYSERİ**

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**KALABALIK TAHLİYESİ İÇİN ETMEN TABANLI
SİMÜLASYON İLE ELDE EDİLEN VERİLER ÜZERİNDE
YAPAY ARI KOLONİ PROGRAMLAMA TABANLI
SEMBOİK REGRESYON ÇALIŞMASI
(Yüksek Lisans Tezi)**

**Hazırlayan
İbrahim ŞAHAN**

**Danışman
Dr. Öğr. Üyesi Beyza GÖRKEMLİ**

**Temmuz 2022
KAYSERİ**

BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.

İbrahim ŞAHAN

İmza

“Kalabalık Tahliyesi İçin Etmen Tabanlı Simülasyon ile Elde Edilen Veriler Üzerinde Yapay Arı Koloni Programlama Tabanlı Sembolik Regresyon Çalışması” adlı Yüksek Lisans Tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi’ ne uygun olarak hazırlanmıştır.

Hazırlayan

İbrahim ŞAHAN

İmza

Danışman

Dr. Öğr. Üyesi Beyza GÖRKEMLİ

İmza

Bilgisayar Mühendisliği ABD Başkanı

Prof. Dr. Veysel ASLANTAŞ

İmza

TEŐEKKÜR

Bana her türlü yardımı ve fedakârlığı sağlayan, karşılaştığım her zorlukta bana rehberlik ederek akademik yolculuğumda yanımda olan, ilgi ve emeğini hiçbir zaman esirgemeyen değerli hocam Dr. Öğr. Üyesi Beyza Görkemli'ye saygı ve teşekkürlerimi sunarım.

Sahip olduğum her şeyde katkı ve emekleri olan, hayatım boyunca bütün zorluklarda beni yalnız bırakmadan sonsuz destekleyen, haklarını ödeyemeyeceğim; kıymetli annem Hülya ve babam Mehmet'e, ablam Zeynep ve abim Ömer'e sonsuz teşekkürü borç bilirim.

Bütün çalışmalarımnda, hayatımın her aşamasında yanımda yer alarak bana yardımcı olan, sabırla ve sevgiyle beni destekleyen, hayat yoldaşım Pınar Güner'e minnettarım.

İbrahim ŞAHAN

07 2022, KAYSERİ

KALABALIK TAHLİYESİ İÇİN ETMEN TABANLI SİMÜLASYON İLE ELDE EDİLEN VERİLER ÜZERİNDE YAPAY ARI KOLONİ PROGRAMLAMA TABANLI SEMBOLİK REGRESYON ÇALIŞMASI

İbrahim ŞAHAN

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü
Yüksek Lisans Tezi, Temmuz 2022
Danışman: Dr. Öğr. Üyesi Beyza GÖRKEMLİ

ÖZET

İnsan kaynaklı veya doğal yollarla meydana gelen acil durumlarla karşılaşıldığında kişilerin etkin bir şekilde tahliye edilebilmesi oldukça önemlidir. Günümüzde kalabalık tahliyesi oldukça popüler hale gelen ve önemini gün geçtikçe daha da artıran bir araştırma alanıdır. Acil durum öncesinde yapılan planlamalar ve alınan önlemler sürecin doğru yönetilebilmesini sağlar. Uygun planlama yapılabilmesi için sistem içerisindeki planlamaya etki edebilecek bütün parametrelerin önem derecesinin ve bu parametrelerin birbirleriyle olan ilişkilerinin bilinmesi önemlidir. Bu ilişkilerin tespiti için doğal yollarla yeterli verinin elde edilebilmesi pek mümkün olmamaktadır. Bu tez çalışmasında, bir alışveriş merkezi için kalabalık tahliyesi modellenerek etmen tabanlı simülasyon ile tahliye verileri üretilmiştir. Üretilen bu veriler üzerinde evrimsel hesaplama tabanlı otomatik programlama yöntemlerinden biri olan yapay arı koloni programlama (ABCP) kullanılarak sembolik regresyon çalışması yapılmıştır. Sembolik regresyonda kapı miktarı, kapı genişliği ve kişi sayısı olası bağımsız değişkenler (giriş değişkenleri), tahliye süresi ise bağımlı değişken (çıkış değişkeni) olarak dikkate alınmıştır. Sonuç olarak, çalışmada ele alınan alışveriş merkezinde kalabalık tahliyesi için farklı açılardan avantaj sağlayabilecek alternatif modeller sunulmuş ve ABCP'nin bu tür problemlerde kullanımına dair örnek bir çalışma literatüre kazandırılmıştır.

Anahtar Kelimeler: Kalabalık tahliyesi, otomatik programlama, sembolik regresyon, yapay arı koloni programlama (ABCP), etmen tabanlı simülasyon, simülasyon.

**SYMBOLIC REGRESSION STUDY BASED ON ARTIFICIAL BEE COLONY
PROGRAMMING ON DATA OBTAINED BY AGENT BASED SIMULATION
FOR CROWD EVACUATION**

İbrahim ŞAHAN

Erciyes University, Graduate School of Natural and Applied Sciences

Master Thesis, July 2022

Supervisor: Assist Prof. Dr. Beyza GÖRKEMLİ

ABSTRACT

It is very important that people can be evacuated effectively in the event of an emergency caused by human or natural means. Nowadays, crowd evacuation is a research area that has become very popular and is increasing its importance day by day. The plans made before the emergency and the precautionary measures taken ensure that the process can be managed correctly. In order to make proper planning, it is important to know the importance of all the parameters that may affect the planning in the system and the relations of these parameters with each other. It is unlikely that sufficient data can be obtained by natural means to determine these relations. In this thesis, crowd evacuation for a shopping mall was modeled and evacuation data were generated by agent based simulation. A symbolic regression study was performed on these generated data using artificial bee colony programming (ABCP), which is one of the evolutionary computation based automatic programming methods. In the symbolic regression, the number of doors, door width, and number of people were taken into account as possible independent variables (input variables) and the evacuation time as a dependent variable (output variable). As a result, alternative models that can provide advantages in different aspects for crowd evacuation in the shopping mall discussed in the study were presented, and a case study on the use of ABCP in such problems was introduced to the literature.

Keywords: Crowd evacuation, automatic programming, symbolic regression, artificial bee colony programming (ABCP), agent based simulation, simulation

İÇİNDEKİLER

KALABALIK TAHLİYESİ İÇİN ETMEN TABANLI SİMÜLASYON İLE ELDE EDİLEN VERİLER ÜZERİNDE YAPAY ARI KOLONİ PROGRAMLAMA TABANLI SEMBOLİK REGRESYON ÇALIŞMASI

BİLİMSEL ETİĞE UYGUNLUK.....	ii
YÖNERGEYE UYGUNLUK	iii
KABUL VE ONAY	iv
ÖNSÖZ.....	v
ÖZET	vi
ABSTRACT.....	vii
İÇİNDEKİLER	viii
KISALTMALAR.....	x
TABLolar LİSTESİ.....	xi
ŞEKİLLER LİSTESİ	xii
GİRİŞ.....	1

1. BÖLÜM

GENEL BİLGİLER ve LİTERATÜR ÇALIŞMASI

1.1. Kalabalık Tahliye Problemi	4
1.2. Kalabalık Tahliyesinde Etmen Tabanlı Simülasyon.....	5
1.3 Otomatik Programlama ve ABCP ile Sembolik Regresyon.....	8

2. BÖLÜM

YÖNTEM VE MATERYAL

2.1. Etmen Tabanlı Simülasyon ve Kalabalık Tahliyesi.....	14
2.2. ABCP ve Sembolik Regresyon Çalışması.....	18

3. BÖLÜM

DENEYSEL ÇALIŞMALAR

3.1 Etmen Tabanlı Simülasyon ile İlgili Parametrelerin Ayarları ve Deneyler....	25
3.2 ABCP ile İlgili Parametrelerin Ayarları ve Deneyler.....	30

4. BÖLÜM

TARTIŞMA-SONUÇ ve ÖNERİLER

4.1.Tartışma ve Sonuç.....	35
4.2.Öneriler	36

KAYNAKÇA.....	38
---------------	----

EKLER.....	43
------------	----

EK 1.....	43
-----------	----

EK 2.....	49
-----------	----

EK 2.1.....	49
-------------	----

EK 2.2.....	67
-------------	----

ÖZGEÇMİŞ.....	74
---------------	----

KISALTMALAR

AVM:	Alışveriş Merkezi
ECBAP:	Evrimsel Hesaplama Tabanlı Otomatik Programlama (Evolutionary Computation Based Automatic Programming)
ABC:	Yapay Arı Kolonisi (Artificial Bee Colony)
ABCP:	Yapay Arı Koloni Programlama (Artificial Bee Colony Programming)
CGP:	Kartezyen Genetik Programlama (Cartesian Genetic Programming)
GA:	Genetik Algoritma (Genetic Algorithm)
GP:	Genetik Programlamayı (Genetic Programming)
MHABCP:	Çok kovanlı ABCP (Multi Hive ABCP)
qABCP:	Hızlı ABPC (Quick ABCP)
sABCP:	Semantik ABCP (Semantic ABCP)
qsABCP:	Hızlı semantik ABCP (Quick Semantic ABCP)
SABCP:	Ölçekli ABCP (Scaled ABCP)
shABCP:	Daralan ABCP (Shrinking ABCP)
IDF:	Yağış Şiddet-Süre-Frekans (Intensity–Duration–Frequency)
ODE:	Basit Diferansiyel Denklemler (Ordinary Differential Equation)
AAP:	Yapay Karınca Probleminin (Artificial Ant Problem)

TABLÖLAR LİSTESİ

Tablo 3.1: Simülasyonda kullanılan parametre değerleri.....	25
Tablo 3.2: Kapı sayılarına göre açık olacak kapı numaraları.....	26
Tablo 3.3: ABCP yöntemine ait parametre değerleri.....	31
Tablo 3.4: ABCP deney sonuçları.....	32



ŞEKİLLER LİSTESİ

Şekil 2.1. Yöntemlerin genel mekanizması.....	13
Şekil 2.2. Tasarlanan AVM planı.....	15
Şekil 2.3. Sadeleştirilmiş müşteri davranış modeli örnek blok diyagramı.....	16
Şekil 2.4. Sadeleştirilmiş çalışan davranış modeli örnek blok diyagramı.....	17
Şekil 2.5: Sadeleştirilmiş örnek tahliye davranış blok diyagramı.....	18
Şekil 2.6: Üç giriş ve bir çıkış değişkeni olan sistemin blok diyagramı.....	18
Şekil 2.7: ABCP akış diyagramı.....	20
Şekil 2.8: Ağaç yapısında temsil edilen örnek çözüm.....	21
Şekil 2.9: ABCP yönteminin adımları.....	22
Şekil 2.10: Mevcut çözüm	24
Şekil 2.11: Popülasyondaki farklı bir çözüm x_k	24
Şekil 2.12: x_k 'nin paylaşacağı bilgi	24
Şekil 2.13: Üretilen aday çözüm v_i	24
Şekil 3.1: Deney sonucu elde edilen insan sayısı - ortalama tahliye süresi grafiği (bütün değerler için).....	27
Şekil 3.2: Deney sonucu elde edilen insan sayısı - ortalama tahliye süresi grafiği (ortanca değerler için).....	27
Şekil 3.3: Deney sonucu elde edilen kapı sayısı - ortalama tahliye süresi grafiği (bütün değerler için).....	28
Şekil 3.4: Deney sonucu elde edilen kapı sayısı - ortalama tahliye süresi grafiği (ortanca değerler için).....	28
Şekil 3.5: Deney sonucu elde edilen kapı genişliği - ortalama tahliye süresi grafiği (bütün değerler için).....	29
Şekil 3.6: Deney sonucu elde edilen kapı genişliği - ortalama tahliye süresi grafiği (ortanca değerler için).....	29
Şekil 3.7: Deney sonucu elde edilen en iyi değere sahip model (maksimum derinlik=15).....	32
Şekil 3.8: Deney sonucu elde edilen en iyi değere sahip model (maksimum derinlik=6).....	32

Şekil 3.9: Deney sonucu elde edilen en iyi değere sahip modelin ağaç yapısında temsili (maksimum derinlik=6).....	33
Şekil 3.10: ABCP'nin yakınsama performansı (maksimum derinlik=6).....	33
Şekil 3.11: ABCP'nin yakınsama performansı (maksimum derinlik=15).....	34



GİRİŞ

Tarih boyunca dünya üzerinde büyük felaketler yaşanmış, çok sayıda insan yaşamını yitirmiş, milyonlarca insan ise evsiz kalmıştır. İnsan kaynaklı veya doğal yollarla meydana gelen bu felaketlerde; bireylerin etkin bir şekilde yönlendirilmesi ve sürecin en iyi şekilde yönetilmesi hayati öneme sahiptir. Süreçlerin etkili bir şekilde yönetilebilmesi ise doğru planlamalara ve efor gerektiren çalışmalara bağlıdır. Deprem, yangın, sel, saldırı gibi acil durum ve felaketlerde yönetilmesi gereken en önemli süreçlerden birisi kalabalık tahliyesidir. Felaket esnasında ve sonrasında kişilerin en kısa sürede, güvenli bir şekilde tehlikeli alandan tahliye edilebilmesi uygun planlama ile mümkündür. Planlama yaparken tahliye dinamiklerine etki edebilecek bütün parametrelerin belirlenmesi, bu parametrelerin önem derecesi ve birbirleriyle olan ilişkilerinin bilinmesi planın başarısı için oldukça önemlidir. Planda yer alan parametrelerin ve birbirleriyle olan ilişkilerinin belirlenmesi ne kadar doğru olursa planlama o kadar uygun şekilde gerçekleştirilebilir. Gerçek hayatta bu verileri detaylıca toplamak, hangi parametreye dikkat edileceği bilinmeden veri bulmak, veri seti oluşturabilecek kadar ilgili olaylara rastlamak çok mümkün ve kolay değildir. Deneyleerin maliyetinin azaltılması, zaman tasarrufunun yapılabilmesi, verilerin elde edilmesi ve incelenmesi bakımından bu tür çalışmalarda simülasyon ortamları kullanılmaktadır.

Kalabalık tahliye modelleri literatürde iki tür altında sınıflandırılmakta ve incelenmektedir [1, 2, 3]. Bunlar; kalabalığın tamamına odaklanarak bireylerin akışını inceleyen makroskopik model ve bireyi temelde ele alan onun davranışlarını, kalabalıkla olan etkileşimini modelleyen mikroskopik modeldir. Makroskopik modelin mikroskopik modelden temel farkı, makroskopik modelde kalabalık tahliyesi için bireylerin duyguları hesaba katılmaz [3, 4]. Ayrıca mikroskopik modelde yalnızca topluluktaki bireylerin davranışı ve bireylerin toplulukla olan etkileşimi incelenmez, aynı zamanda her bireyin uzamsal ve zamansal davranışı da bu modelde tanımlanır [3, 4, 2].

Temelleri 1940'ların sonunda atılan, 1990'lardan sonra yaygınlaşmaya başlayan etmen tabanlı modelleme; özerk etmenlerin (agent) eylemlerini, birbirleriyle ve çevreleri ile olan etkileşimlerini değerlendirmeyi sağlamaktadır. Etmenlerin ilişkilerini ve etkilerini bir bütün olarak ele almayı sağlayan bu yöntem yakın zamanda, özellikle yapay zekâ çalışmaları ile gündeme gelerek daha popüler hale gelmiştir.

Kalabalık tahliyesinde etmen tabanlı simülasyon ile yapılan çalışmaların sayısı her geçen gün hızla artmaktadır [1]. Çalışmalarda; yayaların sosyal davranışlarını ve yaya akışını içeren kalabalık hareketleri, yapılan tahliye planlamalarının etkinliği, planlamada etkili olan parametrelerin keşfedilmesi, bu parametrelerin önem derecesi ve birbirleriyle olan etkileşiminin incelenmesi sıklıkla yer almaktadır. Ayrıca tasarımların, binanın iç mekan düzeni ve tahliye edilecek kişilerin deprem sonrası tahliye davranışlarıyla uyumluluğunu geliştirme üzerine de çalışmalar yapılmaktadır [5].

Bağımlı ve bağımsız değişkenler arasındaki ilişkinin incelenmesi için kullanılan istatistiksel yöntemlerden biri olan sembolik regresyon; ilişkilerin genel yapısı için analist öngörüsüne ihtiyaç duymaması yönüyle, klasik regresyon çalışmalarından ayrılmaktadır. Sembolik regresyon problemi adı verilen bu regresyon işleminde; sisteme ait giriş ve çıkış verileri kullanılarak sistemin matematiksel modeli üretilmeye çalışılır. Sembolik regresyon problemlerinin çözümünde evrimsel hesaplama tabanlı otomatik programlama (Evolutionary Computation Based Automatic Programming – ECBAP) yöntemleri kullanılmaktadır [6, 7, 8]. Bu tür yöntemler yapısal olarak evrimsel hesaplama temelli klasik optimizasyon algoritmalarından çözümlerin gösterimi ve yeni çözüm üretmede kullanılan varyasyon operatörlerinin farklılığı ile ayrılmaktadır. ECBAP yöntemlerinde alternatif çözümler olarak ilgili sistem için alternatif programların (modellerin) araştırma uzayında yer aldığı bir optimizasyon işlemi söz konusudur. Bu nedenle çözümlerin gösterimi için genellikle ağaç yapısı esas alınır. Karaboğa ve arkadaşları yapay arı kolonisi (Artificial Bee Colony - ABC) [9] algoritmasından esinlenerek yapay arı koloni programlama (Artificial Bee Colony Programming - ABCP) adı verilen bir ECBAP yöntemi geliştirmiştir [6]. ABCP birçok sembolik regresyon ve diğer otomatik programlama problemlerinde başarıyla kullanılmaktadır [6, 10, 11].

Daha önce de belirtildiği gibi tahliye için önemli olan yapısal parametrelerin neler olduğu ve bu parametrelerin birbirleri ile ilişkilerinin belirlenmesi kalabalık tahliyesinde önem

arz etmektedir. ECBAP yöntemleri ile gerçekleştirilen sembolik regresyon çalışmalarında bu yöntemlerin doğası gereği otomatik olarak gerçekleştirdikleri özellik seçimi işlemi de dikkate alındığında bu konuda kullanımlarının önemli avantajlar sağlayacağı görülmektedir. Ancak literatürde bu tür bir çalışmaya rastlanmamıştır. Bu tez çalışmasında, kalabalık tahliyesi için ilgili mekânın fiziksel özellikleri ile tahliye süresi arasındaki bağıntıyı gösterecek alternatif modellerin (fonksiyonların) tespit edilmesi hedeflenmiştir. Bu kapsamda tek katlı bir alışveriş merkezi (AVM) dikkate alınmış, AVM'nin kapı sayısı, kapı genişliği ve mevcut kişi sayısının olası giriş değişkenleri, tahliye süresinin ise çıkış değişkeni olduğu bir sistem için giriş ve çıkış değişkenleri arasındaki ilişkiyi göstermek üzere alternatif matematiksel bağıntılar sembolik regresyon çalışması ile elde edilmiştir. Sembolik regresyon için birçok çalışmada performansını kanıtlamış otomatik programlama yöntemlerinden biri olan ancak bildiğimiz kadarıyla henüz kalabalık tahliyesi problemlerinin çözümünde bir uygulaması bulunmayan ABCP kullanılmıştır. Sembolik regresyonda çözümlerin değerlendirilmesi için gerekli olan verilerin üretimi içinse etmen tabanlı modelleme ve simülasyon çalışması yapılmıştır.

Tezin ilk bölümünde kalabalık tahliye problemi, kalabalık tahliyesinde etmen tabanlı simülasyon, otomatik programlama ve ABCP ile sembolik regresyon konularında genel bilgiler ve literatür çalışması sunulmuştur. Tezin ikinci bölümünde problemin çözümü için kullanılan yöntemlerin detaylı anlatımı bulunmaktadır. Üçüncü bölümde ise yapılan deney çalışmaları ve bu deneylerin sonuçları paylaşılmaktadır. Son bölüm olan dördüncü bölümde tartışma, tez çalışmasının sonuçları ve ilerleyen zamanlarda yapılması planlanan, önerilen çalışmalara yer verilmiştir.

1. BÖLÜM

GENEL BİLGİLER ve LİTERATÜR ÇALIŞMASI

Bu bölümde kalabalık tahliye problemi, bu problem için etmen tabanlı simülasyon ve ABCP ile sembolik regresyon konularında genel bilgiler verilmekte, ilgili literatür bilgisi sunulmaktadır.

1.1. Kalabalık Tahliye Problemi

İnsan kaynaklı veya doğal yollarla meydana gelen acil durumlar esnasında kişilerin etkin bir şekilde tahliye edilebilmesi oldukça önemlidir. Daha önce alınmış önlemler ile felaket esnasında ve felaket sonrasında süreci doğru yönetebilmek hayati önem taşımaktadır. Yönetilmesi gereken süreçlerin en acil ve en önemli olanlarından biri kalabalığın tahliyesidir. Tahliye edilecek kişileri en kısa sürede ve güvenli bir şekilde tehlikeli alandan tahliye edebilmek yalnızca uygun planlama yapılabilmesi ile mümkündür. Uygun planlama yapılabilmesi için; sistem içerisindeki planlamaya etki edebilecek bütün parametrelerin önem derecesinin ve bu parametrelerin birbirleriyle olan ilişkilerinin bilinmesi önem arz etmektedir. Bu parametrelerin ve ilişkilerinin belirlenmesi ne kadar doğru olursa planlama o kadar uygun şekilde gerçekleştirilebilir. Bu konuda da gerçek hayatta bu verileri detaylıca toplamak, hangi parametreye dikkat edileceği bilinmeden veri bulmak çok mümkün ve kolay değildir. Bu nedenle, simülasyon ortamında veri üretmek ve yapılan planlamaların etkinliğini ölçmek için yoğun bir şekilde efor sarf edilmektedir.

1.2. Kalabalık Tahliyesinde Etmen Tabanlı Simülasyon

Etmen tabanlı model; özerk etmenlerin (agent) eylemlerinin, birbirleri ve çevreleri ile olan etkileşimlerinin sistem üzerindeki etkilerini bir bütün olarak değerlendirme amacıyla sistemi simüle etmeye yarayan bir hesaplama modelleri sınıfıdır. Temelleri 1940'ların sonunda basit olarak atılmasına rağmen yoğun işlem yüküne sahip olmasından dolayı 1990'lara kadar yaygınlaşmamıştır. Robert Axelrod çeşitli çalışmalar yaparak etmen tabanlı modeller geliştirmiştir [12, 13].

Kalabalık tahliyesinde etmen tabanlı simülasyon kullanılarak birçok çalışma yapılmıştır [13, 14, 5, 15, 1]. Yapılan bu çalışmalarda hedef ve başarı kriteri genellikle tahliye süresinin azaltılması olarak görülmektedir [16, 17, 15]. Bununla birlikte, kalabalık hareketleri (kalabalığın tahliye davranışları, dinamikleri), yapılan planlamaların etkinliği, planlamaya etki eden parametrelerin önem derecesi ve birbirleri ile ilişkisi de incelenmiştir [13, 14, 5, 15, 1].

Kalabalık tahliye modellemede hücreli otomat modeli [2, 3], sosyal kuvvet modeli [20], akış tabanlı model [21] ve etken tabanlı model [22, 23, 24, 14] gibi kalabalığın tahliye dinamiklerini farklı açılardan simüle eden yaklaşımlar ele alınmıştır [25].

Zhong ve arkadaşları tahliye planlama problemini çözmek için etmen tabanlı simülasyon ile evrimsel algoritma tabanlı yeni bir yaklaşım önermiştir [25]. Önerilen yöntem, tüm tahliye bölgesini optimizasyon sonucunda belirlenen bir ayırıcı fonksiyona bağlı olarak alt bölgelere bölmektedir. Her alt bölgeye bir çıkış kapısı atanmakta ve bireyler tahliye için ilgili kapıya doğru koşmaktadırlar. Böylece tahliye planlama problemi, bir sembolik regresyon problemine dönüştürülmüştür. Sembolik Regresyon için genetik ifade programlama (Genetic Expression Programming) kullanılmış, uygunluk (fitness) hesabında etmen tabanlı kalabalık simülasyonu ile elde edilen veriler dikkate alınmıştır. Simülasyon çalışmalarında, kapalı ortamların farklı karmaşıklık seviyesine göre iki senaryo belirlemişlerdir ve etmenlerin hareketlerine rehberlik etmek için sosyal kuvvet modelini [20] kullanmışlardır. Fonksiyon kümesi olarak temel matematik işlemleri $\{+, -, *, /\}$, terminal kümesi olarak da $\{d, w, n\}$ kullanılmıştır. d çıkış kapısına olan uzaklık, w çıkış kapısının genişliği ve n çıkış kapısına p 'deki ajandan daha kısa mesafede olan ajanların sayısıdır. Buradaki p , sınıflandırma görevini yerine getirmek için verilen herhangi bir p konumudur.

Zhong ve arkadaşları 2 modülden ve kapalı ortamların farklı karmaşıklık seviyesine göre 6 farklı senaryodan oluşan bir çalışma da gerçekleştirmişlerdir [26]. Eğitim modülünde, kartezyen genetik programlama (Cartesian Genetic Programming – CGP) için etmen tabanlı simülasyon kullanılarak sezgisel kuralın uygunlukları değerlendirilmiştir. Sonra da test modülünde en iyi sezgisel kural yine etmen tabanlı simülasyon kullanılarak test edilmiştir. Burada ilk çalışmadan farklı olarak kalabalık tahliye süresini en aza indirebilme amacıyla her alt bölgedeki etmenler için genel bir davranış kuralı bulmaya odaklanılmıştır. Simülasyonda özellikle ilk 4 senaryo eğitim vakalarını oluşturmak için kullanılmıştır. Her bir senaryo ile bireylerin sayısının sırasıyla 100, 200 ve 400 olarak ayarlandığı 3 eğitim vakası oluşturulmuştur. Dolayısıyla, toplam 12 eğitim vakası vardır. Benzer şekilde, 6 senaryonun tamamına dayalı olarak 24 test senaryosu dikkate alınmıştır. Her senaryo; bireylerin sayısının sırasıyla 180, 280, 550 ve 1000 olarak ayarlandığı 4 adet test vakası oluşturmak için kullanılmıştır.

Mahmood ve arkadaşları kitlesel toplanmalar için acil tahliye stratejilerini analiz etmek amacıyla etmen tabanlı simülasyon kullanmışlardır [17]. Kalabalığın tahliyesini modellemek için Kabe'de hac ziyaretinde bulunan kişilerin tahliye edilmesini senaryo olarak ele almışlardır. Çalışma kapsamında rastgele kalabalık tahliyesi (Random Crowd Evacuation), en kısa bölgesel mesafe (Shortest Regional Distance), genetik algoritma (Genetic Algorithm - GA) probleme uyarlanarak simülasyon ile test edilmiştir.

Haris ve arkadaşları etmen tabanlı simülasyon kullanarak toplamda 8 adet çıkış kapısı ve içinde engeller bulunan bir ortam üzerinde çalışmalarını gerçekleştirmişlerdir [27]. Kapalı ortamı 20 adet mantıksal alt bölgeye ayırarak bu ortamda en kısa mesafeli tahliye (Shortest Distance Evacuation) ve harmony arama algoritmasını (Harmony Search Algorithm) test etmişlerdir. Çalışmada tek kapının kapalı olduğu (7 kapılı) ve 2 kapının kapalı olduğu (6 kapılı) 2 senaryo kullanılmıştır.

Mahmood ve arkadaşları çalışmalarını devam ettirerek etmen tabanlı simülasyon ile kampüs tahliyesini ele almışlardır [15]. En yakın çıkış en kısa süre (Nearest Exit Shortest Time) algoritmasını öne sürmektedirler. En yakın çıkış en kısa süre algoritması 2 ana hedefe odaklanır. İlk olarak; algoritma, tehlike altındaki bina içinde bulunan tüm yayaların en yakın çıkışa erişmesini sağlar ve yayalar mevcut çıkışlar arasında, çıkışlarda tıkanma riskini en aza indirecek şekilde bölünür. Çalışmada bir kampüs binası ve 3 farklı

türde etmen modellenmiştir. Etmenlerin farklı türde olmasının amacı popülasyonu, toplanan verilere göre başlatmaktır. 160, 500 ve 1000 kişinin bulunduğu 3 senaryo üzerinde çalışma yürütülmüştür.

Lopez-Carmona ve Paricio Garcia davranışsal strateji optimizasyonu kullanarak bireysel düzeyde optimum çıkış seçimi ile koordineli grup dinamiğinin nasıl olması gerektiği üzerine çalışma yapmışlardır [16]. Farklı genişliklerde 8 kapıya sahip olan Madrid Arena vaka çalışması için seçilmiştir. Çıkış tercihi davranışının modellenmesi, simülasyonun modellenmesi, tahliye yolu bulma algoritmaları ve bunların etkinliğinin test edilmesi üzerine çalışmalar yapılmıştır. Etmen tabanlı simülasyon ve ayırık olay simülasyonu kullanılmıştır. Çok terimli logit model (Multinomial Logit Model) tabanlı oluşturulan sistemi incelemek ve bunu CGP yaklaşımıyla karşılaştırmak için gerçekleştirmeler yapılmıştır.

Hassanpour ve arkadaşları deprem sonrası tahliye dayalı bina içi yerleşim değerlendirmesi için bir hibrit hiyerarşik etmen tabanlı simülasyon yaklaşımı geliştirmişlerdir [5]. Çalışma kapsamında yapısal olmayan hasarın tahmini için bir hücrel otomata ve bir 2 boyutlu bina bilgi modeli hasar görselleştirmesi ile hiyerarşik hibrit etmen tabanlı model çatısı geliştirmişlerdir. Çalışmada akıllı etmenler modellenmeye çalışılmıştır ve bu etmenler için navigasyon modeli ile bir gezinme modeli oluşturulmaya çalışılmıştır. Navigasyon modeli ile pekiştirmeli öğrenme kullanılmıştır. Öğrenme bölümünün temeli, markov karar süreçlerine dayanmaktadır. Hasarlı ortamın farklı etki seviyeleri ile etmenlerin etkileşimi dikkate alınmıştır. Tahliye esnasında yaralı kişilerle karşılaşan ve onlara yardım eden insanların davranışları da modellenmiştir. Ayrıca çalışmada insanların korku seviyelerine göre davranışlarına da yer verilmiştir. Auckland Üniversitesi'ndeki yeni mühendislik binasının bir katından yaya tahliye süreci vaka çalışması olarak ele alınmıştır. Farklı senaryolar ile 5 popülasyon büyüklüğünün tahliye edildiği, 2 farklı iç yerleşim tasarım alternatifi gerçekleştirilmiştir. Alternatiflerin birinde 2 çıkış kapısı bulunmaktadır ve bu kapılar katın sağında ve solunda yer almaktadır. Diğer alternatifte ise daha geniş bir kapı katın ortasında bulunmaktadır.

Zong ve arkadaşları tahliye dinamiklerini simüle etmek için hücrel otomatalara ve sosyal güce (social force) dayalı bir tahliye modeli sunmuşlardır [28]. Tahliye sürecini optimize etmek için görsel rehberlik tabanlı bir ABC algoritması önermektedirler. Her

tahliye edilen kiři kendi g6rüş alanındaki 6nde gelen arının rehberlięinde ıkıřlara doęru hareket eder. alıřma kapsamında sosyal g6 modelindeki ekim ve itme, hızlı hareket ve daha az tıkanıklık elde etmek iin uygunluk fonksiyonunu tanımlamada kullanılmaktadır.

Kaur ve Kaur afet y6netimi iin oklu etmen tabanlı tahliye planlaması zerine bir derleme alıřması yapmıřlardır [1]. alıřmada, uygulanan tahliye teknikleri ile eřitli afet trlerini dikkate alan mevcut etmen tabanlı tahliye modellerinin sistematik bir řekilde incelemesi yapılmaktadır. Kullanılan stratejiler ve simlasyon programları karřılařtırılarak bunların g6receli avantaj ve dezavantajlarına yer verilmektedir.

1.3. Otomatik Programlama ve ABCP ile Sembolik Regresyon

Regresyon analizi baęımlı ve baęımsız deęiřkenler arasındaki iliřkinin incelenmesi iin kullanılan istatistiksel y6ntemlerden birisidir. 6nceleri, iliřkilerin genel yapısı iin analist 6ng6rüsü gerekirken ve modelde yer alan katsayılar eřitli y6ntemlerle belirlenirken; g6nmzde kaba 6n modele ihtiya duymadan regresyon iřlemleri yapılabilir. 1992 yılında J. R. Koza, GA'dan esinlenerek en 6nl ECBAP y6ntemlerinden biri olan genetik programlamayı (Genetic Programming - GP) [8] geliřtirerek bunun mmkn olduęunu g6stermiřtir. Sembolik regresyon adı verilen bu regresyon iřleminde; sisteme ait giriř ve ıkıř verileri kullanılarak sistemin matematiksel modeli retilmeye alıřılır. Bu model, katsayılar ve fonksiyon yapısı bakımından analist 6ng6rüsü kullanılmadan retilmeye alıřılmaktadır. 6zetle; klasik regresyon alıřmasında model oluřturulurken insan zekâsı, analist 6ng6rüsü kullanılırken sembolik regresyon ile bu iřlemin bilgisayarlar tarafından gerekleřtirilmesi saęlanmıřtır [29].

Otomatik programlama y6ntemlerinde ama elde bulunan verilerle sistem giriř ve ıkıř deęiřkenleri dikkate alınarak, girdilerden ıktıların nasıl elde edildięini aıklayan bir model oluřturma. ECBAP y6ntemleri istenen modeli oluřturabilmek iin evrimsel hesaplamaya dayalı optimizasyon algoritmalarını ve temel makine 6ęrenmesi iřlemlerini kullanmaktadır.

Karaboęa 2005 yılında, bal arılarının yiyecek kaynaęı arama davranıřlarından esinlenerek sr zekasına dayalı optimizasyon algoritmalarından ABC algoritmasını geliřtirmiřtir [9]. Literatrde, ABC'nin birok arařtırmacı tarafından eřitli problemlere bařarı ile

uygulandığı ve bilinen çoğu algoritma ile kıyaslandığında başarılı sonuçlar elde edildiği görülmektedir [22, 23, 32]. Karaboğa ve arkadaşları ABC algoritmasını geliştirerek ABCP adı verilen bir otomatik programlama yöntemi ortaya çıkarmıştır [6]. ABCP sembolik regresyon problemleri üzerinde test edilmiş ve performansı standart GP ve bazı GP versiyonlarıyla kıyaslanmıştır [6].

Golafshani ve Ashour kendiliğinden yerleşen betonun (Self-Compacting Concrete - SCC) elastik modülünün tahmini için biyocoğrafik tabanlı programlama (Biogeographical Based Programming) adı verilen bir sembolik regresyon yöntemi geliştirmiştir [33]. Çalışmada kendiliğinden yerleşen betona ait deneylerin veri seti kullanılarak biyocoğrafik tabanlı programlama modeli oluşturulmuştur. Ayrıca ABCP kullanılarak da model oluşturulmuş, bu 2 model karşılaştırılmıştır. Değerlendirme sonuçları, mevcut formüllere ve ABCP modeline göre biyocoğrafik tabanlı programlama modelinin biraz daha performanslı olduğunu göstermektedir.

Arslan ve Öztürk yüksek boyutlu sembolik regresyon problemlerinde kullanılmak üzere çok kovanlı ABCP (Multi Hive ABCP - MHABCP) adı verilen yeni bir öznitelik seçim yaklaşımı geliştirmiştir [34]. Standart ABCP'den farklı olarak; çözüm olan her bir sembolik model tek ağaç ile temsil edilirken, MHABCP'de her çözüm kovana denk gelebilecek bir dizi ABCP ağacından elde edilmiş çıktılarının ağırlıklı doğrusal bir bileşimini içinde barındırmaktadır. Çalışmada çeşitli veri setleri ile model performansı test edilmiş, GP ve ABCP kıyaslanmıştır. Deney sonuçlarına göre, MHABCP yüksek boyutlu sembolik regresyon problemleri üzerindeki öznitelik seçiminde diğer yöntemlere göre daha iyi performans göstermiştir.

Görkemli ve Karaboğa ABCP'nin performansını yükseltmek için üç yeni versiyon geliştirmiştir [7]. Hızlı ABPC (Quick ABCP - qABCP) geliştirilerek yakınsama performansının artması sağlanmış, yerel arama kabiliyeti artmış ve erken çevrimlerde daha kaliteli sonuçlara ulaşılmıştır. Standart ABCP'den farklı olarak; qABCP'de gözcü arı yiyecek kaynağını seçtikten sonra seçilen yiyecek kaynağının merkezde olduğu bölgeye gider ve daha sonra bu bölgedeki en iyi kaynağı bulur ve iyileştirmeye çalışır. Diğer bir versiyon olarak semantik ABCP (Semantic ABCP - sABCP) geliştirilerek yüksek yerellik sağlanmaya çalışılmış, çözüm kalitesinin artması sağlanmıştır. Semantik ABCP standart ABCP'nin bilgi paylaşım mekanizmasında semantik benzerlik temelli

operatörler kullanılarak oluşturulmuştur. İki yöntemin birleştirilmesi sonucunda da hızlı semantik ABCP (Quick Semantic ABCP - qsABCP) geliştirilmiştir. Çalışma kapsamında yöntemler literatür test problemlerinde iyi performans gösteren bazı otomatik programlama yöntemleriyle kıyaslanmıştır. Deney sonuçlarına göre önerilen yöntemlerin sembolik regresyon problemlerinde etkin bir şekilde kullanılabileceği görülmektedir. Ayrıca çalışmada, akış sürtünmesinde Colebrook denkleminin yaklaşımları için ABCP kullanılmıştır.

Boudardara ve Görkemli robotik yol planlaması için temel test problemlerinden biri olan yapay karınca probleminin (Artificial Ant Problem - AAP) çözümünde ABCP kullanmıştır [35]. Performans analizi için Santa Fe ve Los Altos Hills parkurlarında deneyler yapılmış ve sonuçlar GP'nin sonuçlarıyla kıyaslanmıştır. Deney sonuçlarına göre GP'den çok daha iyi çözümler ürettiği gözlemlenen ABCP'nin bu tür problemlerde etkin bir şekilde kullanılabileceği görülmektedir.

Arslan ve Öztürk sembolik regresyon, tahmin ve öznitelik seçimi problemlerinde GP ve ABCP için karşılaştırmalı bir çalışma yapmıştır [36]. Sembolik regresyonda cherkassy fonksiyonu için eğitim veri seti, tahminde pH veri seti, öznitelik seçiminde ise beton basınç dayanım veri seti kullanılmıştır. Deney sonuçları ABCP'nin yapısal tasarım için güçlü bir yöntem olduğunu göstermektedir.

Hara ve arkadaşları daha önce GP'de semantik tabanlı çaprazlamalardan biri olarak önerdikleri semantik kontrol çaprazlama'yı (Semantic Control Crossover - SCC) ABCP'de de kullanmıştır [37]. Önerilen yöntemin sembolik regresyon problemlerinde standart ABCP'den daha yüksek performansa sahip olduğu gözlemlenmiştir. Çalışma kapsamında SCC kullanan iki tür ABCP üretilmiştir. Standart ABCP'den farklı olarak yöntemlerin ilkinde, görevli ve gözcü arılar farklı arama stratejileri ile hareket etmektedirler. Gözcü arılar yerel, görevli arılar ise küresel arama yaparlar. Farklı olarak ikinci yöntemde ise arama stratejileri durağanlık derecesine göre değiştirilmektedir. Çözümler geliştirilebilirken yerel arama, uzun vadede geliştirilemediğinde ise küresel arama yapılmaktadır.

Boudouaoui ve Habbi standart ABCP'ye eklenen yapısal değişiklikler sonucunda geliştirdikleri ölçekli ABCP'yi (Scaled ABCP - SABCP) beş sembolik regresyon

kıyaslama problemi üzerinde test etmişlerdir [38]. Bu yöntemin standart ABCP'den temelinde farkı; yiyecek kaynakları ve çözüm temsili ile tahmin modeli formudur. Standart ABCP'de her yiyecek kaynağı bir ağaçla temsil edilirken, SABCP'de modelin verimliliğini artırmak için önyargı terimi b_0 ve ağırlıklandırma terimi b_1 eklenmektedir.

Arslan ve Öztürk çok sınıflı dokuları sınıflandırmada kullanmak üzere yapay arı kolonisi programlama tanımlayıcısı (ABCP-Descriptor) geliştirmiştir [39]. Tanımlayıcıya eğitim aşamasında her bir dokunun sınıflandırılması için rastgele iki örnek verilmektedir. Performans standart yerel ikili model (Local Binary Pattern) ve GP tanımlayıcısı ile kıyaslanmıştır. Yöntemin görüntü işlemede kullanışlı olduğu ve diğer iki yöntemden daha iyi performans gösterdiği belirtilmiştir.

Bozoğullarından ve Öztürk ABCP'nin programlama bilgisine gerek duyulmadan kolayca kullanılabilmesi için bulut tabanlı bir web uygulaması (ABCPWeb) geliştirmiştir [40].

Öztürk ve arkadaşları metabolomik verilerin özellik seçimi ve sınıflandırılması için ABCP kullanmışlardır [10]. Çalışma kapsamında üç geleneksel sınıflandırma ve öznelik seçme yöntemi, GP ve ABCP üç ayrı veri setine uygulanmıştır. Metabolomik verilerin sınıflandırılmasında otomatik programlama yöntemlerinin, biyobelirteç keşfinde ise GP'ye göre ABCP'nin daha performanslı olduğu gözlemlenmiştir.

Bozoğullarından ve arkadaşları standart makine öğrenmesi yöntemlerine ve GP'ye uygulanan transfer öğrenme yaklaşımını ABCP'ye uygulamıştır [41]. Daha önce eğitilmiş bir modelin benzer bir probleme uygulanmasını sağlayan bu yaklaşımın ABCP'ye uyarlanmasıyla ortaya çıkan algoritmalar ABCP-T olarak adlandırılmıştır. Çalışma kapsamında sembolik regresyon problemleri ile deneyler yapılmıştır ve bu deneylerin sonucunda standart ABCP'ye göre ABCP-T'nin daha performanslı olduğu görülmüştür.

Boudardara ve Görkemli AAP'yi ABCP ve onun yeni bir versiyonu olan daralan ABCP (Shrinking ABCP - shABCP) ile çözmüştür [11]. AAP için daha önceki çalışmalarında [35] ABCP kullanılsa da bu çalışmada analizler derinleştirilmiş ve yeni kıyaslama çalışmaları yapılmıştır. Yine çalışma kapsamında, periyodik olarak popülasyon büyüklüğünün azaltıldığı shABCP geliştirilmiş ve AAP'ye uygulanmıştır. ABCP ve shABCP rakip otomatik programlama yöntemleri ile Santa Fe ve Los Altos Hills

parkurlarında karşılaştırılmıştır. Sonuçlar shABCP ile gelen yeni özelliklerin ABCP’de performans artışı sağladığını göstermektedir.

Boudouaoui ve arkadaşları ABCP ile diferansiyel denklemleri çözmek için bir çalışma yapmıştır [42]. Standart ABCP doğrudan uygulanamadığı için probleme özel bir ABCP algoritması geliştirilmiştir. Yöntem birinci ve ikinci mertebeden doğrusal ve doğrusal olmayan basit diferansiyel denklemler (Ordinary Differential Equation - ODE), ODE sistemleri ve farklı boyutlardaki kısmi diferansiyel denklemler dahil olmak üzere farklı türde diferansiyel denklemlere uygulanmıştır. Geliştirilen yöntemin GP ve karınca koloni programlamaya kıyasla daha başarılı olduğu vurgulanmıştır.

Nekoei ve arkadaşları ifade programlama (expression programming) ile birleştirilen yapay arı koloni ifade programlamasını (Artificial Bee Colony Expression Programming) önermiştir [43]. Yöntem ABCP, genetik ifade programlama (Genetic Expression Programming), GP ve qABCP gibi otomatik programlama yöntemleriyle kıyaslanmış ve tahmin probleminde yapay arı koloni ifade programlamasının daha iyi sonuçlar verdiği belirtilmiştir.

Abbasianjahromi ve arkadaşları bir doğrusal ABCP versiyonu geliştirerek inşaat projelerinin güvenlik performansı için bir tahmin modeli sunmuştur [44]. Model, etkin kriterleri ölçerek projenin başlangıcında güvenlik performansını tahmin etmektedir. Güvenlik performansı zayıfsa personel önemli kriterleri iyileştirmeye çalışırken güvenlik performansı iyi ise yaptığı işin güvenliğinden emin olmaktadır.

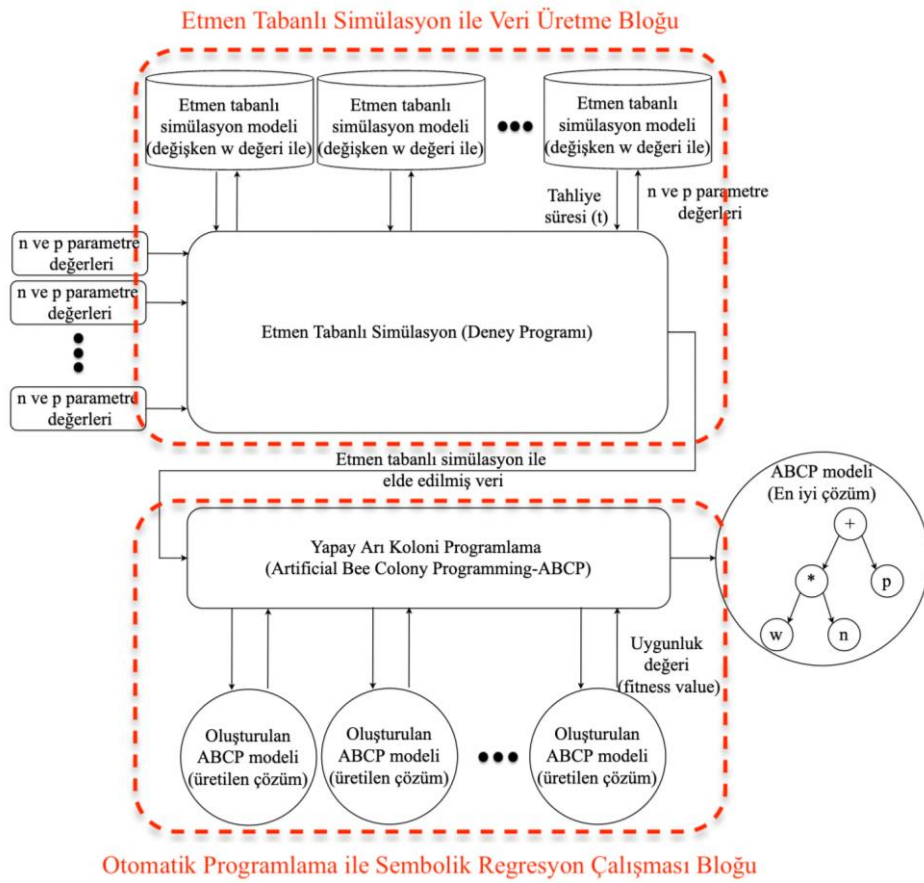
Özger ve arkadaşları venturili kondüitlerin havalandırma performansını modelleyebilecek fonksiyonlar elde etmek için ABCP kullanmıştır [45]. İki veri seti üzerinde ABCP, GP ve yapay sinir ağları kıyaslanmış, ABCP’nin diğer iki yöntemde de daha performanslı olduğu gözlemlenmiştir.

Görkemli ve arkadaşları İç Anadolu Bölgesi için bölgesel yağış şiddet-süre-frekans (Intensity–Duration–Frequency - IDF) ilişkisini tanımlamak üzere ABCP yöntemini kullanmıştır [46]. Çalışma sonucunda ABCP modellerinin, 2 ila 10.000 yıl arasındaki frekans aralığında IDF ilişkisini ampirik denklemlerden daha iyi tanımladığı görülmüştür.

2. BÖLÜM

YÖNTEM VE MATERYAL

Bu bölümde, öncelikle etmen tabanlı modelleme ve simülasyondan bahsedilerek tahliye verilerini üretmek için nasıl bir yol izlendiği açıklanmaktadır. Daha sonra, elde edilen veriler üzerinde yapılan sembolik regresyon çalışmasında kullanılan ABCP yöntemi sunulmaktadır. Çalışma kapsamında kullanılan yöntemlerin genel mekanizması Şekil 2.1’de gösterilmektedir.

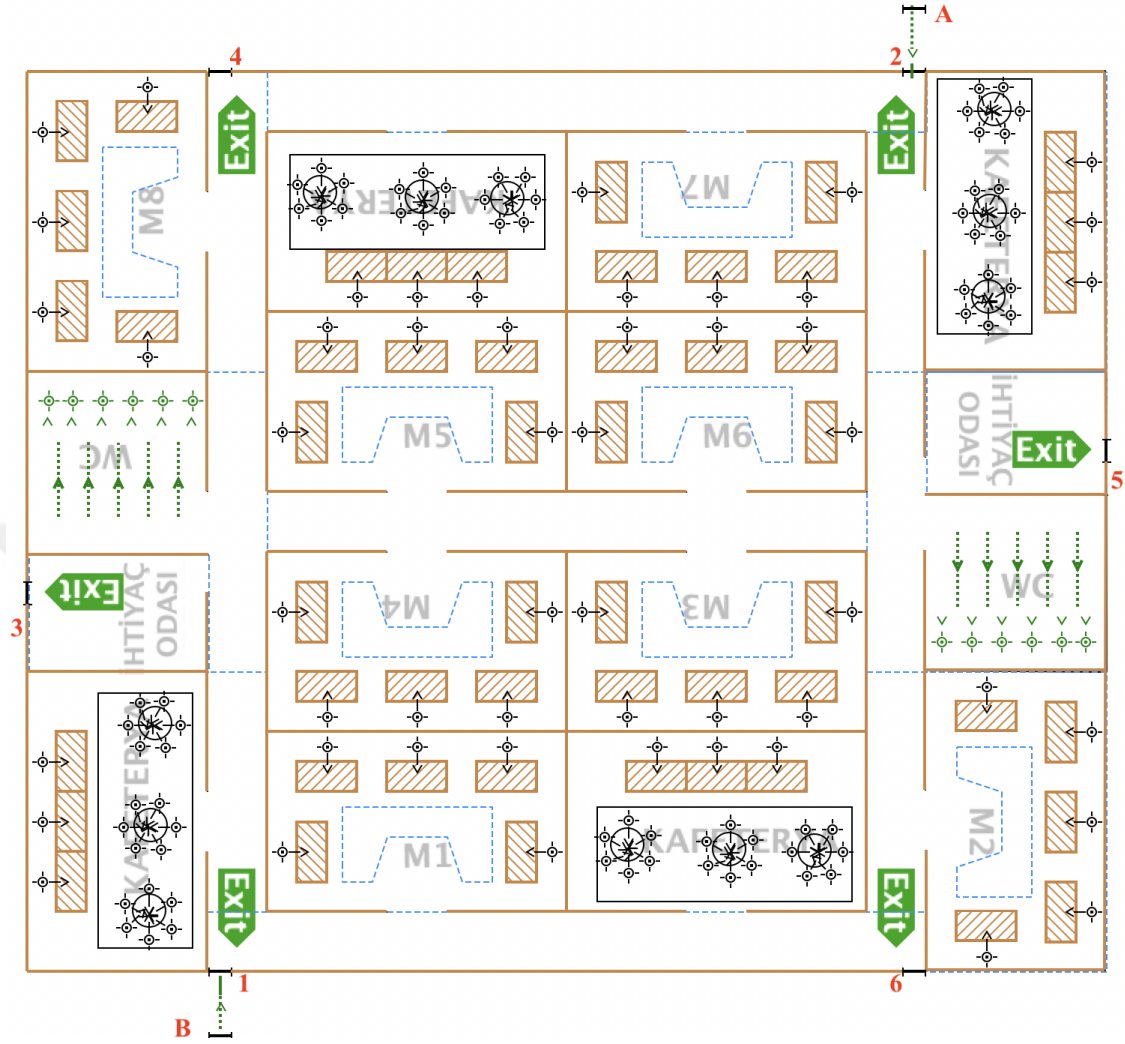


Şekil 2.1. Yöntemlerin genel mekanizması

2.1. Etmen Tabanlı Simülasyon ve Kalabalık Tahliyesi

Etmen tabanlı simülasyon; özerk etmenlerin birbirleri ve çevreleri ile olan etkileşimlerini, bu etkileşim sonucu sistem üzerinde oluşan etkilerini bir bütün olarak değerlendirmeye yarayan ve sistemin simüle edilebilmesini sağlayan bir hesaplama modeli sınıfıdır. Bahsedilen bu simülasyon modeli, her bir etmenin veya etmenleri etkileyecek herhangi bir olayın sisteme rahatlıkla dahil edilebilmesini ve sisteme ait bütün parçaların bir bütün olarak değerlendirilebilmesini sağlamaktadır.

Tez kapsamında AnyLogic simülasyon programı kullanılmış ve tahliye senaryoları için bir AVM vaka çalışması ele alınmıştır. AnyLogic, etmen tabanlı simülasyon modellerinin sürükle-bırak yöntemiyle rahatlıkla oluşturulabildiği, java yazılım dili ile geliştirmelere açık, içerisinde birçok kütüphane barındıran bir simülasyon programıdır. Program kullanılarak herhangi bir etmen için eylem veya parametre tanımlanabilmekte ve simülasyon sonucunda bunların etkileri gözlemlenebilmektedir. Simülasyon modeli oluşturulurken sırası ile çevre ve yaya modellemesi yapılmıştır. Çevre modellemesi için 8 mağaza, 4 kafeterya, 2 wc, 2 adet giriş-çıkış kapısı ve senaryolara göre değişen sayıda olmak üzere, maksimum 4 adet çıkış kapısı bulunmaktadır. Bu acil çıkış kapıları yalnızca tahliye anında çıkış yapılması için açılmaktadır. Senaryolara göre hiç acil çıkış kapısı bulunmayabileceği gibi; sırasıyla 1, 2, 3, 4 adet çıkış kapısı da bulunabilmektedir. Mağazalarda ve kafeteryalarda alışveriş için tezgahlar tasarlanmıştır. Ayrıca kafeteryalarda müşterilerin kullanabilmesi için masalar da simülasyon programında oluşturulmuştur. Simülasyon programında tasarlanmış olan AVM modeli Şekil 2.2'de verilmiştir.

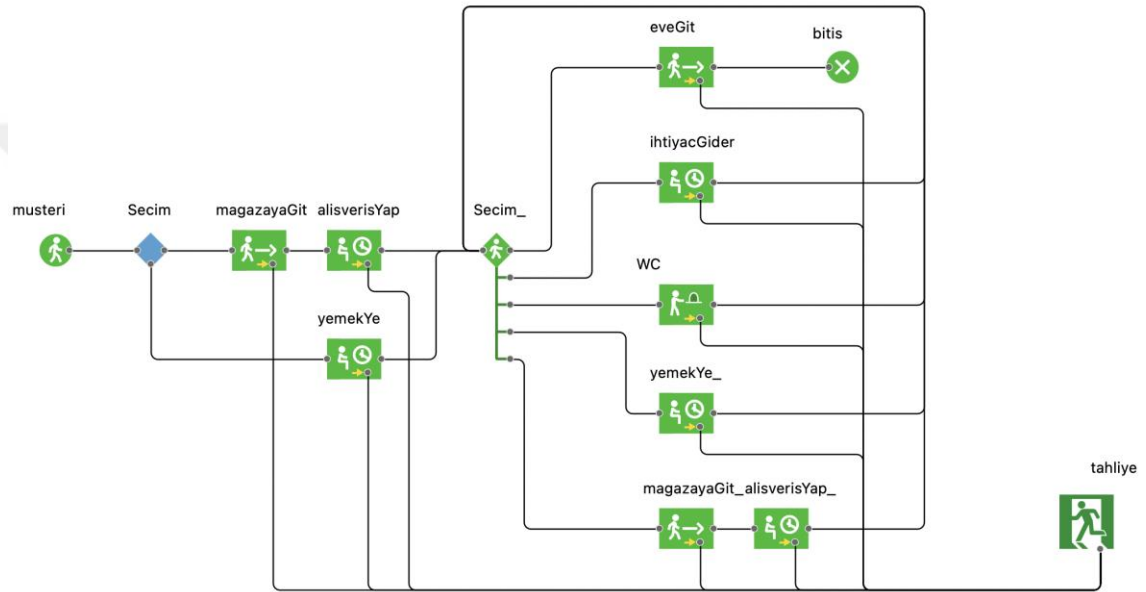


Şekil 2.2. Tasarlanan AVM planı

Yaya modellemesi için öncelikle yaya fiziki olarak tasarlanmış, daha sonra yaya davranışları (en yakın çıkışa yönelmek vb.) ve psikolojik etkenler (endişe seviyesi vb.) tanımlanmıştır. Müşteri ve çalışan olmak üzere 2 tür yaya çeşidi bulunmaktadır. Çalışanlar 08.00-23.00 saatleri arasında, müşteriler ise değişken yoğunluk seviyelerinde olmak üzere 10.00-22.00 saatleri arasında AVM’de bulunmaktadır. 104 çalışan 2 vardiya olacak şekilde (her vardiyada 52 çalışan bulunmaktadır) 7 saat 30 dakika çalışmaktadır. Müşteriler ve çalışanlar, AVM’ye kapıları rastgele seçerek ve her kapıdan yalnızca 1 kişi geçebilecek şekilde girmektedir. Her kapıdan yalnızca 1 kişi geçmesinin sebebi kapılarda güvenliklerin bulunmasıdır. Bütün yayalar simülasyon ortamına Şekil 2.2’de A ve B ile işaretlenen çizgilerden (TargetLine) giriş yapmaktadır. Simülasyon ortamından çıkışlar

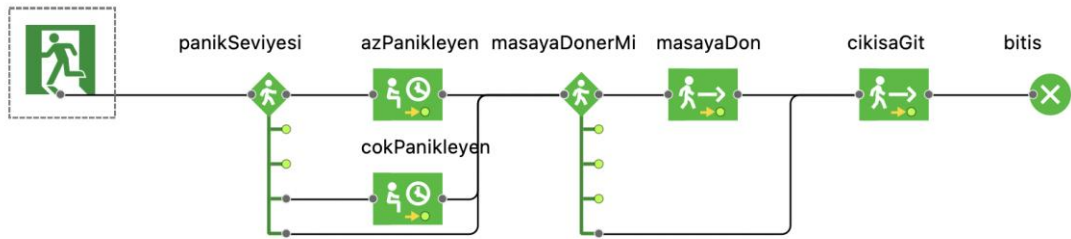
ise kapılarda bulunan çizgilerden yapılmaktadır. Alışveriş yapmak isteyen müşteriler mağazaları, kafeteryaya gitmek isteyen müşteriler kafeteryaları rastgele seçmektedir.

Müşteri davranış modelinin sadeleştirilmiş bir hali örnek olarak Şekil 2.3'te verilmektedir. Sürükle-bırak yöntemi ile oluşturulan bu blok diyagramda yeşil renkli parçaların hepsi temelde etmen görevi üstlenmektedir. Bu etmenlere değer atanabilmekte, çeşitli görevler üstlenebilmesi için kodlanabilmektedir.



Şekil 2.3. Sadeleştirilmiş müşteri davranış modeli örnek blok diyagramı

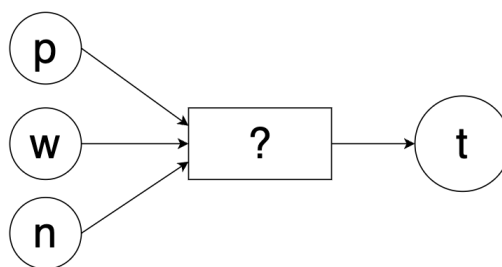
Müşteri blok diyagramda gösterildiği gibi öncelikle yemek yemek veya alışveriş yapmak için tercihte bulunmaktadır. Gerçekleştirdiği bu eylemden sonra yeni bir tercih yaparak sırasıyla eve gidebilir, çeşitli ihtiyaçları için ihtiyaç odasını kullanabilir, wc kullanabilir, daha önce yemek yedi ise alışveriş yapmaya başlayabilir, alışveriş yapıyorsa alışverişe devam edebilir veya yemek yemeye başlayabilir. Müşteriler bu eylemleri çalışma kapsamında belirli olasılıklarla tercih etmektedir. Çalışanlar ise müşterilerden farklı olarak çalışacağı alana giderek çalışmaya başlar. Devam edeceği eylemler mesaisinin bitmesine göre çeşitlenmektedir. Mesai bitmedi ise belirli bir saati geçtikten sonra sırasıyla yemek molası verebilir, wc kullanabilir, çeşitli ihtiyaçları için ihtiyaç odasını kullanabilir veya çalışmaya devam edebilir. Mesaisi bitti ise eve gidebilir veya müşteri davranışlarına geçebilir. Bütün bu eylemleri müşteride olduğu gibi belirli bir



Şekil 2.5: Sadeleştirilmiş örnek tahliye davranış blok diyagramı

2.2. ABCP ve Sembolik Regresyon Çalışması

Karaboğa 2005 yılında, bal arılarının yiyecek kaynağı arama davranışlarını simüle eden bir sürü zekasına dayalı optimizasyon yöntemi olarak ABC algoritmasını literatüre tanıtmıştır [9][30][32]. ABCP ise Karaboğa ve arkadaşları tarafından geliştirilen ve ABC algoritmasına dayanan yüksek seviyeli bir otomatik programlama yöntemidir [6]. ABCP gibi otomatik programlama yöntemleri ile sembolik regresyon çalışması yapılırken sisteme ait giriş ve çıkış verileri kullanılarak sistemin matematiksel modeli üretilmeye çalışılmaktadır. Tez kapsamında ele alınan sistem, AVM'de bulunan insan sayısı (p), çıkış kapısı sayısı (n), kapı genişliği (w) olmak üzere 3 adet girişe; tahliye süresi (t) olarak da 1 adet çıkışa sahiptir. Sistemin genel yapısı Şekil 2.6'da verilen blok şema ile gösterilmektedir.

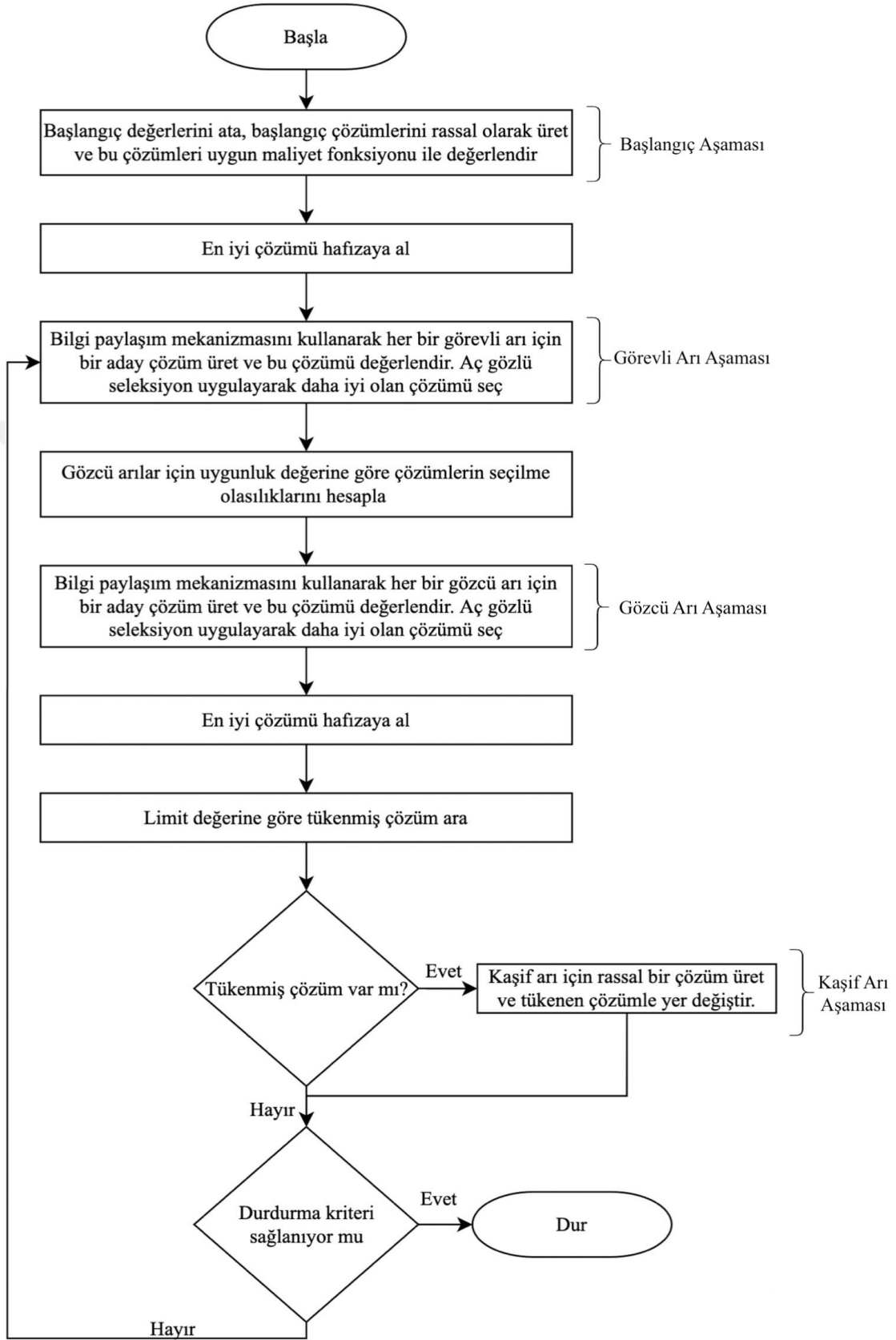


Şekil 2.6: Üç giriş ve bir çıkış değişkeni olan sistemin blok diyagramı

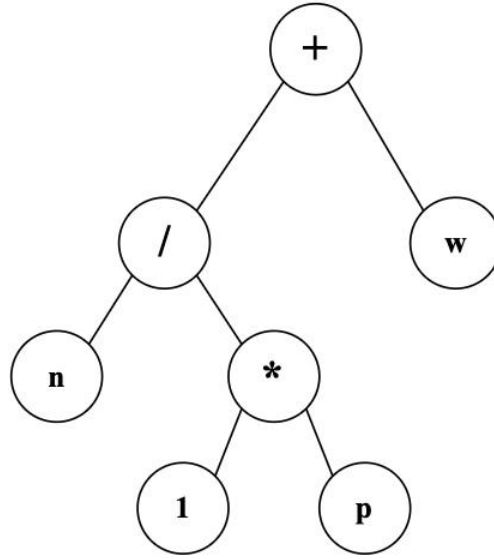
Akış diyagramı Şekil 2.7'de verilen ABCP, ABC ile benzer adımlara sahiptir ve ABC'nin genel araştırma yapısını esas almaktadır. Her iki yöntemde de popülasyonda yer alan çözümler yiyecek kaynağı pozisyonlarını temsil etmektedir ancak ABC'de çözümlerin gösterimi için sabit boyutlu diziler, ABCP'de ise ağaç yapısı kullanılmaktadır. Ağaç yapısı ile gösterilen her bir çözüm (program), probleme özgü terminal ve fonksiyon

kümelerinden seçilen değerlerin uygun bir kombinasyonudur. Fonksiyon kümesi problemin yapısına göre; aritmetik, mantıksal, matematiksel işlemler ve programlama işlemleri ile sisteme özgü farklı fonksiyonları içermektedir. Sembolik regresyon problemlerinde ise genellikle $+$, $-$, $*$, $/$ vb. gibi aritmetik ve \sin , \cos , \log , \exp vb. gibi matematiksel işlemler kullanılmaktadır. Burada çeşitli ilişkilerle çocuk düğümleri birbirine bağlayan ebeveyn düğümler bulunmaktadır. Terminal kümeleri ise sisteme ait bağımsız değişkenlerden ve sabitlerden oluşmaktadır. Burada yaprak düğümlerde kullanılan değerler bulunmaktadır. Tez kapsamında yapılan çalışmada fonksiyon kümesinin elemanları olarak $+$, $-$, $*$, $/$; terminal kümesinin elemanları olarak ise AVM’de bulunan insan sayısı (p), çıkış kapısı sayısı (n), kapı genişliği (w), sabit bir sayı olarak 1 ve $[-1,1]$ aralığında rastgele üretilmiş bir sayıyı temsil eden R de kullanılmıştır.

ABCP’de alternatif bir çözüm, verilen girişlerden elde edilen çıktılarının oluşmasını açıklamaya yaramaktadır ve sistemin tanımını yapmaktadır. Sistemde bulunan bağımlı, bağımsız değişken tanımlamasını ve bu değişkenlerin arasındaki ilişkilerin açıklanmasını sağlamaktadır. Şekil 2.8’de ağaç yapısında gösterilen örnek çözümün matematiksel ifadesi ise Eşitlik 2.1’de verilmektedir. Burada t ile bağımlı değişken, n, w, p ile de bağımsız değişkenler temsil edilmektedir.



Şekil 2.7: ABCP akış diyagramı



Şekil 2.8: Ağaç yapısında temsil edilen örnek çözüm

$$t = f(n, w, p) = (n/(1 * p)) + w \quad (2.1)$$

ABCP'nin algoritmik adımları Şekil 2.9'da sunulmuştur. Kolonide işlevleri farklı olan 3 arı tipi bulunmaktadır. Bunlar; hafızalarındaki yiyecek kaynağını tüketmek üzere kovandan ayrılan ve kovanda yiyecek kaynağı ile ilgili danslarıyla bilgi paylaşımı yapan görevli arılar, görevli arıdan aldığı bilgileri değerlendirerek yiyecek kaynağı seçen gözcü arılar ve bilgi paylaşımından yararlanmadan, hafızasında yer alan bir kaynağı hedeflemeden yiyecek arayan kâşif arılardır. Başlangıçta çözüm üretilirken bütün görevli arılar kâşif pozisyonunda olur ve rastgele çözümler üretirler. Rastgele çözümler üretildikten sonra kaşifler görevli arı olarak çalışırlar. Yiyecek kaynakları tüketildiğinde (kaynaklarının tükenmesinin kontrol edildiği limit parametresinin değeri aşıldığında), ilgili görevli arılar yeniden kâşif arı pozisyonuna geçer. Standart ABCP'de, kolonide yer alan görevli ve gözcü arı sayısı birbirine ve popülasyondaki yiyecek kaynağı sayısına eşittir. Her optimizasyon çevriminde en fazla bir kâşif arının ortaya çıkmasına izin verilmektedir.

Başlangıç aşaması

Başlangıç çözümlerini (x_i) *ramped-half-and-half* metodu ile rassal olarak üret (Aynı çözümlerin üretilmesini engelle).

Eşitlik 2.2'yi kullanarak çözümleri değerlendir.

En iyi çözümü hafızaya al.

REPEAT**Görevli arı aşaması**

Her bir görevli arı için;

Mevcut çözümden (x_i) bilgi paylaşım mekanizmasını kullanarak aday bir çözüm (v_i) üret.

Çözümü değerlendirmek için maliyet fonksiyonu Eşitlik 2.2 ile v_i 'nin maliyet fonksiyon değerini belirle.

Aç gözlü seleksiyon uygulayarak x_i ve v_i arasında maliyet değerine göre en iyi olan çözümü seç.

Eğer v_i seçilirse deneme sayacını sıfırla, seçilmezse mevcut çözümü kullanmaya devam et ve deneme sayacını bir artır.

Sırası ile Eşitlik 2.3 ve Eşitlik 2.4'ü kullanarak çözümlerin seçilme olasılığını (P_i) hesapla.

Gözcü arı aşaması

Her bir gözcü arı için;

P_i 'ye bağlı olarak bir çözüm (x_i) seç.

Mevcut çözümden (x_i) bilgi paylaşım mekanizmasını kullanarak aday bir çözüm (v_i) üret.

Çözümü değerlendirmek için maliyet fonksiyonu Eşitlik 2.2 ile v_i 'nin maliyet fonksiyon değerini belirle.

Aç gözlü seleksiyon uygulayarak x_i ve v_i arasında maliyet değerine göre en iyi olan çözümü seç.

Eğer v_i seçilirse deneme sayacını sıfırla, seçilmezse mevcut çözümü kullanmaya devam et ve deneme sayacını bir artır.

Şu ana kadar ki en iyi çözümü hafızaya al.

Kâşif arı aşaması

Limit değerine göre tükenmiş bir yiyecek kaynağı (çözüm) var mı belirle.

Eğer tükenmiş bir çözüm varsa kâşif arı için *grow* metodu ile rassal olarak yeni bir çözüm üret ve üretileni tükenmiş çözüm ile değiştir.

İlgili çözümün deneme sayacını sıfırla.

UNTIL (Durdurma kriterlerinin sağlanması)

Şekil 2.9: ABCP yönteminin adımları

ABCP yönteminde başlangıç çözümleri (başlangıç popülasyonu) üretilirken *ramped half-and-half* metodu kullanılmakta ve ayrıca aynı çözümlerin popülasyona dahil edilmesi engellenmektedir. Bir yiyecek kaynağının tükenmesi durumunda kâşif arı için yeni kaynak pozisyonu belirlenirken *grow* metodu kullanılarak popülasyon içerisindeki

mevcut çözümlerden farklı olan yeni bir çözüm üretilmektedir. Ramped half-and-half ve grow metotlarının detaylarına ulaşmak için [8] incelenebilir.

Üretilen çözümlerin değerlendirilmesinde kullanılan maliyet değeri hesabında toplam mutlak hata dikkate alınmış ve optimizasyon sürecinde bu değerin minimizasyonu yapılmıştır. Eşitlik 2.2’de $f(x_i)$ popülasyondaki i . çözümün maliyet değerini, Ne ise toplam uygunluk durumu (fitness case) sayısını ifade etmektedir. fc uygunluk durumu için; c_{fc} ulaşılmak istenen hedef değeri, $s_{(i,fc)}$ ise i . çözümün ürettiği çıkış değerini göstermektedir.

$$f(x_i) = \sum_{fc=1}^{Ne} |s_{(i,fc)} - c_{fc}| \quad (2.2)$$

ABC’de bulunan, maliyet değerini ($f(x_i)$) uygunluk değerine ($fit(x_i)$) dönüştürmede kullanılan Eşitlik 2.3 ABCP’de de aynı şekilde kullanılmaktadır. Maliyet fonksiyonu göz önüne alındığında, çalışma kapsamında bu dönüşüm için Eşitlik 2.4 dikkate alınmıştır.

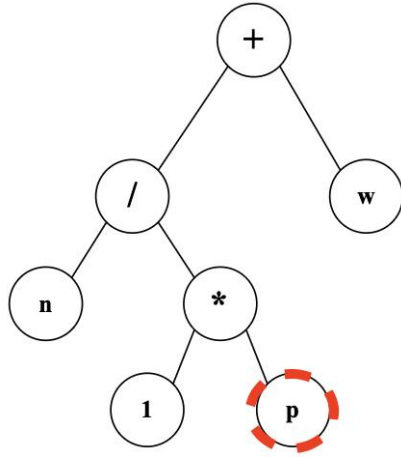
$$fit(x_i) = \begin{cases} \frac{1}{1 + f(x_i)} & \text{eğer } f(x_i) \geq 0 \\ 1 + abs(f(x_i)) & \text{eğer } f(x_i) < 0 \end{cases} \quad (2.3)$$

$$fit(x_i) = \frac{1}{1 + f(x_i)} \quad (2.4)$$

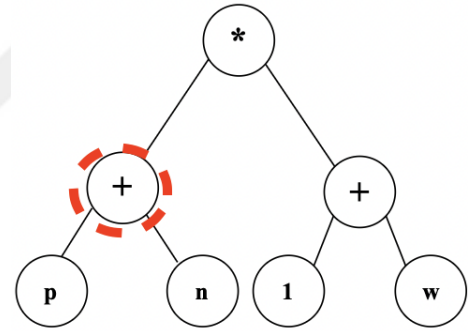
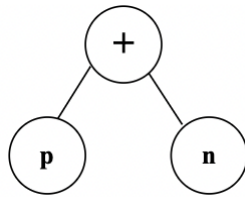
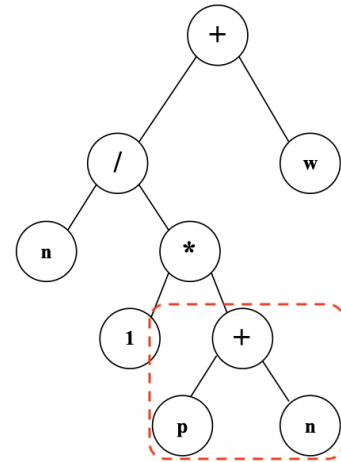
Gözcü arılar tercih edecekleri çözümleri, kalite değerleri ile belirlenen olasılıkları kullanarak seçmektedir. Olasılıkların belirlenmesinde Eşitlik 2.5 kullanılmaktadır. Fakat bu işlem için farklı fonksiyonlar da kullanılabilir. Eşitlikte yer alan alfa değeri $[0,1]$ arasında değer almaktadır.

$$P_i = \frac{\alpha \times fit(x_i)}{fit(x_{best})} + (1 - \alpha) \quad (2.5)$$

ABCP için aday çözüm üretmede kullanılan yöntem ABC'den farklı olacak şekilde geliştirilmiştir. Bilgi paylaşım mekanizması olarak adlandırılan yöntemde; mevcut çözümden (x_i) farklı olarak popülasyon içerisindeki farklı bir çözüm (x_k) bilgi paylaşımı için seçilmektedir. Daha önce belirlenen bir P_{ip} olasılığı dikkate alınarak fonksiyon veya terminal kümelerinden biri seçilmektedir. Seçilen düğüm x_k 'nin x_i ile paylaşacağı bilginin miktarını ve içeriğini belirlemektedir. Şekil 2.10'da verilen mevcut çözümdeki (x_i) seçilen düğümün yerine Şekil 2.11'da verilen popülasyon içerisindeki farklı bir çözümden (x_k) seçilen düğüm yerleştirilerek aday çözüm üretilmektedir. x_k 'nin x_i ile paylaşacağı bilgi Şekil 2.12'de verilmektedir. Şekil 2.13'te ise yeni üretilen aday çözüm v_i verilmektedir.



Şekil 2.10: Mevcut çözüm

Şekil 2.11: Popülasyondaki farklı bir çözüm x_k Şekil 2.12: x_k 'nin paylaşacağı bilgiŞekil 2.13: Üretilen aday çözüm v_i

3. BÖLÜM

DENEYSEL ÇALIŞMALAR

Bu bölümde ilk olarak kalabalık tahliyesi için etmen tabanlı simülasyon ile elde edilen veriler üretilirken yapılan deney çalışmalarına yer verilmektedir. Daha sonra, üretilen veriler üzerinde ABCP ile yapılan sembolik regresyon çalışmaları ve bu çalışmalar sonucu elde edilen çıktılar paylaşılmaktadır.

3.1 Etmen Tabanlı Simülasyon ile İlgili Parametrelerin Ayarları ve Deneyler

Çalışma kapsamında; kapı sayısı, kapı genişliği, insan sayısı gibi parametrelerin kalabalığın tahliye süresi ile ilişkisi incelenmektedir. Bu parametrelerin aldığı farklı değerler ve bu değerlerin çeşitli kombinasyonları simülasyon ortamında test edilmiş, üretilen veriler ile sembolik regresyon çalışması yapılmıştır. Tahliye süresi ile ilişkisi incelenecek parametreler ve bu parametrelerin değerleri Tablo 3.1’de sunulmaktadır.

Tablo 3.1: Simülasyonda kullanılan parametre değerleri

Parametre	Değerler
Kapı sayısı (n)	2, 3, 4, 5, 6
Kapı genişliği (w)	1m, 1,5m, 2m, 2,5m, 3m
İnsan sayısı (p)	75, 150, 225, 300, 375

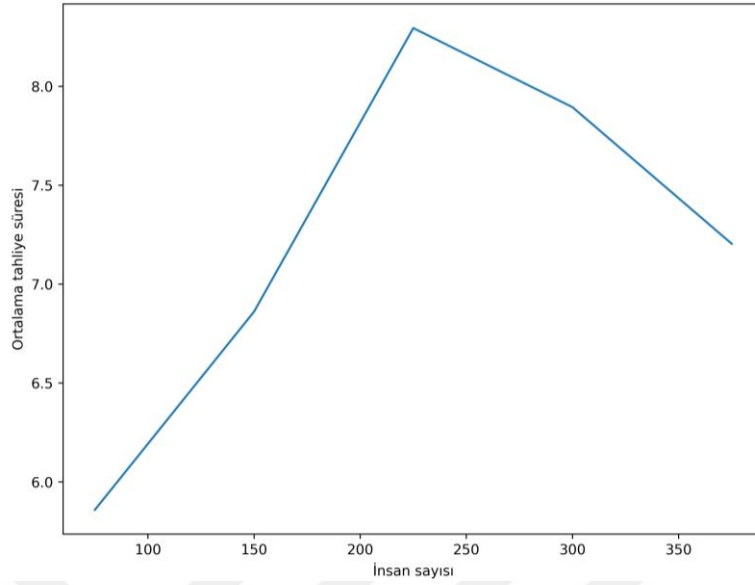
Temel AVM planında, 2 adet giriş ve çıkış kapısı bulunmaktadır. Bu yüzden tahliye anında çıkış için kullanılacak kapı sayısının alabileceği en düşük değer 2'dir. Acil çıkış kapılarının açık olma durumuna göre çıkış yapılabilecek kapı sayısı 6'ya kadar yükselmektedir. Kapı sayısına göre Şekil 2.2'de verilmiş olan numaralandırmalara uygun olarak açık olan kapılar Tablo 3.2'de verilmiştir.

Tablo 3.2: Kapı sayılarına göre açık olacak kapı numaraları

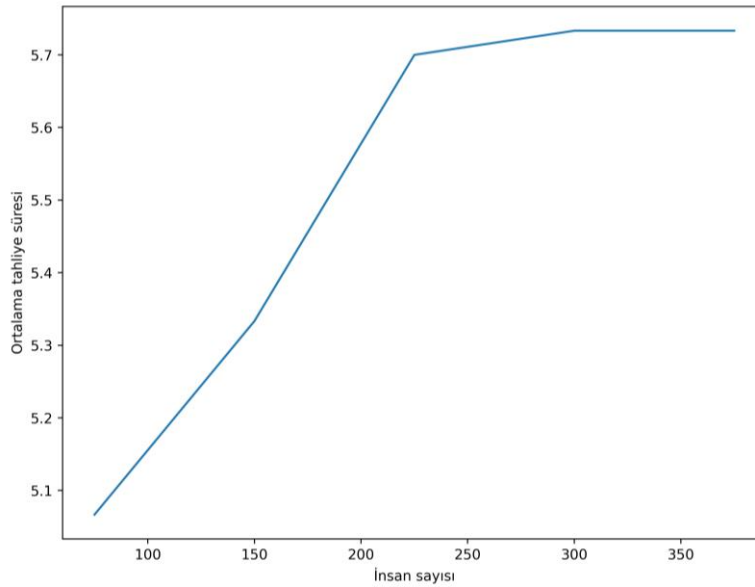
Kapı sayısı	Açık olan kapıların numarası
2	1,2
3	1, 2, 3
4	1, 2, 3, 4
5	1, 2, 3, 4, 5
6	1, 2, 3, 4, 5, 6

Kapı sayısı, kapı genişliği ve insan sayısı için dikkate alınan değerlerin bütün kombinasyonları ile ortaya çıkan 125 test durumu (senaryo) için sistem simüle edilmiştir. Her bir senaryo, simülasyon ortamında birbirinden bağımsız olarak 30 kez koşturulmuştur ve 30 koşmada elde edilen tahliye sürelerinin ortalamaları alınmıştır. Bu şekilde elde edilen veri seti Ek 3.1'de sunulmuştur.

Simülasyon deneylerinde kullanılan senaryolara ait parametre değerleri daha önce bahsedildiği gibi Tablo 3.1'de yer almaktadır. İnsan sayısına göre diğer bütün parametrelerin de dahil edilerek ortalama tahliye süresinin incelendiği grafik Şekil 3.1'de, insan sayısına göre diğer parametrelerin yalnızca ortanca değerlerinin ($w=2m$, $n=4$) dahil edilerek ortalama tahliye süresinin incelendiği grafik Şekil 3.2'de verilmektedir.



Şekil 3.1: Deney sonucu elde edilen insan sayısı - ortalama tahliye süresi grafiği (bütün değerler için)

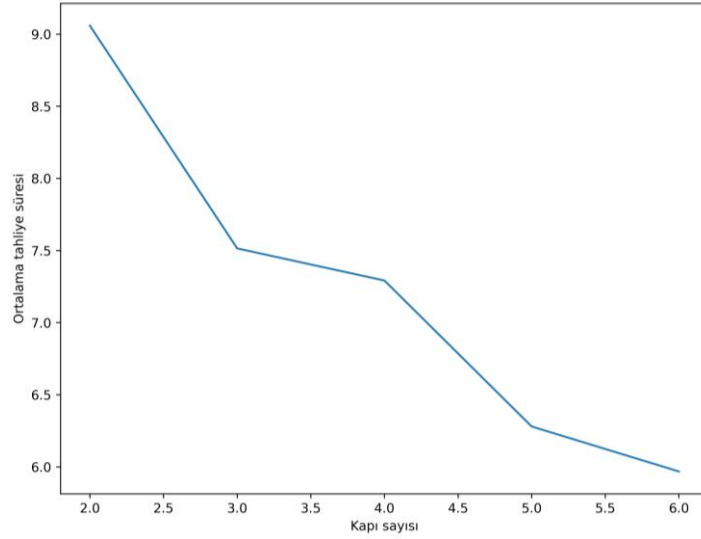


Şekil 3.2: Deney sonucu elde edilen insan sayısı - ortalama tahliye süresi grafiği (ortanca değerler için)

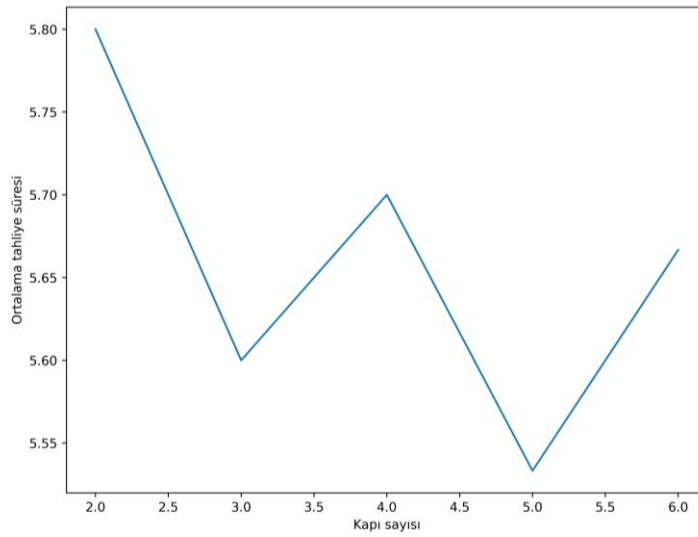
Şekil 3.1 ve Şekil 3.2 incelendiğinde insan sayısının artmasının genel olarak ortalama tahliye süresinde de artışa sebep olduğu gözlemlenebilmektedir.

Kapı sayısına göre diğer bütün parametrelerin de dahil edilerek ortalama tahliye süresinin incelendiği grafik Şekil 3.3'te, kapı sayısına göre diğer parametrelerin yalnızca ortanca

değerlerinin ($w=2m$, $p=225$) dahil edilerek ortalama tahliye süresinin incelendiği grafik Şekil 3.4'te verilmektedir.



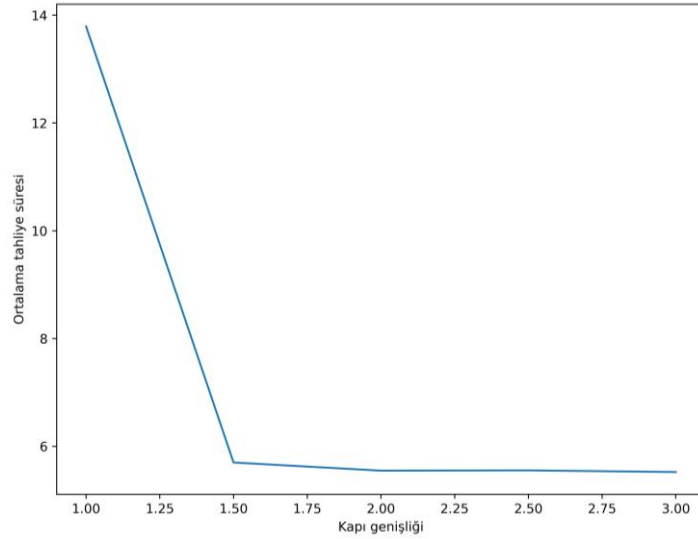
Şekil 3.3: Deney sonucu elde edilen kapı sayısı - ortalama tahliye süresi grafiği (bütün değerler için)



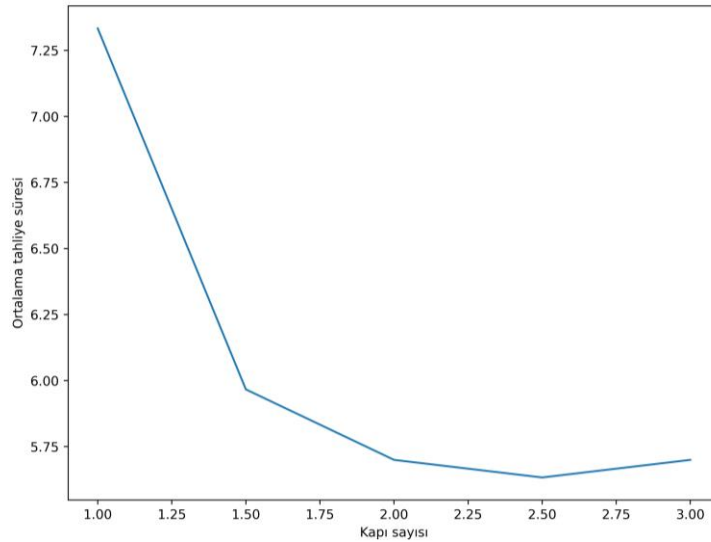
Şekil 3.4: Deney sonucu elde edilen kapı sayısı - ortalama tahliye süresi grafiği (ortanca değerler için)

Şekil 3.3 ve Şekil 3.4 incelendiğinde kapı sayısının artmasının genel olarak ortalama tahliye süresinde azalmaya sebep olduğu gözlemlenmiştir.

Kapı genişliğine göre diğer bütün parametrelerin de dahil edilerek ortalama tahliye süresinin incelendiği grafik Şekil 3.5'te, kapı genişliğine göre diğer parametrelerin yalnızca ortanca değerlerinin ($n= 4$, $p=225$) dahil edilerek ortalama tahliye süresinin incelendiği grafik Şekil 3.6'da verilmektedir.



Şekil 3.5: Deney sonucu elde edilen kapı genişliği - ortalama tahliye süresi grafiği
(bütün değerler için)



Şekil 3.6: Deney sonucu elde edilen kapı genişliği - ortalama tahliye süresi grafiği
(ortanca değerler için)

Şekil 3.5 ve Şekil 3.6 incelendiğinde kapı genişliğinin artmasının genel olarak ortalama tahliye süresinde azalmaya sebep olduğu gözlemlenebilmektedir.

3.2 ABCP ile İlgili Parametrelerin Ayarları ve Deneyler

Bir önceki bölümde belirtildiği gibi çalışma kapsamında; kapı sayısı, kapı genişliği, insan sayısı gibi parametrelerin kalabalığın tahliye süresi ile ilişkisi incelenmektedir. Simülasyon sonucunda elde edilen veriler üzerinde sembolik regresyon çalışması yapılırken ABCP kullanılarak bu ilişki modellenmeye çalışılmıştır. ABCP ile elde edilen modellerde olası bağımsız değişkenler insan sayısı, kapı sayısı, kapı genişliği; bağımlı değişken ise tahliye süresidir.

ABCP için kullanılan parametre değerleri Tablo 3.3'te verilmektedir. Koloni büyüklüğü ve limit parametreleri 500, durdurma kriteri maksimum iterasyon sayısı 500 olacak şekilde alınmıştır. Çözümlerin maliyet hesabında bütün veriler kullanılmıştır. Fonksiyon kümesinde, bölen değeri 0'a eşitse işlemin sonucunu 1 olarak döndüren korunmuş "/" fonksiyonu kullanılmaktadır. Korunmuş bu fonksiyonda bölen değer 0 değilse işlem normal bölme işlemi olarak gerçekleştirilir. Diğer fonksiyonlar da normal olarak toplama, çıkarma ve çarpma işlemlerini yerine getirmektedir. Terminal kümesinde olası bağımsız değişkenler olan sistem giriş değişkenleri; kapı sayısı, kapı genişliği ve insan sayısının yanında sabit bir sayı olarak 1 ve $[-1,1]$ aralığında rastgele üretilmiş bir sayıyı temsil eden R de kullanılmıştır. Ağaç yapısının başlangıç maksimum derinliği 6 iken maksimum derinliği bir deney grubunda 6, diğerinde 15 olarak belirlenmiştir.

Daha önce de belirtildiği gibi üretilen çözümlerin değerlendirilmesinde Eşitlik 2.2 ile formüle edilen toplam mutlak hata dikkate alınmış ve optimizasyon sürecinde bu maliyet değerinin minimizasyonu yapılmıştır. Simülasyon verilerinin tamamı çözümlerin uygunluk hesabında kullanıldığı için Eşitlik 2.2'de $Ne = 125$ değerini almıştır.

Tablo 3.3: ABCP yöntemine ait parametre değerleri

Parametre	Değerler
Koloni Büyüklüğü	500
Limit	500
Ham Uygunluk Değeri (maliyet)	Tüm uygunluk durumları için mutlak hata değerlerinin toplamı
Terminal Kümesi	n, w, p, R, 1
Fonksiyon Kümesi	+, -, *, / (korunmuş versiyon)
Başlangıç Maksimum Derinliği	6
Maksimum Derinlik	6, 15
Durdurma Kriteri	Maksimum iterasyon sayısı
Maksimum İterasyon Sayısı	500
Koşma Sayısı	30

Maksimum ağaç derinliği için 6 ve 15 değerlerinin kullanıldığı 2 deney grubunun her biri için birbirinden bağımsız olarak 30 defa oluşturulmuş olan ABCP'den elde edilen maliyet fonksiyonu değerlerinin ortalaması, ilgili standart sapma, bu değerler arasındaki en iyi ve en kötü Tablo 3.4'te sunulmuştur. Elde edilen en iyi değere sahip olan modellerin ağaç yapısından matematiksel ifadeye sadeleştirilmeden çevrilmiş hali maksimum 15 ve 6 derinlik için sırasıyla Şekil 3.7 ve Şekil 3.8'de verilmektedir. Sadeleştirilmiş halleri ise yine aynı sıralama ile Eşitlik 3.1 ve 3.2'de gösterilmektedir. Maksimum derinlik değeri 6 olan en iyi değere sahip modelin ağaç yapısında temsili Şekil 3.9'da verilmektedir. Bağımsız koşmalar sonucu her bir iterasyonda bulunan maliyet fonksiyonu değerlerinin

ortalamasını veren yakınsama grafiği ise Şekil 3.10'da maksimum derinlik değerinin 6 olduğu durum için, Şekil 3.4'te de 15 olduğu durum için sunulmuştur.

Tablo 3.4: ABCP deney sonuçları

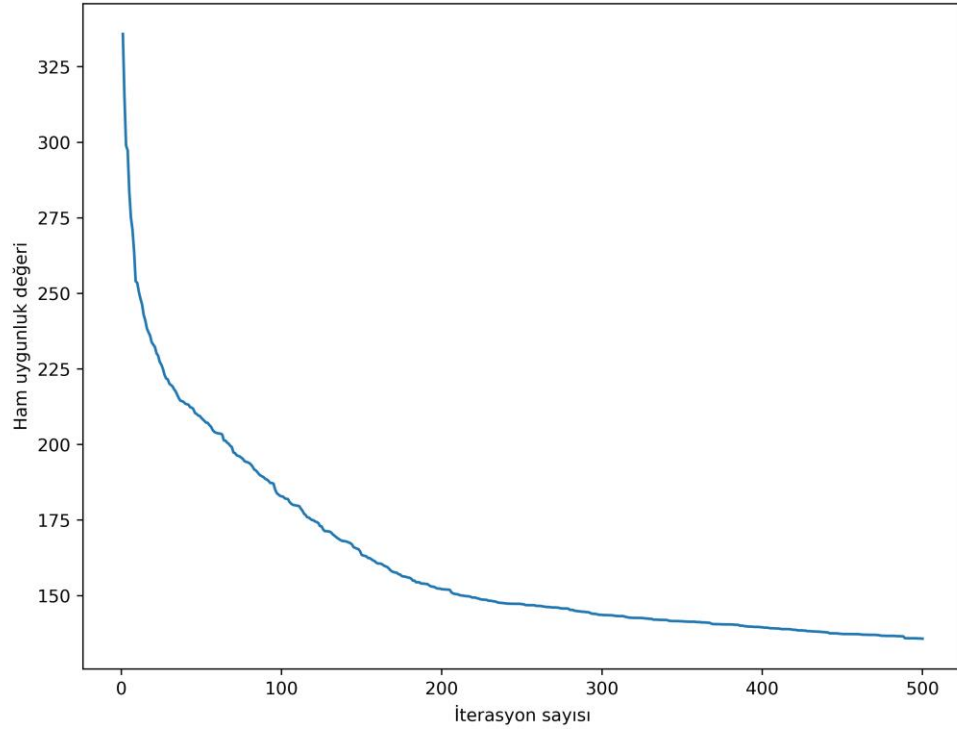
	Maksimum derinlik=6	Maksimum derinlik=15
Ortalama	175.6230	135.7736
Standart sapma	22.6402	33.3540
En iyi	138.4318	113.8034
En kötü	227.8590	153.0763

((-0.12903094716790642)+((((((((((0.9144674790624843)/(0.15697968711935895))*((p)/(p)))*(((p)/(p))-(((0.9144674790624843)/(0.15697968711935895))/(p))))*(((p)/(p))-((n)/(p))))-(0.021825033715844544))-((0.021825033715844544))-((p)/((((0.9144674790624843)-(w))*((n)*(((0.9144674790624843)/(0.15697968711935895))*((0.9333000192108099)-(((0.9144674790624843)/(0.15697968711935895))/(p)))))))*(((0.9144674790624843)/(0.15697968711935895))*((0.9333000192108099)))))))*(((p)/(p))-((((p)/(p))/(p))/(p))))-(p)/((((0.15697968711935895)-(w))*((n)*(((0.9144674790624843)/(0.15697968711935895))-((0.021825033715844544))*(((0.9333000192108099)/(0.15697968711935895))*((n)/(p))+((w))*((0.9333000192108099)-(((p)/(p))-((p)/(p)))/(p))))-(p)/(p))))-(p)/(p))-((0.9144674790624843)-(w)))))*((((((p)/(p))-((n)/(p)))-((n)/(p)))-((((0.9333000192108099)/(0.15697968711935895))*((p)/(p)))/(0.15697968711935895))*(((0.9144674790624843)/(0.15697968711935895))*((0.9333000192108099))/(p)))-(((n)/(((p)/(p))-((w)))-((w))*((0.9144674790624843)/(0.15697968711935895)))))))-((0.021825033715844544)))

Şekil 3.7: Deney sonucu elde edilen en iyi değere sahip model (maksimum derinlik=15)

((((((0.23367468138862146)/(0.035000474674161675))+0.9295335225432801))-((0.23367468138862146)+((1)+(n))))/((-0.3019073863056988)/(0.4608421993725198))-(((0.8001141230576272)+(w))/(0.035000474674161675))-((0.23367468138862146)/(0.035000474674161675))))+(((1)/((p)+((0.23367468138862146)/(0.035000474674161675)))))*(((p)-(n))*((1))*(((0.23367468138862146)/(0.035000474674161675))-((n)/((1)+(n))+0.5621625061902042))))))

Şekil 3.8: Deney sonucu elde edilen en iyi değere sahip model (maksimum derinlik=6)



Şekil 3.11: ABCP'nin yakınsama performansı (maksimum derinlik=15).

4. BÖLÜM

TARTIŞMA-SONUÇ ve ÖNERİLER

4.1. Tartışma ve Sonuç

Yapılan bu tez çalışmasında kalabalık tahliyesi için etmen tabanlı simülasyon ile elde edilen veriler üzerinde sembolik regresyon çalışması yapılmıştır. Bu kapsamda kalabalığın tahliye süresi ile AVM'ye ait kapı sayısı, kapı genişliği, insan sayısı gibi parametrelerin ilişkisi incelenmek istenmiş ve ABCP ile bu ilişkinin modeli oluşturulmaya çalışılmıştır. Parametreler arasındaki ilişkinin incelenmesi için AnyLogic simülasyon programı ile bir AVM kalabalık tahliyesi modellenmiş, yapılan deneyler sonucu doğal yollarla elde edilmesi zor ve maliyetli olan veriler üretilmiştir. Elde edilen veriler kullanılarak başlangıç maksimum derinlik değeri aynı (maks. 6) olan fakat maksimum derinlikleri için sırasıyla 6 ve 15 değerlerinin kullanıldığı 2 deney grubu test edilmiştir.

Yapılan deneyler sonucunda maliyet fonksiyon değerlerine göre, maksimum derinlik değeri 15 olan modellerin maksimum derinlik değeri 6 olan modellere göre daha iyi sonuçlar ürettiği Tablo 3.4'te gözlemlenmiştir. Maliyet fonksiyon değerlerine göre maksimum derinlik değeri 15 olan modellerin en kötü değeri, maksimum derinlik değeri 6 olan modellerin ortalama değerinden daha iyi sonuç verdiği yine Tablo 3.4'te görülmektedir.

Tablo 3.4 incelendiğinde en iyi, en kötü ve ortalama değerlerin her birinde de maksimum derinlik değeri 15 olan modellerin maliyet fonksiyon değerlerine bakıldığında daha iyi sonuçlar verdiği gözlemlenmiştir.

İlk 25 iterasyonda maksimum derinlik değeri 6 olan modeller daha hızlı yakınsarken, sonraki iterasyonlarda maksimum derinlik değeri 15 olan modellerin yakınsama

performansının daha ön plana geçtiği Şekil 3.10 ve Şekil 3.11’de verilen yakınsama grafikleri üzerinde gözlemlenmiştir. Maksimum derinlik değeri 15 olan modeller 500 iterasyon içerisinde 372.7 ham uygunluk değerinden, 113.8034 ham uygunluk değerine yakınsarken; maksimum derinlik değeri 6 olan modeller 500 iterasyon içerisinde 372.7 ham uygunluk değerinden, 138.4318 ham uygunluk değerine yakınsamıştır.

Model başarıları kıyaslandığında; $[-1, 1]$ arasında rastgele bir değeri temsil eden R modellerde fazlaştıkça sonuçlarda iyileşmeler gözlemlenmiştir. Modeller ve modellere ait ham uygunluk değerleri Ek 2.1’de sunulmaktadır.

Elde edilen modeller (Ek 2.1’de yer alan) incelendiğinde daha karmaşık yapıda olan modellerin daha iyi sonuç verebildiği gözlemlenmiştir. Bu sonuç bu problem için diğer ABCP parametrelerinde değişiklik yapılmadığı durumda sistem modelinin daha az hata ile elde edilmesinde ABCP’nin 6’dan büyük ağaç derinlik değerlerine ihtiyacı olduğunu göstermektedir.

Yapılan deneyler sonucunda kullanılan yöntemin model oluştururken özellik seçimi yaparak verilen giriş değişkenleri arasında tercihlerde bulunabildiği görülmektedir. Bağımsız koşmalar sonucunda belirlenen en iyi modeller incelendiğinde, çoğunluğunda 3’ü de kullanılmakla beraber ABCP’nin 2 veya 1 giriş değişkenini bağımsız değişken olarak seçip elde ettiği modeller (örneğin sadece kapı genişliğinin kullanıldığı EK 1.1 Koşma 0’da bulunan model) de tespit edilmiştir. Ayrıca EK 1.1’de bulunan Koşma 0, 4, 10, 12, 14, 15, 16, 18, 19, 20 ve EK 2.1’de bulunan Koşma 6, 8, 16, 25; 2 giriş değişkenini bağımsız değişken olarak kullanan modellere örnektir.

4.2.Öneriler

Gelecekte yapılacak olan çalışmalarda ABCP modelinin yakınsama hızını artırmak için qABCP temelli yaklaşımlar kullanılabilir. Daha iyi sonuçlar elde etmek amacıyla uygun fonksiyon ve terminal kümelerinin seçimi için araştırmalar yapılabilir. ABCP kontrol parametre değerlerinin belirlenmesinde bir deney tasarımı çalışması yapılabilir. Simülasyon çalışmalarında moloz yığılması, ateşin yolu kapatması gibi daha spesifik etkenler kullanılarak kalabalık tahliyesi için yeni parametreler analiz edilebilir ve model zenginleştirilebilir. Yapılan tez çalışmasının kalabalık tahliyesinde ABCP’nin

kullanıldığı ilk çalışma olması yönüyle bu alandaki birçok vaka çalışmalarına da temel oluşturulabileceği düşünülmektedir.



KAYNAKÇA

1. Kaur, N., Kaur, H., 2022. A multi-agent based evacuation planning for disaster management: a narrative review. **Archives of Computational Methods in Engineering**. <https://doi.org/10.1007/s11831-022-09729-4>
2. Liu, J., Chen, Y., Chen, Y., 2021. Emergency and disaster management-crowd evacuation research. **Journal of Industrial Information Integration**, **21**: 100191. <https://doi.org/10.1016/j.jii.2020.100191>
3. Cao, M., Zhang, G., Wang, M., Lu, D., Liu, H., 2017. A method of emotion contagion for crowd evacuation. **Physica A: Statistical Mechanics and Its Applications**, **483**: 250-258. <https://doi.org/10.1016/j.physa.2017.04.137>
4. Kalogeiton, V. S., Papadopoulos, D. P., Georgilas, I. P., Sirakoulis, G. Ch., Adamatzky, A. I., 2015. Cellular automaton model of crowd evacuation inspired by slime mould. **International Journal of General Systems**, **44**(3): 354-391. <https://doi.org/10.1080/03081079.2014.997527>
5. Hassanpour, S., Gonzalez, V., Liu, J., Zou, Y., Cabrera-Guerrero, G., 2022. A hybrid hierarchical agent-based simulation approach for buildings indoor layout evaluation based on the post-earthquake evacuation. **Advanced Engineering Informatics**, **51**: 101531. <https://doi.org/10.1016/j.aei.2022.101531>
6. Karaboga, D., Ozturk, C., Karaboga, N., Gorkemli, B., 2012. Artificial bee colony programming for symbolic regression. **Information Sciences**, **209**: 1-15. <https://doi.org/10.1016/j.ins.2012.05.002>
7. Gorkemli, B., Karaboga, D., 2019. A quick semantic artificial bee colony programming (qsABCP) for symbolic regression. **Information Sciences**, **502**: 346-362. <https://doi.org/10.1016/j.ins.2019.06.052>
8. Koza, JohnR., 1994. Genetic programming as a means for programming computers by natural selection. **Statistics and Computing**, **4**(2). <https://doi.org/10.1007/BF00175355>
9. Karaboga, D. (t.y.). An idea based on honey bee swarm for numerical optimization. 10.
10. Öztürk, C., Tarım, M., Arslan, S., 2020. Feature selection and classification of metabolomics data using artificial bee colony programming (ABCP).

- International Journal of Data Mining and Bioinformatics**, **23**(2): 101.
<https://doi.org/10.1504/IJDMB.2020.107378>
11. Boudardara, F., Gorkemli, B., 2020. Solving artificial ant problem using two artificial bee colony programming versions. **Applied Intelligence**, **50**(11): 3695-3717. <https://doi.org/10.1007/s10489-020-01741-0>
 12. Axelrod, R. M., 1997. The complexity of cooperation: agent-based models of competition and collaboration. **Princeton University Press**.
 13. Agent-based model, (2022). (Web sayfası: https://en.wikipedia.org/wiki/Agent-based_model), (Erişim tarihi: Nisan 2022).
 14. Zhong, J., Luo, L., Cai, W., Lees, M., 2014. Automatic rule identification for agent-based crowd models through gene expression programming, 1125–1132. *AAMAS: International Conference on Autonomous Agents and Multiagent Systems*, May 5 – 9, Paris, France.
 15. Mahmood, I., Nadeem, T., Bibi, F., Hu, X., 2019. Analyzing emergency evacuation strategies for large buildings using crowd simulation framework, 3116-3127. *2019 Winter Simulation Conference (WSC)*. December 8-11, National Harbor, Maryland <https://doi.org/10.1109/WSC40007.2019.9004906>
 16. Lopez-Carmona, M. A., Paricio Garcia, A., 2021. CellEVAC: an adaptive guidance system for crowd evacuation through behavioral optimization. **Safety Science**, **139**: 105215. <https://doi.org/10.1016/j.ssci.2021.105215>
 17. Mahmood, I., Haris, M., Sarjoughian, H., 2017. Analyzing emergency evacuation strategies for mass gatherings using crowd simulation and analysis framework: hajj scenario. **Proceedings of the 2017 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation**, 231-240. <https://doi.org/10.1145/3064911.3064924>
 18. Daoliang, Z., Lizhong, Y., Jian, L., 2006. Exit dynamics of occupant evacuation in an emergency. **Physica A: Statistical Mechanics and Its Applications**, **363**(2): 501-511. <https://doi.org/10.1016/j.physa.2005.08.012>
 19. Liu, S., Yang, L., Fang, T., Li, J., 2009. Evacuation from a classroom considering the occupant density around exits. **Physica A: Statistical Mechanics and Its Applications**, **388**(9): 1921-1928. <https://doi.org/10.1016/j.physa.2009.01.008>
 20. Helbing, D., Farkas, I., Vicsek, T., 2000. Simulating dynamical features of escape panic. **Nature**, **407**(6803): 487-490. <https://doi.org/10.1038/35035023>

21. Hughes, R. L., 2003. The flow of human crowds. **Annual Review of Fluid Mechanics**, **35**(1): 169-182. <https://doi.org/10.1146/annurev.fluid.35.101101.161136>
22. ZARBOUTIS, N., MARMARAS, N., 2004. Searching efficient plans for emergency rescue through simulation: the case of a metro fire. **Cognition, Technology & Work**, **6**(2): 117-126. <https://doi.org/10.1007/s10111-004-0150-6>
23. Toyama, M.C., Bazzan A.L.C., Silva, R., 2006. An agent-based simulation of pedestrian dynamics: from lane formation to auditorium evacuation, 108-110. *AAMAS: International Conference on Autonomous Agents and Multiagent Systems*, 8 - 12 May, Hakodate, Japan.
24. Luo, L., Zhou, S., Cai, W., Lees, M., Low, M. Y. H., Sornum, K., 2011. HumDPM: a decision process model for modeling human-like behaviors in time-critical and uncertain situations. İçinde M. L. Gavrilova, C. J. K. Tan, A. Sourin, & O. Sourina (Ed.), *Transactions on Computational Science XII* (C. 6670, ss. 206-230). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-22336-5_11
25. Zhong, J., Luo, L., Cai, W., Lees, M., 2014. EA-based evacuation planning using agent-based crowd simulation, 395-406. *Proceedings of the Winter Simulation Conference 2014*. December 7-10, Savannah, Georgia. <https://doi.org/10.1109/WSC.2014.7019906>
26. Zhong, J., Cai, W., Luo, L., 2015. Crowd evacuation planning using Cartesian Genetic Programming and agent-based crowd modeling, 127-138. *2015 Winter Simulation Conference (WSC)*, December 6-9, Huntington Beach, California <https://doi.org/10.1109/WSC.2015.7408158>
27. Haris, M., Mahmood, I., Badar, M., Alvi, M. S. Q., 2018. Modeling safest and optimal emergency evacuation plan for large-scale pedestrians environments, 917-928. *2018 Winter Simulation Conference (WSC)*, December 9-12, Gothenburg, Sweden. <https://doi.org/10.1109/WSC.2018.8632418>
28. Zong, X., Liu, A., Wang, C., Ye, Z., Du, J., 2022. Indoor evacuation model based on visual-guidance artificial bee colony algorithm. **Building Simulation**, **15**(4): 645-658. <https://doi.org/10.1007/s12273-021-0838-z>
29. Görkemli, B., 2019. Yapay arı koloni programlama (ABCP) yöntemlerinin geliştirilmesi ve sembolik regresyon problemlerine uygulanması. Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, Kayseri.

30. Karaboga, D., Gorkemli, B., Ozturk, C., Karaboga, N., 2014. A comprehensive survey: Artificial bee colony (ABC) algorithm and applications. **Artificial Intelligence Review**, **42**(1): 21-57. <https://doi.org/10.1007/s10462-012-9328-0>
31. Karaboga, D., Akay, B., 2009. A comparative study of Artificial Bee Colony algorithm. **Applied Mathematics and Computation**, **214**(1): 108-132. <https://doi.org/10.1016/j.amc.2009.03.090>
32. Akay, B., Karaboga, D., Gorkemli, B., Kaya, E., 2021. A survey on the Artificial Bee Colony algorithm variants for binary, integer and mixed integer programming problems. **Applied Soft Computing**, **106**: 107351. <https://doi.org/10.1016/j.asoc.2021.107351>
33. Golafshani, E. M., Ashour, A., 2016. Prediction of self-compacting concrete elastic modulus using two symbolic regression techniques. **Automation in Construction**, **64**: 7-19. <https://doi.org/10.1016/j.autcon.2015.12.026>
34. Arslan, S., Ozturk, C., 2019. Multi Hive Artificial Bee Colony Programming for high dimensional symbolic regression with feature selection. **Applied Soft Computing**, **78**: 515-527. <https://doi.org/10.1016/j.asoc.2019.03.014>
35. Boudardara, F., Gorkemli, B., 2018. Application of Artificial Bee Colony Programming to Two Trails of the Artificial Ant Problem., 1-6. *2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, October 20-22, Ankara, Türkiye. <https://doi.org/10.1109/ISMSIT.2018.8567048>
36. Arslan, S., Öztürk, C., 2019. A Comparative Study of Automatic Programming Techniques. **Informatica**, **43**(2): 281-289. <https://doi.org/10.31449/inf.v43i2.2133>
37. Hara, A., Kushida, J.-I., Takemoto, R., Takahama, T., 2018. Artificial Bee Colony Programming Using Semantic Control Crossover, 189-194. *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, October 7-10, Japan. <https://doi.org/10.1109/SMC.2018.00043>
38. Yassine, B., Hacene, H., 2018. Scaled artificial bee colony programming, 1-5. *2018 International Conference on Applied Smart Systems (ICASS)*, November 24-25, Algeria. <https://doi.org/10.1109/ICASS.2018.8651986>

39. Arslan, S., Ozturk, C., 2019. Artificial Bee Colony Programming Descriptor for Multi-Class Texture Classification. **Applied Sciences**, **9**(9): 1930. <https://doi.org/10.3390/app9091930>
40. Bozogullarindan, C., Ozturk, C., 2019. Cloud-based Artificial Bee Colony Programming Web Application, 1-5. 2019 *Innovations in Intelligent Systems and Applications Conference (ASYU)*, 31 Ekim – 2 Kasım, İzmir, Türkiye. <https://doi.org/10.1109/ASYU48272.2019.8946344>
41. Bozogullarindan, E., Bozogullarindan, C., Ozturk, C., 2020. Transfer Learning in Artificial Bee Colony Programming, . 1-7. 2020 *Innovations in Intelligent Systems and Applications Conference (ASYU)*, 15-17 Ekim, İstanbul, Türkiye. <https://doi.org/10.1109/ASYU50717.2020.9259801>
42. Boudouaoui, Y., Habbi, H., Ozturk, C., Karaboga, D., 2020. Solving differential equations with artificial bee colony programming. **Soft Computing**, **24**(23), 17991-18007. <https://doi.org/10.1007/s00500-020-05051-y>
43. Nekoei, M., Moghaddas, S. A., Mohammadi Golafshani, E., Gandomi, A. H., 2021. Introduction of ABCEP as an automatic programming method. **Information Sciences**, **545**: 575-594. <https://doi.org/10.1016/j.ins.2020.09.020>
44. Abbasianjahromi, H., Mohammadi Golafshani, E., Aghakarimi, M., 2022. A prediction model for safety performance of construction sites using a linear artificial bee colony programming approach. **International Journal of Occupational Safety and Ergonomics**, **28**(2), 1265-1280. <https://doi.org/10.1080/10803548.2021.1889878>
45. Özger, Z. B., Yağci, A. E., Unsal, M., 2021. Estimating The Aeration Performance Of Venturi-Conduit By Artificial Bee Colony Programming. **Journal of Polytechnic**. **25**(1): 389 – 398. <https://doi.org/10.2339/politeknik.972844>
46. Görkemli, B., Citakoglu, H., Haktanir, T., Karaboga, D., 2022. A new method based on artificial bee colony programming for the regional standardized intensity–duration–frequency relationship. **Arabian Journal of Geosciences**, **15**(3): 1-19. <https://doi.org/10.1007/s12517-021-09377-1>

EKLER

EK 1.

Kapı genişliği	Kapı sayısı	İnsan sayısı	Ortalama Tahliye süresi (dakika)
----------------	-------------	--------------	----------------------------------

1	2	75	8.0333
---	---	----	--------

1	2	150	33.0000
---	---	-----	---------

1	2	225	29.1667
---	---	-----	---------

1	2	300	19.4667
---	---	-----	---------

1	2	375	21.8667
---	---	-----	---------

1	3	75	9.4667
---	---	----	--------

1	3	150	9.4000
---	---	-----	--------

1	3	225	20.8333
---	---	-----	---------

1	3	300	22.3333
---	---	-----	---------

1	3	375	12.4000
---	---	-----	---------

1	4	75	16.4333
---	---	----	---------

1	4	150	7.2333
---	---	-----	--------

1	4	225	7.3333
---	---	-----	--------

1	4	300	22.3000
---	---	-----	---------

1	4	375	15.8000
---	---	-----	---------

1	5	75	5.2333
---	---	----	--------

1	5	150	5.9333
1	5	225	19.3000
1	5	300	10.7667
1	5	375	6.6000
1	6	75	5.0333
1	6	150	6.3333
1	6	225	16.9667
1	6	300	6.5667
1	6	375	6.9667
1.5	2	75	5.2333
1.5	2	150	5.8333
1.5	2	225	5.9000
1.5	2	300	6.0333
1.5	2	375	6.2667
1.5	3	75	5.3000
1.5	3	150	5.6667
1.5	3	225	5.7667
1.5	3	300	5.8000
1.5	3	375	6.2333
1.5	4	75	5.1333

1.5	4	150	5.6000
1.5	4	225	5.9667
1.5	4	300	7.0333
1.5	4	375	6.1667
1.5	5	75	4.9667
1.5	5	150	5.3333
1.5	5	225	5.7667
1.5	5	300	5.7333
1.5	5	375	5.8000
1.5	6	75	5.0333
1.5	6	150	5.3333
1.5	6	225	5.5333
1.5	6	300	5.5333
1.5	6	375	5.5000
2	2	75	4.9667
2	2	150	5.6667
2	2	225	5.8000
2	2	300	6.1000
2	2	375	5.9000
2	3	75	5.4000

2	3	150	5.6000
2	3	225	5.6000
2	3	300	5.9333
2	3	375	5.9000
2	4	75	5.0667
2	4	150	5.3333
2	4	225	5.7000
2	4	300	5.7333
2	4	375	5.7333
2	5	75	5.1667
2	5	150	5.4000
2	5	225	5.5333
2	5	300	5.5667
2	5	375	5.6333
2	6	75	4.6333
2	6	150	5.5000
2	6	225	5.6667
2	6	300	5.4667
2	6	375	5.7000
2.5	2	75	5.3333

2.5	2	150	5.6333
2.5	2	225	5.7333
2.5	2	300	5.9667
2.5	2	375	6.1000
2.5	3	75	5.2333
2.5	3	150	5.6333
2.5	3	225	5.8333
2.5	3	300	5.7667
2.5	3	375	5.6000
2.5	4	75	5.1333
2.5	4	150	5.4667
2.5	4	225	5.6333
2.5	4	300	5.8333
2.5	4	375	5.8333
2.5	5	75	5.1000
2.5	5	150	5.3667
2.5	5	225	5.6667
2.5	5	300	5.6667
2.5	5	375	5.6667
2.5	6	75	4.9333

2.5	6	150	5.2000
2.5	6	225	5.4000
2.5	6	300	5.5000
2.5	6	375	5.6000
3	2	75	5.1667
3	2	150	5.6333
3	2	225	5.7333
3	2	300	5.9333
3	2	375	6.0000
3	3	75	5.2000
3	3	150	5.5667
3	3	225	5.7000
3	3	300	5.8333
3	3	375	5.8667
3	4	75	5.2000
3	4	150	5.4000
3	4	225	5.7000
3	4	300	5.6333
3	4	375	5.9000
3	5	75	5.1000

3	5	150	5.1333
3	5	225	5.5667
3	5	300	5.4000
3	5	375	5.6000
3	6	75	4.9667
3	6	150	5.3333
3	6	225	5.5667
3	6	300	5.4667
3	6	375	5.4667

EK 2.**EK 2.1**

Koşma:0

Ham uygunluk değeri:133.881057

$$\begin{aligned}
& ((((((1)/((1)*(((1)+((-0.7538625149772793)/((1)*(1))))*(w)*(w))*((w)*(w))+((-0.7538625149772793)+(-0.7439429972991082))/(n)))))*((1)*((w)*(w))*((w))))+((0.0660311482222895)/(p)) \\
&)+(((((((1)/(n))+((-0.9213953921205342)/((0.0660311482222895)+(-0.22914044243709208)))/(n)+(p))*((p)))/((p)*(1))*((p))/(1)/(1)))/(((1)/((0.0660311482222895)*((w)*(w))*((w)*((1)/(1)*(n))*((1)*(w)))+(w)/(n)))))+((((-0.9213953921205342)/((0.0660311482222895)+(-0.22914044243709208)))-((((-0.22914044243709208)+(-0.22914044243709208))+(-0.7439429972991082)))/(((0.8728893389333455)/((0.0660311482222895)+(-0.22914044243709208)))+((-0.7538625149772793)+(-0.7439429972991082))/(n))))+(-0.7439429972991082))/((p)/(p))))+(p))*((p))
\end{aligned}$$

Koşma:1

Ham uygunluk değeri:136.576258

$$\begin{aligned} & (((((((1)+(((((-0.9485170049353117)+(0.015270118143069666))*((1)/(- \\ & 0.3049118021991625))))+(1))+(((1)/((1)/(- \\ & 0.3049118021991625))))+((0.8156076110040804)+(((\\ & 0.9485170049353117)+(0.015270118143069666))*((1)/(- \\ & 0.3049118021991625)))))/(w))))+(1)+((((((1)/(n)))/((- \\ & 0.9485170049353117)+(0.015270118143069666))/((0.828881372617968)/(w)))))/(- \\ & 0.018646848862360677))+((1)/(n)))+(1)+((((((0.828881372617968)/(0.0152701181 \\ & 43069666))/(w))/(w))/((w)/(- \\ & 0.6957362423165405))/((0.828881372617968)/(w))/(w))))/(- \\ & 0.018646848862360677))/(n)))+(n))/((p)+(-0.3049118021991625))/((- \\ & 0.9485170049353117)+(0.015270118143069666))+(0.015270118143069666)))+((((\\ & 1)+((0.8156076110040804)+(((\\ & 0.9485170049353117)+(0.015270118143069666))*((1)/(- \\ & 0.3049118021991625)))))+(1))+(-0.7922638607175387))-((- \\ & 0.36998555593657567)/(w))))+((((((- \\ & 0.36998555593657567)/((1)+((0.8156076110040804)+((- \\ & 0.9485170049353117))*((1)/(-0.3049118021991625)))))+(w)/(- \\ & 0.6957362423165405))/((0.828881372617968)/(w))/(w))*((- \\ & 0.9485170049353117)/(-0.3049118021991625)))))/((w)/((- \\ & 0.9485170049353117)+(0.015270118143069666))/((0.828881372617968)/(w)))/(((\\ & 0.828881372617968)/(w))/(w))/(w))))/(- \\ & 0.018646848862360677))+((((((0.828881372617968)/(w))/((w)/(- \\ & 0.6957362423165405))/((0.828881372617968)/(w))/(w)))/((0.828881372617968)/((\\ & 1)/(n)))/(w))))/(- \\ & 0.018646848862360677))/(n))+((w)))+(1))/((1)+(w))/(n))+(((1)+(((1)/(n)))/((- \\ & 0.9485170049353117)+(0.015270118143069666))/((0.828881372617968)/(w)))/((1)/ \\ & ((1)/((n)/((1)+1)))/((0.828881372617968)/(w))/(w))/(w)/((- \\ & 0.36998555593657567)/(w)))))+(((- \\ & 0.9485170049353117)+(0.015270118143069666))/(n))+((0.8156076110040804)+(w) \end{aligned}$$

))))/(((0.828881372617968)/(w))/(w)))/(-
0.018646848862360677)))/((0.8156076110040804)+(w)))/(n)))

Koşma:2

Ham uygunluk değeri:135.164319

(((1)/(0.17776788267203036)))/((((0.808110045179776)*((0.0012581530032951
26)*(p))*(((0.808110045179776)/(0.17776788267203036))*(1))+((0.80811004517
9776)/(0.17776788267203036)))+(0.808110045179776)/(0.17776788267203036)))
)*(1))+(((0.808110045179776)/((0.001258153003295126)*(p)))*(1))*((1)+(1)))+(((
0.808110045179776)/(0.17776788267203036))*(1))+(((0.001258153003295126)*(p)
)*(1))*(((1)*(((1)/(0.17776788267203036))+1))/(((1)*(w))*(0.808110045179776))/((
0.808110045179776)/(0.17776788267203036))))+(0.17776788267203036))/((((((w)
(n))(w))+w))*((1)*(w)))*((0.808110045179776)/(0.17776788267203036)))/((0.80
8110045179776)/(0.17776788267203036))))+(1))/((((1)+((0.00125815300329512
6)*(p))*((0.808110045179776)*(((w)*(n))*((1)*(w)))*((1)*(w))))))+((((1)*(w))*
(0.001258153003295126)*(p)))*(((0.001258153003295126)*(p))*((0.001258153003
295126)*(p)))*(((w)*(w))*((1)*(w)))*(((w)*(n))*((1)*(w)))*(((0.00125815300329
5126)*(p))*((w)))*((1)*(w)))*((((0.808110045179776)/(0.17776788267203036))*
((1)*(w))*((1)*(w)))*((1)*(w)))*((0.808110045179776)/(0.17776788267203036)))/
((0.808110045179776)/(0.17776788267203036)))/((0.808110045179776)/(0.177767
88267203036)))/((0.808110045179776)/(0.17776788267203036)))))))+(1)*((0.001
258153003295126)*(p))))

Koşma:3

Ham uygunluk değeri:143.316865

((((((w)/(((w)*(0.20015712697066235))/(1))*(((1)+(1))/(n))/(1)))))-(-
0.004355549348683763))/(n))+((((1)/(((w))*((w))*((w)*(0.20015712697066235)))/
1))/(1)+(1)))/((w))*(((w))*((w))*((w)*(0.20015712697066235)))/(((1)+(1))/(n))))*(1
) + (((n)+((w)/(((w)*(0.20015712697066235))/(1))*((1)))))-(-
0.004355549348683763)))-
((1)+(n))+((((0.26919159538540605)+(((1)+(1))/(n))/(((w))*((w))*((w))*0.2001571

0.20767924990862574)))))))+(1)-((((1)/(-0.20767924990862574))/((1)*(-
0.41031045159805124)))+(w))/((1)*(-
0.896856712129366)))+(w))/(w)))+(1)+(0.10524131595401154)))/(n)/(n))-((1)/(-
0.20767924990862574))))))

Koşma:6

Ham uygunluk değeri:150.277945

(((((0.44691101715290504)/(-0.06119336144122922))-1))-((-
0.8072630464133168)/(w)*((0.8818527273283585)*(((1)/(1)-
(0.8818527273283585)))-((1)/(w)*(0.06840488084983298)))/(n))/((-
0.7437811688258225)/(((0.7583189773179213)-
(1)*((0.7942386203418665)+(w)*(1))))+(1))))))-((1)/(((w)*((-
0.44691101715290504)/(-0.06119336144122922))-
(((1)/(w)*(0.06840488084983298)))+(n))/(n)))*((w)*(0.3061318776133153))*((w
)*((w)))-1))*((0.06840488084983298))-((-0.8072630464133168)/(n))-1))

Koşma:7

Ham uygunluk değeri:130.942375

((((n)+(n))/((-0.8339668716462174)+((((n)/(w))/(n)+(n)))/((-
0.013839392463601885)+(w))*(((n)+(1))-(-
0.534435049413906)))+(n)+(n))+((n)+(n)+(1)+(n)))-(-
0.534435049413906)+(n))-
(((p)/(w)/(1)))/(w)/((n)/(w))/(n*(w)))/(w)/(1))+(-
0.18645166567827187)))/(w)/(p))-((-0.7902039698279482)+((-
0.18645166567827187)*(n)+(n)))))-((((((((((-
0.18645166567827187)*(n)+(n)))+(p))/(w)/(1)))/(w)/((w)/(1)+(1)))*((n))+((n)*
w))+((-0.7902039698279482)+((-
0.18645166567827187)*(((w)/(1)+(p))/(w)/(1)))/(n)+(n)))))/((n)*((1)-((-
0.8339668716462174)+((n)*(0.5567343121193042)))+((-
0.18645166567827187)*(n)+(n)))+(n)))+((-
0.18645166567827187)*(n)+(n)))+(n)+(n)))))+(w))+((w)/(1))*((n)+(n)+(n)))/

$$(w)+((-0.8339668716462174)+((n)+(n))*(((((((1)-(n)*(w)))+(p)))+(n))*((-0.18645166567827187)*(n)+(n))))+((-0.18645166567827187)*((n)*(w))*(w)))*((p)/(p))/(p)/(p)))/(p))$$

Koşma:8

Ham uygunluk değeri:147.964749

$$(((0.46267646712375643)+(((((((0.6109321087649706)/(0.17122760097087708))/((n)+(0.8955016871427659))-(((((((0.6109321087649706)/(0.17122760097087708))/(w))/(w))/(w))/(w))*((1))))*(1))/((((1)/(0.5932775240360189))*((1)/(0.5932775240360189)))/(w))/(w)-(0.17122760097087708)))/(w)/(w))+(((0.5932775240360189)/(0.17122760097087708)))+(0.17122760097087708)))+(((((((n)+(0.8955016871427659)))+(0.9165779351799646)))+(0.9165779351799646))/(((n)+(0.8955016871427659)))+(0.9165779351799646))-(((((((0.6109321087649706)/(0.17122760097087708))/(w))/(w))/(w))/(w))*((1))/(1))*((1)))$$

Koşma:9

Ham uygunluk değeri:136.383859

$$((0.7062811701122118)+(((1)/(w))/((((1)+(((1)*(((1)+(n))*((n)+(n)))+(1))-((p)+((0.7062811701122118)+((0.01915611374152637)+(n)))))/(p))+(((1)*((n)/(w)))+(n)-((0.7062811701122118)+(n)+(1)))))*(((1)+((-0.22646242902915104)/(-0.046855177286479255))+((1)/(0.15723723320161787)))+(1))))*(-0.851169119519726)))/((0.7062811701122118)+(((1)/(p))+1))/((n)+(n))+(((((((n))*((n))+((-0.36049548413627575)/(-0.012274659710132863))+((1)/(0.15723723320161787)))/((-0.22646242902915104)/(-0.046855177286479255)))/(((((((n)/(w))+n))*(-0.851169119519726))/((n)/(1)/(w)))+(n)-((0.7062811701122118)+(n)+(1)))))+(((1)+(n))*((n)+(n)))/((n))+((-0.22646242902915104)/(-0.046855177286479255))-(((((-0.22646242902915104)/(-0.046855177286479255))+((1)+((-0.22646242902915104)/(-$$

0.046855177286479255)))+((-0.22646242902915104)/(-
 0.046855177286479255)))+((((1)+((1)+((1)/(0.15723723320161787)))+((1)+((1)/(
 0.15723723320161787))*((n)))))+(0.8629914251449478)+(0.8629914251449478)))/(
 p))))

Koşma:10

Ham uygunluk değeri:134.338221

(((((1)/(0.22306447253705208))+((n)/(n)))/((1)/(0.22709246875117173))+((1))-
 ((1)/(0.22709246875117173))*((n)/((-0.02965930431599695)+(((1)-((n)/(-
 0.02965930431599695)+((n)))))*(((1)-
 0.1369659160901634)*((1)/((w)*(w)))/((0.5659290335913789)/(p)))/((((1)-
 0.02965930431599695)+((-0.02965930431599695)+((n)))/(((1)+(-
 0.14637970325834104))+((n)))+(-0.14637970325834104))*((1))*((w)/(n)))/((w)/(-
 0.053982524692072764))*((w)/(n))))+((n)))))*(((w)+(p))*((1)+(0.77859778878213
 74)))+((((1)-0.1369659160901634)*((1)/((0.5659290335913789)/(p)))/((-
 0.02965930431599695)+((n))*((n)/((-0.02965930431599695)+((-
 0.02965930431599695)+((n)))/((0.22709246875117173)))/((1)-(-
 0.24507802037758664)))+(((1)-
 0.1369659160901634)+((n))+((w)))/(((1))*((w)/(n)/(0.22709246875117173)))/(((w)/(n
))/(p)))))*((1)+((1)+((n)))))/(((n)-(p))-((w)/(1)))-((n)/(n)/((((1)-
 0.1369659160901634)*((1)/((0.5659290335913789)/(p)))/((w)/(0.906180377074601
 3))*((1)*(0.2586549172451045)))/((n))*((w)/(-
 0.053982524692072764))*((w)/(n)))/((0.22709246875117173))/((0.22709246875117
 173))/((n))*(((n)-((n)/(n)))+(0.22709246875117173))*(((w)*(w))/(-
 0.053982524692072764))*((1)*(0.2586549172451045)))*((w)/(n))))+((n)))/((0.2270
 9246875117173))))))

Koşma:11

Ham uygunluk değeri:131.276132

(((((1)/(0.22306447253705208))+((n)/(n)))/((1)/(0.22709246875117173))+((1))-
 ((1)/(0.22709246875117173))*((n)/((-0.02965930431599695)+(((1)-((n)/(-
 0.02965930431599695)+((n)))))*(((1)-
 0.1369659160901634)*((1)/((w)*(w)))/((0.5659290335913789)/(p)))/((((1)-
 0.02965930431599695)+((-0.02965930431599695)+((n)))/(((1)+(-
 0.14637970325834104))+((n)))+(-0.14637970325834104))*((1))*((w)/(n)))/((w)/(-
 0.053982524692072764))*((w)/(n))))+((n)))))*(((w)+(p))*((1)+(0.77859778878213
 74)))+((((1)-0.1369659160901634)*((1)/((0.5659290335913789)/(p)))/((-
 0.02965930431599695)+((n))*((n)/((-0.02965930431599695)+((-
 0.02965930431599695)+((n)))/((0.22709246875117173)))/((1)-(-
 0.24507802037758664)))+(((1)-
 0.1369659160901634)+((n))+((w)))/(((1))*((w)/(n)/(0.22709246875117173)))/(((w)/(n
))/(p)))))*((1)+((1)+((n)))))/(((n)-(p))-((w)/(1)))-((n)/(n)/((((1)-
 0.1369659160901634)*((1)/((0.5659290335913789)/(p)))/((w)/(0.906180377074601
 3))*((1)*(0.2586549172451045)))/((n))*((w)/(-
 0.053982524692072764))*((w)/(n)))/((0.22709246875117173))/((0.22709246875117
 173))/((n))*(((n)-((n)/(n)))+(0.22709246875117173))*(((w)*(w))/(-
 0.053982524692072764))*((1)*(0.2586549172451045)))*((w)/(n))))+((n)))/((0.2270
 9246875117173))))))

(1))/(((w)*((w)*((w)*(n))))-
 (1))))+(w)/(w))+((1)/(0.33537192332342824)))/(1))/((p)+((n)+((((0.90432925238
 47564)+(((1)/(0.33537192332342824)))/(w)))/(0.33537192332342824)))/(((w)*(n))-
 (1)))+(-
 0.10039832866769205)))/(0.33537192332342824))+((1)+(((((((1)/(0.335371923323
 42824)))/(0.33537192332342824)))/(w)))/(((w)*(n))-
 (1)))+(n)+(1))*((1)+(1)))))))/(1))*((p)+(1)))+(1)

Koşma:12

Ham uygunluk değeri:113.803435

((-
 0.12903094716790642)+((((((((((0.9144674790624843)/(0.15697968711935895))*
 (p)/(p)))*((p)/(p))-
 (((0.9144674790624843)/(0.15697968711935895))/(p)))))*((p)/(p))-((n)/(p))))-
 (0.021825033715844544))-(0.021825033715844544))-
 ((p)/(((0.9144674790624843)-
 (w))*((n)*(((0.9144674790624843)/(0.15697968711935895))*((0.933300019210809
 9))-
 (((0.9144674790624843)/(0.15697968711935895))/(p))))))*(((0.9144674790624843
 /0.15697968711935895))*(((0.9144674790624843)/(0.15697968711935895))*((0.93
 33000192108099))))))*((p)/(p))-((((p)/(p))/(p))/(p)))-
 ((p)/(((0.15697968711935895)-
 (w))*((n)*((((0.9144674790624843)/(0.15697968711935895))-
 (0.021825033715844544))*((((0.9333000192108099)/(0.15697968711935895))*(((
 n)/(p))+((w))*((0.9333000192108099))-((((p)/(p))-((p)/(p)))/(p)))))-((p)/(p))))-
 (((p)/(p))-((0.9144674790624843)-(w))))))*((((((p)/(p))-((n)/(p)))-((n)/(p)))-
 ((((((0.9333000192108099)/(0.15697968711935895))*((p)/(p)))/(0.15697968711935
 895))*(((0.9144674790624843)/(0.15697968711935895))*((0.9333000192108099)))/(
 p)))-((n)/(((p)/(p))-(w))))-
 (w))*((0.9144674790624843)/(0.15697968711935895)))))))-
 (0.021825033715844544)))

Koşma:13

Ham uygunluk değeri:126.356561

$$\begin{aligned} & (((p)/((0.22116106246652123)*(p)))+(((n)+(((w)+(-0.8657996355862355))- \\ & (1))/(w))/(w)))/((p)/((- \\ & 0.7084276469929273)/(w))/(0.23598132246918113)))+((((p)+(n))*((w)+(- \\ & 0.8657996355862355))*((w)+(n)+(1)))+(((p)/(((w)+(- \\ & 0.8657996355862355))*((p)+(1)))/(1))/(0.23598132246918113))+((0.221161062466 \\ & 52123)*((0.22116106246652123)*(p))))+(1)))+((((((0.22116106246652123)*(p))/ \\ & ((-0.7084276469929273)/(w))/(w))/(p))+((((p)/(((p)+(n))*((w)+(- \\ & 0.8657996355862355))*((w)))- \\ & ((n)+(w)))+(((0.22116106246652123)*((0.22116106246652123)*(p)))*(0.908364167 \\ & 3942862)))/(n))*(((0.22116106246652123)*(p))/(((p)+((((1)/(0.235981322469181 \\ & 13)))+(1)/(0.9034985377935221)))+(0.031378322761216326))/(1))+((n)- \\ & ((p)/(((1)+(n))+(((0.22116106246652123)*(p))*((w)+(- \\ & 0.8657996355862355))))*(((1)/(0.23598132246918113)))+((- \\ & 0.7084276469929273)/(w))/(0.23598132246918113)))+(- \\ & 0.36995136708484555))/((0.3858971490459038)/(n))))*((w)+(- \\ & 0.8657996355862355))*((w)))) \end{aligned}$$

Koşma:14

Ham uygunluk değeri:127.870315

$$\begin{aligned} & ((((((((((((-0.7303488867964358)/(- \\ & 0.13216090068787378))/(0.9786203903977855))-((-0.7303488867964358)/(- \\ & 0.13216090068787378))/(p)/(n))))/(0.9786203903977855))- \\ & (((n)+(0.7160435112733596))/(n)+(0.9786203903977855)))/(p)/(n)))/(0.97862039 \\ & 03977855))-((n)/(p))*((-0.7303488867964358)/(- \\ & 0.13216090068787378))/(0.9786203903977855)))-(((((-0.7303488867964358)/(- \\ & 0.13216090068787378))-((n)/(p))*((n))/(p)/(n)))-((((((((n)/(p))/((1)-(w)))/((- \\ & 0.7303488867964358)/(- \\ & 0.13216090068787378)))+(0.7160435112733596))/(0.9786203903977855)))/((1)- \\ & (w))*((1))*(((((-0.7303488867964358)/(- \end{aligned}$$

0.13216090068787378)))/(0.9786203903977855))/(0.9786203903977855)))+(0.71604
 35112733596))-(((1)/((1)-(w)))*(1)))/(0.9786203903977855)))/(-
 0.32564770305792223))/(p))*((((((-0.7303488867964358)/(-
 0.13216090068787378)))-
 ((n)+(0.9786203903977855))/(p)/(n)))/(0.9786203903977855))-(((
 0.7303488867964358)/(-0.13216090068787378))/(p))*(((0.7303488867964358)/(-
 0.13216090068787378)))+(0.679337384961237))/(0.9786203903977855)))-
 ((n)+(0.7160435112733596))/((-0.7303488867964358)/(-0.13216090068787378))-
 ((n)/(p))))-(((w)/(n))-((-0.7303488867964358)/(-0.13216090068787378))*((1)-
 (w)))/(0.9786203903977855)))-
 ((w)/(n)/(n)))/(n)/(p)/(n)))/(0.9786203903977855))-((((((n)+((-
 0.7303488867964358)/(-0.13216090068787378))-
 (0.9786203903977855))/((((((p)/(0.7160435112733596))/(p)/(n)))/((((n)/(0.97862
 03903977855))/(p)/(n))/(0.9786203903977855))/(1)-(w))))*(1))+((p)/(n))*((-
 0.7303488867964358)/(-0.13216090068787378)))+((((n)/(p)))/(-
 0.13216090068787378))+(-0.7303488867964358)/(0.9786203903977855)))/((1)-
 (w))*((1)/(-0.32564770305792223))/(-0.32564770305792223))/(p)/((((((-
 0.13216090068787378)/(n)/(p)))/(w)/(1)))+(1))*((-0.7303488867964358)/(-
 0.13216090068787378))-((n)/(p))*((-0.7303488867964358)/(-
 0.13216090068787378))))-(((((-0.7303488867964358)/(-
 0.13216090068787378))/(p))*((((n)/(p))/(1)-(w))*((p)/(n)))/(-
 0.32564770305792223))-((((((-0.7303488867964358)/(-0.13216090068787378))+((-
 0.7303488867964358)/(-
 0.13216090068787378)))/(0.9786203903977855))*(((n)+(0.7160435112733596))/(0
 .9786203903977855))*((w))/(p)/(n))))-((-0.7303488867964358)/(p)/(n)))-
 (((0.9786203903977855)/(1)-(-0.7303488867964358))*(((n)/(p))*((1)-(w))/(p)/((-
 0.7303488867964358)/(-0.13216090068787378))))/(0.9786203903977855))))))

Koşma:15

Ham uygunluk değeri:121.718039

(((1)+((((1)+(1)+(1)))+((((0.2206046288928969)/(1)/((w)*((w)*(n)))))+((((1)+((
 (1)+(1))/(w)))+(((1)/(w))+1)))/(0.13437090773804616))/((w)*((w)*(w)*(n))*((w))

$$\begin{aligned} &)) + (((w) * ((w) * (n))) / ((1) + (1))) / (((1) / (w)) - (- \\ &0.9110133423986907)) * ((w) * (1))) / (w)) - (-0.9110133423986907)) * (((((p) - \\ &(0.8551941932436051)) - ((p) + (1))) - \\ &(((((((1) + (1)) + (1)) + ((n) / (0.13437090773804616)) / ((w) * ((w) * ((w) * (n)))))) - ((- \\ &0.705645230461678) / (0.6909271113997917))) + (1)) + ((1) + ((1) + (((1) + (1)) + ((1) + (1))) \\ &- (-0.9110133423986907)))) + (1)) + ((1) + (1)) / (((w) * (n)) + ((p) - \\ &((((((1) + (1)) / (w)) / (0.13437090773804616)) / (0.13437090773804616)) / ((w) * ((w) * ((w) \\ &* ((w) * ((w) * (n))))))))) + (((1) + (1)) + ((1) + (1) + (((0.3548185356682252) + (((0.134 \\ &37090773804616) + (1)) + ((n) / (0.13437090773804616)) / ((w) * ((w) * ((w) * (n)))))) / (0.13 \\ &437090773804616)) / ((w) * ((w) * ((w) * ((w) * ((w) * (n)))))) / ((w) * (n)) / (w)))) + (1 \\ &))) \end{aligned}$$

Koşma:16

Ham uygunluk değeri:147.796132

$$\begin{aligned} &((((((1) / (w)) / (((n) - (((n) - \\ &(1)) * (0.8410806440939571)) * ((1) / (w))) + (((((1) / (w)) * ((1) / (w))) * (0.84108064409395 \\ &71)) / ((1) * (0.8410806440939571)))) * (0.8410806440939571)) * (0.841080644093957 \\ &1)) * (((((n) - (1)) * (0.8410806440939571)) / (n)) * (0.8410806440939571)) - (- \\ &0.22298186888125815)) * (0.8410806440939571))) + ((((((1) / ((1) - (- \\ &0.3384386148948402)) * (0.8410806440939571))) + ((1) - (- \\ &0.3384386148948402))) + (0.8512005144037309)) - ((w) - (w))) - \\ &((0.3335447932284068) * (1)) + (0.8512005144037309)) - (((((1) - (- \\ &0.3384386148948402)) / (w)) + (((1) / (w)) * (((1) / (0.8410806440939571)) / (w)))) / ((n) - \\ &((1) * (0.8410806440939571)) * (0.8410806440939571)))) + (0.8512005144037309)) - \\ &(-0.22298186888125815)) + ((1) / (((n) - (((n) - ((1) + (0.5681147387102781)) / (w)))) - \\ &(1)) * (0.8410806440939571)) * (0.8410806440939571)) * (0.8410806440939571)) * (0. \\ &8410806440939571)))) \end{aligned}$$

Koşma:17

Ham uygunluk değeri:145.287513

$$\begin{aligned} & (((1)/((1)+(n))+((((1)+(p)/(p)))/(1))/((p)/(p))+((0.23093964030544267)+(- \\ & 0.17003831880634568))-(n))) * (-0.358416195660092)) + ((((- \\ & 0.5499318742891457)-(n))/(n)+(-0.17003831880634568))/(p))+n)-(- \\ & 0.30326397591422494)) / (((w)+(w))+((((0.23093964030544267)*(w))*(w))*(w)) \\ & +(-0.17003831880634568))+(-0.17003831880634568))-((1)+((w)+(w))))- \\ & ((p)/(p)))) + (((0.23093964030544267)/(((n)/(n))*(-0.7779772606575756))- \\ & (((p)/(p))+(-0.9845856856482968))-((1)*(w))+(- \\ & 0.17003831880634568)))) / (((0.23093964030544267)-(n)- \\ & (1)))/(0.23093964030544267)/((((0.23093964030544267)*(w))+(- \\ & 0.17003831880634568))*(1))+((0.23093964030544267)*((n)/(n)))) + (((- \\ & 0.5499318742891457)-(1))*((0.23093964030544267)/((-0.5499318742891457)- \\ & (n))+((1)/(0.23093964030544267)))) * (- \\ & 0.358416195660092)) + ((1)/(0.23093964030544267)) + (1)) / (1) \end{aligned}$$

Koşma:18

Ham uygunluk değeri:117.055284

$$\begin{aligned} & (((((((0.7491290279427212)/(0.14104619023438825))- \\ & (((0.14104619023438825)- \\ & (p))/((0.7491290279427212)/(0.14104619023438825))))- \\ & (p))/((0.7491290279427212)/(0.14104619023438825)))/(p))- \\ & (((((((((p)/(w))*((0.7491290279427212)/(0.14104619023438825)))/((w)*(p)))/(n))* \\ & (((0.7491290279427212)/(0.14104619023438825)))/(0.14104619023438825))- \\ & ((p)/(w)))/((0.1751894583810072)/(w))/(w))/(p)) + (((((p)/(w))*((0.7491290279427 \\ & 212)/(0.14104619023438825)))/((p)/(1)))/(n))/((((p)/(w))*((0.7491290279427212)/(0 \\ & .14104619023438825)))/((p)/(1)))/((1)/(1))))- \\ & (((p)/(w))/((w)*(w)))/((p)/(1))) + ((w)/(p))/((p)/(1))) - \\ & (((0.7491290279427212)/(0.14104619023438825))/(0.1751894583810072))/((p)/(1 \\ &)) - \\ & (((((((((p)/(w))*((1)/(0.1751894583810072)))/(((1)*(n)))/(0.1751894583810072)))/(0 \\ & .1751894583810072)))- \\ & (((((((p)/(w))/(w))*((1)*(n)))/(((n)*(n)))/(0.1751894583810072))- \\ & ((p)/(w)))/(((1)*(n)))/((p)/(w)))/(n))/((p))/((0.1751894583810072)*((w)/(1)))/((\end{aligned}$$

$$\frac{(p)/(((0.5648511897608877)+(w)-(1))-(p))/(0.7491290279427212)))-}{\frac{((((((((p)/(w))^*(1)/(0.1751894583810072)))/(p)/(1)))/(n))^*((0.14104619023438825)-(p))-(p)/(w)))/(0.1751894583810072))-}{(p)/((1)*(n))/(0.1751894583810072)))/(1)*(w))/((((1)+((0.7491290279427212)/(0.14104619023438825)))/(0.1751894583810072)))/(0.1751894583810072))-}{(p)/(p)))/(p)/(p))}$$

Koşma:19

Ham uygunluk değeri:138.986687

$$\frac{(((1)+((((((n)*(-0.2630278101484421))+((((((n)*(-0.2630278101484421))+1))+(((n)*(-0.2630278101484421))+1))+((w))+(((n)*(-0.2630278101484421))+1))+(((n)*(-0.2630278101484421))+1))+(((0.6622800937212445)+1)/(w)))))/(w))/(w))/(w))+}{\frac{(((((((n)*(-0.2630278101484421))+1)/(w))+((w))+((((((n)*(-0.2630278101484421))+1))+((w))/(w)))/(w))/(w))/(w))/(w))+}{\frac{((((((((((((((n)*(-0.2630278101484421))+1)/(w)))/(w))+(((1)/(0.3585817363851618)))/(0.3585817363851618))/(n)))/(w))/(w))/(w))/(w))/(w))/(w))+}{\frac{(((0.6622800937212445)+1)/(w))-}{\frac{((n)+(w))+(n))/(w))/(w))+((w))+((w))/(w))+}{\frac{(((n)/(-0.7529980948907314))+(((1)/(0.3585817363851618)))/(0.3585817363851618))/(0.3585817363851618))*(-0.2630278101484421))+((0.6648292521316694)*(p))/(p))^*((((((((n)*(-0.2630278101484421))+((0.6951521307673083)/(w)-(0.10604043170159705))))+(w))/(w))/(w))+((w))/(w)-(0.10604043170159705)))/(0.3585817363851618))+1)))))}$$

Koşma:20

Ham uygunluk değeri:131.948723

$$\frac{(((p)-}{\frac{(((((((n)/(n))/((0.0703468001775196)/(1)))+(((n)*(w))/(w)))/(w))/(w))+n)))/(p)+}{\frac{((((1)+1)/(w))/(w))/(w)))*((0.0703468001775196)/((0.13336209539946275)*((0.3062121343362201)/(1))+(((1)*(((1)/((0.0703468001775196)/(1)))+((0.30621213433$$

62201)/(1))/(((0.3062121343362201)/(1))/((0.0703468001775196)/(1))))/(w))/(w))/(w))+(((1)+(1))/(w))/(w)+((((0.3062121343362201)/(1))/((1)/(w))/(w)))/((0.0703468001775196)+(-0.9346041865342316)))-
 (((((1)/(w))/(w))/(w))/(w))/(1)/(n)))))))+((((((((((1)/(n))+1))+1))/(w))/(w))/(w))/(n))/((0.0703468001775196)/((((1)+(1))/(w))/(1)/(w))/(n)/(w))))/(w))/(w))/(w))+(((1)+(1))/(w))+((0.3062121343362201)/(1))/((0.0703468001775196)/(1))))))

Koşma:21

Ham uygunluk değeri:141.015619

((((((((((1)/(w))+(((1)+(1))/(w))+(-0.6636199986858387)))/(w))+((((1)/(w))/(w))/(w))+(-0.6636199986858387)))/(w))+(((1)/(w))+(-0.6636199986858387)))-
 (1))/(w))+1))/(w))+0.8756838039847481))/(n)-
 (1))/(w))/(w))/(w))+((0.9299842789443136)+((1)+((((1)+(((1)/(1)/(w))+(-0.6636199986858387)))/(w))+(((n)-
 (1))+1))/(w))+((1)+(n))))+((0.9299842789443136)+(n)))/(n)+0.8756838039847481)))-
 (((((1)+(((n)+0.8756838039847481))/(w))+(((1)+(n))+((0.9299842789443136)+(n))/(1))/(w)))/(0.8756838039847481))/(w))+(-0.6636199986858387)))+((1)+((0.9299842789443136)+((0.9299842789443136)+(1)/(w)))+(((1)+(n)+0.8756838039847481)))+((0.8756838039847481)+(n)))+((1)+(1)+(n))))/(w)))/((0.39206433547291186)*(p)))))))+(1)

Koşma:22

Ham uygunluk değeri:138.989979

((((((((((1)*(0.8126151244214805))+1))*((1))+(-0.08882984104046121))+((n)+(n))/(n)+(n))))+(0.7439515505656373))*((1))+(-0.17045894226546354)))+(1))+((1)+(-0.17045894226546354)))+(1)/((n)-(p))*((1)+(((1)+(-0.08882984104046121))*0.8126151244214805))+1))-
 (((0.08998852087649922)+1))*((1)+(1)))))))+((1)+((-0.17045894226546354)+(p)))/(1))*(((1))+(-

0.17045894226546354))+(0.08998852087649922)+(1)))+(n)+(n))+(p))) -
 (((((n)+(n))*((1)+(1)))+(1)+(1)+(-
 0.08882984104046121))))/(p)/(((1)+((1)+(1))/(0.08998852087649922)+(1))))+(1
)+((1)*(0.8126151244214805))))+(((1)+(-
 0.08882984104046121))*((1)+(1)))))))/((((0.7439515505656373)+(n))+((w)-
 ((1)+(-0.17045894226546354))+(0.08998852087649922)+(1))))+(1))-(((1)+(-
 0.08882984104046121))*((n))-((w)/(1))))-(((1)+(-
 0.08882984104046121))*((1)+(1))-((-0.08882984104046121)/(1)))))))+(((1)*((-
 0.17045894226546354)+(p)))/(n)*(p)))**((1)+(-0.08882984104046121))*(((1)+(-
 0.08882984104046121))+(-0.08882984104046121)))+(1)

Koşma:23

Ham uygunluk değeri:134.865735

(((((((((((((1)+(1))*(((1)+(n))/(w))/(w))+1)))-((1)/(-
 0.8188980314037287)))+(1))/(w))+((((1)+(1)+(1))/(w))+(((1)+(1))/(w))+0.2334
 6256522157338))/(w))*(((1)+((1)+(1))*((1)+(1)+(1)))/(w))+((w)))-
 ((1)/(w))*(p))/((w)+(0.9694904228530314))*(p)+(0.34083972095550963))))*(((
 (((((1)+(1))/(w))/(w))/(w))+(((1)+((1)+(1)))/(w))/(w))*((1)+(1))/(n))/(w))/(1)*(p)
)))*(((0.703088566522621)-(w))*((p)/(0.07431040381747778))))/((((-
 0.6158804519175927)+(-0.014624609153077284))-((-
 0.5843397451491745)/(0.9517708569540506)))*(((n)-(n))+((-
 0.7403855853436447)/(p))))*(((p)-
 ((1)/(w)))+(1)+((((1)+(1))+(((1)+(1))/(w))*((n)+(1))))-((1)/(-
 0.8188980314037287))*((n)*((1)+(1)))))*((w)*(p)-(((p)/(w))+((1)*(-
 0.7119051570593868)))+(w)/(-0.5626977801149236))/((((n)-
 (w))/(w))/(w))/(w))+((p)))))))/((((w))*(((1)+(1))+((1)/(n)))/(w)))+(1))+(((((-
 0.1918544523380019)*(-0.14657222719237772))*((p)+(n)))+(((((-
 0.1918544523380019)*(-
 0.14657222719237772))/(w))/(w))/(w))*((((1)+(1))/(w))/(w))/(w))+0.23346
 256522157338))/(n)+(1))+(((p)*(0.6346596300763356))+((1)+((((((((1)+(1))/(w)
))+(1))/(w))/(w))/(w))+1))*((n)+((((w)/((0.23346256522157338)/(n)))+(1))/(w))+1
)))+(1))))-

$$\begin{aligned} & (1)))/((((1)+(1))*((((1)+(((1)+(1))/(w))/(w)))/(w))+1)/(w))+1)))+(0.7540497583 \\ & 123156))/(((n)+(p))*((w)*(w)))- \\ & (((p)/(w)/(w))*((1)/(w)))))))+(((0.8323340675944155)*(1))+((1)+(((1)+(1))*(((w)/ \\ & (((0.23346256522157338)+((1)+(1)))+(1))/(w))+1)))+(1))/(w))+1)))))) \end{aligned}$$

Koşma:24

Ham uygunluk değeri:130.286965

$$\begin{aligned} & ((((((((((p)/((n)/(0.14243042149694185))+p)))/(w))/(w))/(w))*((1)+(((n)/(w))/(w) \\ &))/(0.14243042149694185)*n)))+(1))/((0.14243042149694185)*n))/(w)/(n)/(w) \\ &))/(w))+((0.6235018212457661)+(1)+(((p)/((((0.6235018212457661)+n))+((n) \\ & +n))+1)+(n)/(n))))-(- \\ & 0.7855376898243733))+(((p)/(p))*((n)/(w))/(w))+((n)/(w))+(- \\ & 0.7855376898243733))+1))/((0.14243042149694185)*n))+n)+p))/(0.295568 \\ & 5189438837))+1)))) \end{aligned}$$

Koşma:25

Ham uygunluk değeri:149.502160

$$\begin{aligned} & ((1)+((0.8688537906244647))+((0.8688537906244647))+((0.8718828698023608))+(((\\ & 0.8718828698023608))+(((n)+((1)-((0.8688537906244647)/((- \\ & 0.371719937478993)*w)))/((- \\ & 0.371719937478993)*w)))/(((0.8688537906244647)+n)-((w)+n)))))+(n)+((n)- \\ & (1)))/(n))/((((n)+1)-((w)+n))/(((n)+1)+(-0.807662478558562))+1))- \\ & ((w)+n)))-((n)*(-0.7032001398984343)))+(1)))) \end{aligned}$$

Koşma:26

Ham uygunluk değeri:135.924495

$$\begin{aligned} & (((((1)/(0.1969663860262214))*((1)+((((p)/(w))/(w))-(- \\ & 0.29722015247550804)))/(0.1969663860262214)+(((n)*(w))*w)/(0.19696638602 \\ & 62214))))+((p)/(w))/(((n)*(w))*n))/((1)+((w)/(0.004511721899971288))/((((1)+(\\ & (1)/(w))/(w))/((-0.7902367784596219)/((n)+1))+(- \end{aligned}$$

0.8732132175346898)*((n)*(w))))+((1)/(w)))-(-0.7703245588440097)))))))-(-
 0.7703245588440097))-((((1)/(0.1969663860262214))+n))-(-
 0.7703245588440097))-((-
 0.8732132175346898)*((1)/(0.1969663860262214))))*(n)/(p))))

Koşma:27

Ham uygunluk değeri:139.482775

(((((((((1)+((1)/((-0.7537926750973765)+w)))-(-
 0.8384848599501349)))/(w))/(w))/(w))/(w))/(n)/(((((((1)+((-
 0.3304941967737368)/(-0.08763422774506469)))/(w))/(w))/(w))+1))+((w)/(w))-
 (n)))/(w))/(w)))-(-0.7429873951445275))/(w))+(((((((1)+((w)+(1)/((-
 0.7537926750973765)+w)))/(w)))-(-
 0.8384848599501349)))/(w))/(w))/(w))/(w))+((w)-(n))-((p)+(p))-
 (1))+((n)+(p))*((((1)/((-0.7537926750973765)+1)))-(-
 0.8384848599501349))+((1)/(1)))/((-0.3304941967737368)/(-
 0.08763422774506469))*((-0.8911098837345419)/(-0.5250432079308868))/(-
 0.30061995205498293))+((n)+(w))-((n)+(p))))))

Koşma:28

Ham uygunluk değeri:122.421472

(((((1)/(0.17763903326244979))-(((((-
 0.2705852981985478)/(w)/(1)))/(w)*(0.9873699257091479)))/(w)*(1)))/(w)*((n)/
 (1))-(((((-0.2705852981985478)/((-
 0.2705852981985478)/(w)/(1))+w)))/(w)*(1)))/(n)/(p)))))))/((-
 0.45144785449441893)+w)))-((n)/(p))/(0.17763903326244979))-(((((-
 0.2705852981985478)/(w)/(1)))/(w)*(w)))/((-
 0.45144785449441893)+w)))/(w)*((((p)-((p)-((1)/(0.17763903326244979))))-
 ((n)/(p))/(1)/((1)/(0.17763903326244979))-((1)/(n)/(1)))))))-
 (((((((0.0726551409217786)*((n)*(1)))/((-
 0.45144785449441893)+w)))/(w)*(1))/(1))/(1))/(w)*((n)/(p))/(1)/(p)-((p)-
 (w)*((1)/(0.17763903326244979))*1)))))))-((-0.2705852981985478)/(n)))-

$$\begin{aligned} & ((w)/((((p)-(p)-((1)/(0.17763903326244979))))-(((-0.2705852981985478)/((- \\ & 0.45144785449441893)+(w))*(((-0.22601148636360258)/(1))/(p))*(- \\ & 0.4235534106490917)))/((n)/(p))/(0.17763903326244979))))/(p)/(1))- \\ & (((n)/(p))/((((-0.2705852981985478)/(w)*(w))/(w)*(1))/(1)/(n)))-((1)))))) \end{aligned}$$

Koşma:29

Ham uygunluk değeri:141.699038

$$\begin{aligned} & (((p)+((-0.2135071042988017)*((n)+(((n)+(w))*((1)/(((w)- \\ & (0.5962755771336499))+((-0.8777102566686041)-(- \\ & 0.6160788431838522))*((w))*((n)))+(n)))+(n)+((n)/((- \\ & 0.8516858042458472)+(w))*((n))))+(((((((1)+(n))/(- \\ & 0.2135071042988017)*(1)))+(n))+((w)/(-0.6543816931798969)))+((((w)/(- \\ & 0.6543816931798969))+(-0.27954511683413064))+(((w)/(- \\ & 0.6543816931798969))+((w)/(-0.6543816931798969)))+((- \\ & 0.0771084758812135)+(((w)/(-0.6543816931798969))/((- \\ & 0.8516858042458472)+(w)))/((w)-(-0.5962755771336499)))+((- \\ & 0.8777102566686041)-(-0.6160788431838522)))))))+(((w)/(- \\ & 0.6543816931798969))+p))+(((w)/(-0.6543816931798969))+((w)/(- \\ & 0.6543816931798969)))+((w)/(-0.6543816931798969))))/(w))/((n)*(w))+((w)/((- \\ & 0.8516858042458472)+(w))/(-0.6543816931798969))-(((-0.8777102566686041)-(- \\ & 0.6160788431838522))+((((((n)+1))/(-0.8516858042458472)+(w)))/(- \\ & 0.6543816931798969))+((n)+1))/(-0.27596643160840795)))+(((w)/(- \\ & 0.6543816931798969))+(-0.6543816931798969))/((- \\ & 0.806413660667098)+(w)))/((w)-(-0.5962755771336499)))/((- \\ & 0.8777102566686041)-(- \\ & 0.6160788431838522)))))))/((n)*(p))+((0.9187012752139481)+(((1)-((((w)/(- \\ & 0.6543816931798969))+(-0.6543816931798969))/((n))*((w)- \\ & (0.5962755771336499))+((-0.8777102566686041)-(-0.6160788431838522)))))))+(- \\ & 0.6543816931798969))/((-0.8516858042458472)+(w)))+((((- \\ & 0.2135071042988017)*((n)*(w))+1)))+((0.14930308477454957)+(1)))+(1))+(((n) \end{aligned}$$

$$\frac{*((-0.806413660667098)+(w)))/(-0.6543816931798969))+(-0.6543816931798969)))/((n)*(w)))+(1)+(1)))$$

EK 2.2

Koşma:0

Ham uygunluk değeri:197.718659

$$\frac{((((1)/(0.23963998315839086))+1))-(((0.23963998315839086)-((1)/(w))/(w))*(((1)/(w))/(w))/(0.23963998315839086)))+((((1)/(w))/(w))/(w))-((1)/(w))-0.6201662018989056))$$

Koşma:1

Ham uygunluk değeri:146.924843

$$\frac{((((1)+(1)+(1)))/((-0.764379983658148)+(w))*((n)*(w)))/((w)*((0.7418099091117316)*(w)))/((w)*((w)*((-0.764379983658148)+(w))*((n)*(w))))+((((1)-0.7094933808359752)/(0.7418099091117316))/(((p)+(n)+(n))/(p)+(p)))+(1))$$

Koşma:2

Ham uygunluk değeri:151.204783

$$\frac{((((1)/(n)+(n))/(n))+(((w)/((1)/(w))+1))-((w)/(-0.26687282475962903)))/((w)-((1)/(n))/((w)-0.20234589427818817)))/((w)-0.20234589427818817))+((w)-(1))))$$

Koşma:3

Ham uygunluk değeri:143.020191

$$\frac{(((((-0.997477669267672)/(-0.2822942637290313))-((-0.9731425326192484)+(-0.12380619771955825)))+(1))+(((n)/(p))*((-0.997477669267672)+(-0.12380619771955825)))*((-0.997477669267672)/(-0.2822942637290313))+((-0.997477669267672)/(-0.2822942637290313)))+(1)-0.6982503941740144)))+(((((-0.997477669267672)/(-0.2822942637290313))/((1)-(-0.6982503941740144)))/(n))/(w))*((((-0.997477669267672)/(-0.2822942637290313))+1))/(w))/(((1)*(n)*(w)))+(1)-0.857271901265379)*(n))/(w))))$$

Koşma:4

Ham uygunluk değeri:227.645277

$$\begin{aligned} & (((n)/((-0.49320539808515707)/(0.5666549185135656)))/((- \\ & 0.8837036061490438)/(n)))/((0.5619596138419396)+(-0.34490202523064895))- \\ & ((n)/((n)*(-0.49320539808515707)/(0.1036811848653858))-(p))/(((\\ & 0.49320539808515707)/(0.5666549185135656)))/((- \\ & 0.8837036061490438)/(n))/((1)*((p)/(1)))) \end{aligned}$$

Koşma:5

Ham uygunluk değeri:170.102749

$$\begin{aligned} & (((1)+(((1)*(1))*(1))+((1)/(0.32629252752582194))))+(((1)/((0.202396677435560 \\ & 56)*(w)))/(n))/(((0.20239667743556056)*(w))*(w)))- \\ & (((w)/((0.20239667743556056)*(w)))/(w))/(p))*(((1)/(0.32629252752582194))/((0. \\ & 20239667743556056)*(w)))/((0.20239667743556056)*(w)))) \end{aligned}$$

Koşma:6

Ham uygunluk değeri:138.431762

$$\begin{aligned} & ((((((0.23367468138862146)/(0.035000474674161675)))+(0.9295335225432801))- \\ & ((0.23367468138862146)+((1)+(n)))))/((- \\ & 0.3019073863056988)/(0.4608421993725198))-(((\\ & 0.8001141230576272)+(w))/(0.035000474674161675))- \\ & ((0.23367468138862146)/(0.035000474674161675))))+(((1)/((p)+((0.23367468138 \\ & 862146)/(0.035000474674161675)))))*((p)- \\ & (n))*(1))*(((0.23367468138862146)/(0.035000474674161675))- \\ & ((n)/(((1)+(n))+0.5621625061902042)))) \end{aligned}$$

Koşma:7

Ham uygunluk değeri:181.865971

$$\begin{aligned} & ((((((n)/(w))/(w))/(0.21962596719135807))- \\ & (((n)/(w))/(w))/(1))/(((0.21962596719135807)+(n))/((1)/((n)/(w))+((1)/(- \\ & 0.14686407015978553)))))-(((1)*(-0.24989867920516928))/(1))/((p)-(p))/((1)- \\ & (n)))/(((n)/(w))-((w)+(n)))-((-0.8030078372932076)/(w))))- \\ & ((1)+((1)/(0.21962596719135807)))) \end{aligned}$$

Koşma:8

Ham uygunluk değeri:219.866905

$((((1)+(0.42581147766942684))/((0.7783434324796978)/((1)+(0.42581147766942684)))-$

$((1)+(1))))+((w)/((0.29919579406231445)*(1))))+((((1)/(0.7783434324796978))/((n)+(1)+(1))))+(((1)/(0.7783434324796978))/((0.29919579406231445)*(1)))+(1))-$
 $((((0.29919579406231445)*(1))*((w)/((w)-(p))))*((1)/(0.7783434324796978))-$
 $((1)+(1))-(((0.6775503520283617)*(n))/((1)/(0.7783434324796978))))))$

Koşma:9

Ham uygunluk değeri:146.579961

$(((((0.1057011853464418)-$

$((n)*(n))/((p)+(1)))+(((0.3997857716864841)/(0.1057011853464418))+(1))+(-$
 $0.20213083513180297))+((((0.3997857716864841)/(0.1057011853464418)))+(0.3$
 $997857716864841)/(0.1057011853464418)))/((n)*(n))/((0.1057011853464418)+(($
 $w)-(1))*((n)/(0.1057011853464418)))))+(1)*(1))))$

Koşma:10

Ham uygunluk değeri:154.494029

$(((((((-0.6219303047386605)/(-0.1356125255746825))+1))+(-$
 $0.6219303047386605)/(-0.1356125255746825)))-((($
 $0.40321593703851843)/((w)*(w)))*((-0.6219303047386605)/(-$
 $0.1356125255746825)))/(((0.40321593703851843)/(w))*((1)+(-$
 $0.6219303047386605)/(-$
 $0.1356125255746825))))+(((w)*(w))*(((w)*(w))*((1)+(n))))+((-$
 $0.40321593703851843)+(((0.6219303047386605)/(-$
 $0.1356125255746825))/((1)+(n))*((-0.6219303047386605)/(-$
 $0.1356125255746825)))+((1)+((-0.6219303047386605)/(-0.1356125255746825))))))$

Koşma:11

Ham uygunluk değeri:154.849540

$(((((1)+(1))+((0.41899222015356274)+((1)/(n))+((0.41899222015356274)+(1)))))+($
 $((0.41899222015356274)+((n)/(p)))/(-$
 $0.10633125300813995)))+(1))+((((1)/(w))+((1)/(w)))*(((1)+(1))/((n)/(1))))*(((1)+($
 $1))/(w))+((1)+(1))/(w)))*(((1)+(1))/(w))/(w))*(((1)/(w))+((1)/(w)))/((n)/(1))))$

Koşma:12

Ham uygunluk değeri:176.667091

$$\begin{aligned} &(((1)*(1)*((-0.8278087264987681)/(-0.32798197089134806))+((- \\ & 0.8278087264987681)/(-0.32798197089134806))))-(((\\ & 0.6067709436671673)*(((1)+(1))/(w)))*((((-0.8278087264987681)/(- \\ & 0.32798197089134806))/(w))/(w))*((((-0.8278087264987681)/(- \\ & 0.32798197089134806))+((1)+(1)))/((n)-(1))/(w)))) \end{aligned}$$

Koşma:13

Ham uygunluk değeri:201.431082

$$\begin{aligned} &(((((((0.9970623627291351)/(0.14251646452700562))/(w))/(w))/((- \\ & 0.3155523106993885)+(w)))/((-0.3155523106993885)+((- \\ & 0.3155523106993885)+((0.9970623627291351)/(0.14251646452700562)))))+(((1)/ \\ & (0.14251646452700562))/(0.14251646452700562))/(((1)/(0.14251646452700562))+ \\ & ((0.9970623627291351)/(0.14251646452700562)))- \\ & (p)))+(((0.9970623627291351)/(0.14251646452700562))+(((1)/(p)))/((- \\ & 0.8888395335007643)/(p)))))) \end{aligned}$$

Koşma:14

Ham uygunluk değeri:218.163426

$$\begin{aligned} &((((1)+(1))-(-0.9856822732769336))+(((1)+(1))-(- \\ & 0.9856822732769336)))+((((1)+(1))-(-0.9856822732769336))*(((1)*(n))-(- \\ & 0.9856822732769336)))/(p))/(-0.2601198774111084)) \end{aligned}$$

Koşma:15

Ham uygunluk değeri:227.859016

$$\begin{aligned} &(((1)+(((1)+(0.82524113255797))+((1)+(1))))+(((1)+((1)+(0.82524113255797)))/(((\\ & n)*(-0.33020368746025663))+((n)*(- \\ & 0.33020368746025663)))+(p)/(p)))+(1)+(n)))) \end{aligned}$$

Koşma:16

Ham uygunluk değeri:191.660185

$$\begin{aligned} &(((((((1)+(1))/(w))/(w))/(w))/(w)/(((1)+(1))/(w))/(w)))+(((1)+(((1)/(n))- \\ & 0.2603160698247683)))+(((1)+(1))+((1)+(1))))-(-0.2603160698247683)) \end{aligned}$$

Koşma:17

Ham uygunluk değeri:203.645118

$$\begin{aligned} &((((0.2699242994514872)+(1))- \\ & ((n)+(n))/(p)))/(0.2434982681849498))+((((0.6145100349627948)/(w))/(w))+(((0.6 \end{aligned}$$

145100349627948)/(w)+(1))-((n)+(n)/(0.2434982681849498))/((1)-(n)-(p))))/(w)))

Koşma:18

Ham uygunluk değeri:172.568767

(((((1)+(1))+((1)+(1)))/(0.7094157793137319)-((1)/(n)))/((1)-(1))/((w)-(1))))+(((1)+(1))+0.23896494286086645))-((1)/(-0.5922717142814173)))+(0.7094157793137319)-(1)))+(1))

Koşma:19

Ham uygunluk değeri:187.857343

(((((0.9870398603319377)/(0.1365775135981746))*(0.9870398603319377))*(0.9870398603319377))+((1)/(0.1365775135981746))/((0.27566329169816495)/(-0.06406686411428586)))+(((1)/(w))/((-0.9207706218216432)+(n)))/(w))/(w))+(((1)/(0.1365775135981746))/((-0.9207706218216432)+(n)))/(w))/(w))/(w)))

Koşma:20

Ham uygunluk değeri:224.219008

(((((p)/(n))/(-0.9109845635997991))+(((0.9109845635997991)/(-0.12346932530564692))-(-0.2273978633933691)))*(((0.029882279704270154)/(n))/((-0.9109845635997991)/(-0.12346932530564692))+(-0.9109845635997991)))*(((0.9109845635997991)/(-0.12346932530564692))+(-0.9109845635997991))-((-0.9109845635997991)/(-0.12346932530564692))-(-0.9109845635997991))))+(((0.9109845635997991)/(-0.12346932530564692))+(-0.9109845635997991))-(-1)))

Koşma:21

Ham uygunluk değeri:148.850380

(((((0.7505287275419239)*(-0.658494356860637))+((1)-(-0.658494356860637)))/(w)+(-0.6360498208720469)))/((w)+(-0.658494356860637)))/(((w)*(n))-((0.7901244972786514)*(n))*(n))+(((p)/(1))+p))/(((1)-(-0.6360498208720469))+((1)-(-0.8991865189276573)))+(p))+(((p)/(1))+p))/(((0.7901244972786514)*(n))+p))+((1)-(-0.658494356860637))))))

Koşma:22

Ham uygunluk değeri:152.384456

$$\begin{aligned} & (((1)+(0.4786289145604843))+((1)+(1))+((1)+((n)-(- \\ & 0.45051302455855213)))/(n))))+((((1)+(1))+((1)+(1)))/(w))/(- \\ & 0.6180024555036809))/((((1)+(0.5780082743512505))/(w))- \\ & ((n)/((1)/(w))))*((((n)*(n))*((w)-(1)))-((w)/(p))-(w)))) \end{aligned}$$

Koşma:23

Ham uygunluk değeri:158.515435

$$\begin{aligned} & ((((((1)+(1))/(w))/(w))/(w))*(((w)+(w))-((p)-(-0.8145355516181028)))/((p)*(- \\ & 0.9221425363431417))*0.07164151457680457))/((n)*((n)- \\ & (1)))/((1)/(w))/(w))))+((1)+((1)+(1))+((0.979208923866604)/((1)+(1)))+(0.97920 \\ & 8923866604)+(1)))) \end{aligned}$$

Koşma:24

Ham uygunluk değeri:143.989170

$$\begin{aligned} & (((((0.7305251107227639)/(0.10574287041357855))+(((0.7305251107227639)/(0.1 \\ & 0574287041357855)))/(p))*((-0.4505926572953317)/(w))- \\ & ((1)/(0.10574287041357855))))-(1))+((0.10574287041357855)/((w)- \\ & (1))+((n)/(p)))) \end{aligned}$$

Koşma:25

Ham uygunluk değeri:204.686129

$$\begin{aligned} & (((1)*(p))/(((p)+(n))+((n))+((n)))*(((1)+(1)+(1)))+((1)*(p))/((p)+(n)))+(1)+(0.13 \\ & 622900896530088)))+((((1)+(1))/(w))+((1))/(w))) \end{aligned}$$

Koşma:26

Ham uygunluk değeri:139.194443

$$\begin{aligned} & (((((-0.9853525645962695)/(-0.20680656526554686))+((p)/(p)))+(((n)+((- \\ & 0.9853525645962695)/(-0.20680656526554686)))/((p)-((w)*(n)))/(- \\ & 0.20680656526554686)))+(((((-0.9853525645962695)/(- \\ & 0.20680656526554686))+((1)/(w))/(0.1866320563418009)))+(n)+((- \\ & 0.9853525645962695)/(-0.20680656526554686)))/((((1)*(w))- \\ & ((1)/(w)))/(((0.9480503741409865)/(n))/(p)))+((- \\ & 0.9853525645962695)+((w)*(n))*((-0.9853525645962695)*(1))+((w)*(n)))) \end{aligned}$$

Koşma:27

Ham uygunluk deęeri:149.737492

$$\begin{aligned} & (((((1)/(w))-((1)-(1)/(w)))+((0.7268670060331315)/(0.11161812353488898))))- \\ & (((((1)/(w))/((0.2942886153675097)*(n)))/(w))/(((0.0350392465642837)*(n))*(w))+ \\ & ((1)-(1)/(w)))))*(-0.8098549497359688) \end{aligned}$$

Koşma:28

Ham uygunluk deęeri:139.924212

$$\begin{aligned} & (((((1)/(n))/(((1)-(w))*((p)/(- \\ & 0.98483236505875))))+(((n)/(0.1459514382974949))/(p))))+(((n)/(0.1459514382974 \\ & 949))+((0.6515119605006241)/(0.1459514382974949)))/(((1)/(n))+1))*((1)*((p)/(- \\ & 0.98483236505875))))+(((0.6515119605006241)/(0.1459514382974949))-(((1)- \\ & (w))/(p))/(p))+((-0.9679848123192717)/(((0.8866893574067807)-(w))*(n)))+(1)))) \end{aligned}$$

Koşma:29

Ham uygunluk deęeri:194.633101

$$\begin{aligned} & ((((((0.8759571341220089)/(0.12505515530940858)))-(0.8478154045752322))- \\ & (((0.8759571341220089)/(0.12505515530940858)))/(0.12505515530940858))/(p))- \\ & (1))-(((n)/((1)/(0.12505515530940858)))-(1))-(((1)/(w))/(w))/(w))- \\ & (((1)/(w))/(w))/(w))) \end{aligned}$$

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı: İbrahim ŞAHAN
Uyruğu: Türkiye (T.C)

EĞİTİM

Derece	Kurum	Mezuniyet Tarihi
Lisans	Erciyes Üniversitesi, Bilgisayar Mühendisliği	2018
Lise	Melikgazi Mustafa Emiñoğlu Anadolu Lisesi, Kayseri	2014

İŞ DENEYİMLERİ

Yıl	Kurum	Görev
2021-Halen	Kocaeli Üniversitesi Bilgisayar Mühendisliği	Araştırma Görevlisi

YABANCI DİL

İngilizce

YAYINLAR

- Güner P., Şahan İ., Görkemli B., 2018. Maden işçilerinin kaza sırasında yönlendirilmesi problemi ve bu problem için alternatif çözüm yöntemlerinin incelenmesi, 85. 38. *Yöneylem Araştırması ve Endüstri Mühendisliği Ulusal Kongresi (YAEM)*, 26-29 Haziran, 2018, Eskişehir, Türkiye.