

T.C.
MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GÜNCEL YAPAY ZEKA TEKNİKLERİNİN
İNCELENMESİ VE EKSİKLİKLERİNİN GİDERİLMESİ
İÇİN HYBRİD ALGORİTMA TASARIMI

Yunus PAÇACI

YÜKSEK LİSANS TEZİ
ELEKTRONİK BİLGİSAYAR EĞİTİMİ ANABİLİM DALI
BİLGİSAYAR KONTROL EĞİTİMİ PROGRAMI

DANIŞMAN

Doç. Dr. Hasan ERDAL

İSTANBUL 2011

T.C.
MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GÜNCEL YAPAY ZEKA TEKNİKLERİNİN
İNCELENMESİ VE EKSİKLİKLERİNİN GİDERİLMESİ
İÇİN HYBRİD ALGORİTMA TASARIMI

Yunus PAÇACI

522108009

YÜKSEK LİSANS TEZİ

ELEKTRONİK BİLGİSAYAR EĞİTİMİ ANABİLİM DALI

BİLGİSAYAR KONTROL EĞİTİMİ PROGRAMI

DANIŞMAN

Doç. Dr. Hasan ERDAL

İSTANBUL 2011

TEŐEKKÖR

Çalıőmalarımın ve eđitimimin her safhasında bana yol gősteren, tez danıőmanım sayın Doç. Dr. Hasan ERDAL'a, tecrübelerini ve manevi desteklerini esirgemeyen sayın Doç. Dr. Mustafa ONAT'a, ve insan üőtü bir sabır ile bana katlanan aileme teőekkürlerimi sunarım.

İÇİNDEKİLER

	SAYFA
TEŞEKKÜR.....	III
İÇİNDEKİLER	IV
ÖZET	VII
ABSTRACT	VIII
SEMBOLLER	IX
KISALTMALAR	X
ŞEKİLLER	XI
TABLolar	XII
BÖLÜM I GİRİŞ VE AMAÇ	1
I.1 GİRİŞ	1
I.2 GENETİK ALGORİTMA UYGULAMALARI	2
I.3 KARINCA KOLONİSİ OPTİMİZASYONU UYGULAMALARI	3
I.4 GENETİK ALGORİTMALAR İLE EN KISA YOL BULMAYA YÖNELİK GEÇMİŞ ÇALIŞMALAR.....	3
I.5 KARINCA KOLONİSİ OPTİMİZASYONU İLE EN KISA YOL BULMAYA YÖNELİK GEÇMİŞ ÇALIŞMALAR.....	6
BÖLÜM II EN KISA YOLU BULMA PROBLEMİ.....	10
II.1 GİRİŞ.....	10
II.2 EN KISA YOLU BULMA PROBLEMİNİN GENEL TANIMI.....	10
II.2.1 Mevcut En Kısa Yol Probleminin Yapısı.....	11
II.2.2 Mevcut Problem İçin Görüntü İşleme İle Ortam Oluşturulması.....	12
II.2.3 Çözüm Uzayının Noktalara İndirgenmesi.....	15

BÖLÜM III GENETİK ALGORİTMALAR VE KARINCA

KOLONİSİ ALGORİTMASI.....	20
III.1 GİRİŞ	20
III.2 GENETİK ALGORİTMALAR.....	20
III.2.1 Giriş ve Tanımı	20
III.2.2 Tarihçesi	22
III.2.3 Temel Bileşenleri ve Yapısı.....	22
III.2.3.1 Çözümlerin Kodlanması	23
III.2.3.2 İlk Popülasyonun Oluşturulması.....	23
III.2.3.3 Uygunluk Değerinin Hesaplanması	23
III.2.3.4 Çoğalma İşleminin Uygulanması.....	24
III.2.3.5 Çaprazlama İşleminin Uygulanması	25
III.2.3.6 Mutasyon İşleminin Uygulanması	25
III.2.3.7 Yeni Kuşağın Oluşması ve Döngünün Durdurulması.....	26
III.2.3.8 İlgili Parametre Seçimleri	27
III.2.4 Avantajları	28
III.2.5 Dezavantaj ve Kısıtlılıkları	30
III.2.6 Tartışma	31
III.3 KARINCA KOLONİSİ ALGORİTMASI.....	34
III.3.1 Giriş ve Tanımı	34
III.3.2 Tarihçesi	34
III.3.3 Temel Bileşenleri ve Yapısı.....	35
III.3.3.1 İlk Değerlerin Oluşturulması	36
III.3.3.2 Başlangıç Durumlarının Belirlenmesi.....	36
III.3.3.3 Karınca Turlarının Gerçekleştirilmesi	37
III.3.3.4 Feromon Düzeylerinin Güncellenmesi	38
III.3.4 Avantajları	38

III.3.5 Dezavantajları ve Kısıtlılıkları.....	40
III.3.6 Tartışma	40
BÖLÜM IV HYBRİD ALGORİTMA VE UYGULAMA.....	42
IV.1 GİRİŞ	42
IV.2 GENETİK ALGORİTMALARIN HYBRİD ALGORİTMADAKİ ETKİSİ.....	42
IV.3 KARINCA KOLONİSİ ALGORİTMASININ HYBRİD ALGORİTMADAKİ ETKİSİ	45
IV.4 OLUŞTURULAN HYBRİD ALGORİTMA	47
IV.5 ALGORİTMANIN PROBLEME UYGULANMASI	49
IV.5.1 Hybrid Algoritmanın KKA'dan Alınan İşlemlerinin Gerçekleştirilmesi.....	50
IV.5.2 Yol İyileştirme İşlemi	51
IV.5.3 Yeni Nesil Yollar İle Önceki Nesillerin Aynı Havuzda Tutulması ...	52
IV.5.4 Hybrid Algoritmanın GA'dan Alınan İşlemlerinin Gerçekleştirilmesi	52
IV.6 UYGULANAN ALGORİTMANIN ÖRNEK SONUÇLARI	55
IV.7 ALGORİTMA PERFORMANSINA YÖNELİK DEĞERLENDİRMELER.....	58
IV.7.1 Görüntü Boyutu	58
IV.7.2 Engel Çeşitliliği	59
IV.7.3 Düğüm Sayısı.....	59
IV.7.4 Genetik İşlemler Sonrası Düğüm Sayıları	60
IV.7.5 İşlem Zamanı	61
IV.7.6 Sonuç Kalitesi	61
BÖLÜM V SONUÇ VE ÖNERİLER	62
KAYNAKLAR.....	65
ÖZGEÇMİŞ.....	71

ÖZET

GÜNCEL YAPAY ZEKA TEKNİKLERİNİN İNCELENMESİ VE EKSİKLİKLERİNİN GİDERİLMESİ İÇİN HYBRİD ALGORİTMA TASARIMI

Bu çalışmada, engellerin bulunduğu ortamda en kısa yol bulma probleminin çözümüne yönelik kullanılan güncel yapay zeka tekniklerinin incelenmesi ve probleme yönelik yeni yapay zeka algoritmalarının geliştirilmesi amaçlanmıştır. Bu amaçla, ilgili problem üzerinde kullanım popülarlığı yüksek olan, genetik algoritmalar (GA) ve karınca kolonisi algoritması (KKA) incelenmiştir. Adı geçen algoritmaların avantaj ve dezavantajları araştırılıp tespit edilerek, algoritmaların performanslarına yönelik tartışmalar gerçekleştirilmiştir.

Çalışmanın ilk basamaklarında problemin tespiti ve uygulama öncesi hazırlığı gerçekleştirilmiştir. Problemin görselleri olarak kabul edilen, engellerin bulunduğu ortam görüntüleri, problemin matematiksel denklem ve algoritmalara uygulanabilmesi için işlenmekte ve sayısal verilere dönüştürülmektedir. Görüntü işleme işlemleri sırasında kullanılan denklem ve yöntemlerin çoğu, istenilen amaca en iyi şekilde ulaşabilmek için, bu çalışma sırasında üretilmiş denklem ve yöntemlerdir.

İlerleyen aşamalarda GA ve KKA'nın irdelenmesi gerçekleştirilmektedir. GA ve KKA'nın irdelenmesi sonucu tespit edilen avantajları geliştirilen hybrid algoritmaya aktarılmaya çalışılmış, dezavantaj ve kısıtlılıkları ise mümkün olduğunca etkisiz kılınmaya çalışılmıştır.

Oluşturulan hybrid algoritma performans değerlendirmesi için çeşitli problem tipleri üzerinde denenmiş ve örnek çözümler oluşturulmuştur. Örnek çözümlerin değerlendirilmesi sonucunda, oluşturulan hybrid algoritmanın benzer işi yapan algoritma tiplerine oranla daha iyi sonuçlar ürettiği gözlemlenmiştir.

Eylül 2011

Yunus PAÇACI

ABSTRACT

INVESTIGATION OF CURRENT ARTIFICIAL INTELLIGENCE TECHNIQUES AND DESIGNING OF A HYBRID ALGORITHM TO ELIMINATE THE SHORTCOMINGS OF THESE TECHNIQUES

In this study, the research of current artificial intelligence techniques is used for in the presence of barriers to finding the shortest path problems, and to develop new problem-oriented artificial intelligence algorithms are aimed. For this purpose, Genetic Algorithm (GA) and Ant Colony Optimization Algorithm (ACO) were investigated which are the high popularity of the use on the related problem. Advantages and disadvantages of the mentioned algorithms will be identified and explored, and the discussions performed for the performance of the algorithms.

In the first stages of the study, the identification of the problem and preparation of the pre-application is performed. Including barriers media images which are the considered for the images of the problem are processed and converted into digital data in order to apply the problem to the algorithms and math equations. Most of the used equations and methods, during image processing processes, are equations and methods which are generated to achieve the desired purpose by the best way during this study.

In the following stages, the investigations of GA and ACO are performed. Determinated advantages of GA and ACO's investigation result is tried to transfer to developed hybrid algorithm, as well as disadvantages and limitations of these algorithms are tried to be ineffective as much as possible.

The developed hybrid algorithm is tried on the various problem types for performance evaluation and created exemplary solutions. As a result of evaluation of exemplary solutions, developed hybrid algorithm is produced better results than other similar algorithm types are observed.

September 2011

Yunus PAÇACI

SEMBOLLER

X	: Bitmap Görüntüsünün Piksel Bazında Genişlik Deęeri
Y	: Bitmap Görüntüsünün Piksel Bazında Yükseklik Deęeri
m	: Görüntüden Elde Edilen Başlangıç Düęüm Sayısı
n	: Genetik İşlemler Sonrası Kalan Düęüm Sayısı
YD	: Bulunan En Kısa Yolun Düęüm Sayısı

KISALTMALAR

YZ	: Yapay Zeka
GA	: Genetik Algoritma
SP	: Shortest Path Problem (En Kısa Yol Bulma Problemi)
ACO	: Ant Colony Optimization (Karıncı Kolonisi Optimizasyonu)
AS	: Ant System (Karıncı Sistemi)
TSP	: Travelling Salesman Problem (Gezgin Satıcı Problemi)
KKO	: Karıncı Kolonisi Optimizasyonu
KKA	: Karıncı Kolonisi Algoritması

ŞEKİLLER

Şekil II.1 Graph yapısı	11
Şekil II.2 Piksel yapısı ve piksel koordinatları	11
Şekil II.3 Orijinal görüntü.....	13
Şekil II.4 Siyah beyaz görüntü.....	13
Şekil II.5 Diyagonal engel tanımlama.....	13
Şekil II.6 Siyah beyaz görüntü.....	15
Şekil II.7 Çözüm uzayı görüntüsü	15
Şekil II.8 Çözüm uzayı	16
Şekil II.9 Teğet noktaları	16
Şekil II.10 Çizgilerdeki düğüm sayısının azaltılması	17
Şekil II.11Teğet kaplamalar	18
Şekil II.12 İndirgenmiş düğümler	18
Şekil II.13 Genel kapsamlı görüntü	19
Şekil IV.1 GA akış diyagramı	43
Şekil IV.2 KKA akış diyagramı	45
Şekil IV.3 Hybrid algoritma akış diyagramı	49
Şekil IV.4 Yol kayıt listesi ve örnek yollar	51
Şekil IV.5 Tüm olası çözüm düğümleri	54
Şekil IV.6 İlk yolların kullandığı düğümler	54
Şekil IV.7 Birinci genetik seçim sonrası kullanılan düğümler.....	54
Şekil IV.8 İkinci genetik seçim sonrası kullanılan düğümler	54
Şekil IV.9 Üçüncü genetik seçim sonrası kullanılan düğümler	54

TABLolar

Tablo I.1 GA ile en kısa yol bulmaya yönelik çalışmalar	4
Tablo I.2 KKA ile en kısa yol bulmaya yönelik çalışmalar.....	6
Tablo IV.1 Yol iyileştirme örneđi	52
Tablo IV.2 Örnek çözüm görüntüleri	55

BÖLÜM I GİRİŞ VE AMAÇ

I.1 GİRİŞ

Günümüzde teknolojinin her geçen gün biraz daha gelişmesi ile bilgisayar ve bilgisayar sistemleri hayatımızın önemli bir parçası haline gelmiştir. Günlük hayatta kullandığımız birçok elektronik alet bilgisayar tabanlı bir sistem ile çalışmaktadır. Peki, bu bilgisayar ihtiyacı ve kullanımı nereden doğmuştur? Bilgisayarlar ilk olarak insan beyninin bir taklidi olarak ortaya atılmaya çalışan hesap makinelerinin gelişmiş şekilleridir. Gün geçtikçe bu teknolojinin geliştirilmesi ile hem yüksek hızlı işlem yapması amaçlanan bilgisayarlar gelişmiş hem de bu bilgisayarlar hayatımızın yüksek hızda devam etmesini sağlayarak birbiri ile iç içe geçmiş bir ihtiyaç döngüsü oluşturmuştur. Teknolojiyi geliştirme çabaları bilgisayar ile alakalı disiplinleri ileriki aşamalarda bölerek farklı dallara ayırmıştır. Kimi araştırmacılar bu teknolojinin daha elverişli hale getirilmesi ile uğraşarak bilgisayarları hızlı ve küçük boyutlarda üretmeye çalışırken, kimi araştırmacılar bu teknolojinin doğuş amacına uygun olarak insan davranışlarının modellenmesi amacına hizmet etmiş ve Yapay Zeka (YZ) adı verilen bilimi sürdürmüşlerdir. Genel olarak 1950'li yıllarda başlayan yapay zeka akımı yoğun ilgi görerek günümüze kadarki zaman diliminde yüksek teknolojik gelişmelerin oluşmasında önemli rol oynamıştır.

Yapay zeka, kabaca; bir bilgisayarın ya da bilgisayar denetimli bir makinenin, genellikle insana özgü nitelikler olduğu varsayılan akıl yürütme, anlam çıkartma, genelleme ve geçmiş deneyimlerden öğrenme gibi yüksek zihinsel süreçlere ilişkin görevleri yerine getirme yeteneği olarak tanımlanmaktadır. [1]

Özellikle son yıllarda geliştirilen yapay zeka algoritmaları disiplinin kendisinden çok problemlerin çözümünde etkili olmak amacıyla geliştirilmektedir. Gerçekleştirilen çalışmalar günümüzde yüksek ilerleme kat etmiş olsa da problemlerin çözümüne yönelik olarak hala kısıtlılıklar ve eksiklikler mevcuttur. Bu kısıtlılık ve eksikliklerin giderilmesine yönelik geliştirilen hybrid (melez) algoritma adı verilen birleşik yapay zeka algoritmaları kısmen faydalı olmaktadır.[2]

En kısa yol bulma problemi (shortest path problem) güncel yapay zeka uygulamalarının son yıllarda popüler olarak ilgilendiği bir problemdir. Bu konuda genetik algoritmalar (genetic algorithm - GA) ve karınca kolonisi optimizasyonu (ant colony optimization - ACO) ile çalışmalar gerçekleştirilmiş ve geçerli sonuçlar elde edilmiştir. Elbette bu algoritmalar sadece bu probleme uygulanabilirlikleri ile değil benzer optimizasyon problemleri için uygulanabilirlikleri ile de değerlendirilmekte ve kullanılmaktadır. Bu amaçla bu algoritmaların performansının artırılmasına yönelik çalışmalar hala devam etmektedir.[3,4]

I.2 GENETİK ALGORİTMA UYGULAMALARI

GA, rastlantısal arama teknikleri kullanarak çözüm bulmaya çalışan, parametre kodlama esasına dayanan sezgisel bir arama tekniğidir[5]. GA, canlıların en iyi olanı yaşar prensibini örnek almaktadır ve iyi bireylerin kendi yaşamlarını muhafaza edip kötü bireylerin yok olması prensibine dayanmaktadır.

GA, ebeveyn bireylerden oluşan yeni bireylerin şartlara uyum sağlayıp yaşamlarını devam ettirmesi prensibine sahiptir. Oluşan yeni bireylerin ebeveynlerinden iyi özellikli genleri alabilecekleri gibi kötü özellikteki genleri de alabilmeleri mümkündür. Bu durum söz konusu olunca kötü özellikteki genlere sahip bireyler yaşamlarını devam ettiremeyeceklerdir[6].

Michigan Üniversitesinde psikoloji ve bilgisayar bilimi uzmanı olan John Holland bu konuda ilk çalışmaları yapan kişidir. Mekanik öğrenme (machine learning) konusunda çalışan Holland, Darwin'in evrim kuramında etkilenecek canlılarda yaşanan genetik süreci bilgisayar ortamında gerçekleştirmeyi düşünmüştür. Tek bir mekanik yapının öğrenme yeteneğini geliştirmek yerine böyle yapılarda oluşan bir topluluğun çoğalma, çiftleşme, mutasyon, vb. genetik süreçlerden geçerek başarılı (öğrenebilen) yeni bireyler oluşturabildiğini görmüştür. Çalışmalarının sonucunu açıkladığını kitabının 1975'te yayınlanmasından sonra geliştirdiği yöntemin adı Genetik Algoritmalar olarak yerleşmiştir. Ayrıca kitabında GA'lara dayalı tam 83 uygulamaya yer vererek GA'nın dünyanın her yerinde çeşitli konularda kullanılmakta olduğunu göstermiştir[7].

GA'ların uygulama alanlarına ve çeşitli problem tipleri üzerindeki örneklerine ait geniş kapsamlı bir çalışma 1989 yılında Goldberg tarafından

gerçekleştirilmiştir[8]. GA'nın en kısa yol bulma problemlerine uygulanmasına örnek olarak Braun'un yaptığı çalışma verilebilir[9]. Benzer şekilde Freisleben ve Merz'in çalışmasında da GA ile gezgin satıcı probleminin çözümüne dair çalışma gerçekleştirilmiştir[10-12].

I.3 KARINCA KOLONİSİ OPTİMİZASYONU UYGULAMALARI

Karınca Kolonisi Optimizasyonu (KKO) bir çok kombinasyonel optimizasyon problemlerinde iyi sonuçlar veren bir meta-heuristic tekniktir. Bu tekniğin geliştirilmesinde gerçek karınca kolonilerinin gıda arama tekniklerinden faydalanılmıştır.

Birçok karınca kolonisinde karıncalar yiyeceklerini ararken, öncelikle yuvalarının etrafında rastgele dolaşarak keşfe başlarlar. Yiyecek kaynaklarını bulduklarında, yiyeceğin kalitesini ve miktarını değerlendirdikten sonra bir kısmını yuvaya taşırlar. Bu dönüş sırasında diğer karıncaların da aynı kaynağı bulabilmeleri için yiyeceğin kalitesine ve miktarına bağlı olarak kimyasal feromon (pheromone) maddesini geçtikleri yolun üzerine bırakırlar. Bırakılan bu izler diğer karıncalara rehberlik ederek belli olasılıkla o yolu takip etmelerini ve kaynağı bulmalarına yardım eder. Bu şekilde feromon vasıtasıyla yapılan dolaylı iletişim karıncaların gıda ile yuva arasında en kısa yolu bulmalarına olanak tanır. İşte karıncaların bu davranışları KKO algoritmalarının geliştirilmesinde ilham kaynağı olmuştur.[13]

KKO algoritması 1991 yılında Marco Dorigo tarafından doktora tezi olarak geliştirilmiş ve yayınlanmıştır. Ant System (AS) adı verdiği bu algoritmayı daha sonra Dorigo çeşitli şekillerde kullanmış ve geliştirmiştir [4,14-17]. Özellikle gezgin satıcı problemi (travelling salesman problem - TSP) üzerinde kullanılan algoritma kısa zamanda adından söz ettirmiş ve birçok farklı uygulama ile karşımıza çıkmıştır.

I.4 GENETİK ALGORİTMALAR İLE EN KISA YOL BULMAYA YÖNELİK GEÇMİŞ ÇALIŞMALAR

Genetik algoritmalar kullanılarak gerçekleştirilen en kısa yol bulmaya yönelik çalışmalar Tablo I.1 deki gibi özetlenebilmektedir:

Tablo I.1 GA ile en kısa yol bulmaya yönelik çalışmalar

Yazar	Yıl	Başlık	Açıklama
Gen, Chang ve Wang[18]	1997	Genetic Algorithms For Solving Shortest Path Problems	SP problemleri üzerine GA uygulanması
Soehodho[19]	1997	Development Of Genetic-Taxonomy Evaluator For Finding Shortest Path İn Transportation Planning Systems	Nakliye planlamada SP bulunması için GA kullanımı ve ikinci bir kriter olarak ağırlıklı genetik işlem yapılması
Ahn ve Ramakrishna[3]	2002	A Genetic Algorithm For Shortest Path Routing Problem And The Sizing Of Populations	SP çözümü için GA kullanımı ve kullanım için gerekli genetik popülasyon büyüklüğü üzerine değerlendirme
Yu ve Wu[20]	2002	Loads Combination Method Based Core Schema Genetic Shortest-Path Algorithm For Distribution Network Reconfiguration	Ağ dizaynı için SP algoritmalarının kullanımı sonrası GA ile optimum çözümün bulunması
Selvanathan ve Tee[21]	2003	A Genetic Algorithm Solution To Solve The Shortest Path Problem İn Ospf And Mpls	Ağlardaki OSPF (Open Shortest Path First) ve MPLS (Multi Protocol Label Switching) yönlendirme protokollerinin yerine GA kullanımı
Abeyesundara, Giritharan ve Kodithuwakku[22]	2005	A Genetic Algorithm Approach to Solve the Shortest Path Problem for Road Maps	Yol haritaları üzerinde SP problemine GA ile çözüm üretilmesi
Kan[23]	2007	Shortest Path Analysis Based On Genetic Algorithm	SP algoritmaları ve MATLAB üzerindeki GA algoritmasının karşılaştırılması ve performans değerlendirmesi

Lin, Lee ve Fuh[24]	2007	A Two-Stage Genetic Algorithm for Solving Shortest Path Problem with Fuzzy Arc Lengths	SP probleminin çözümüne yönelik olarak ilk aşamada Fuzzy Algoritması kullanan ve ikinci aşamada ise GA kullanarak çözüm üreten bir çalışma gerçekleştirilmesi
Lin ve Gen[25]	2007	A Bicriteria Shortest Path Routing Problems By Hybrid Genetic Algorithm In Communication Networks	Ağlardaki iletişim için maliyet ve minimum gecikme kriterlerini göz önüne alan ve parametrelerini otomatik olarak kendisi belirleyen GA tasarımı ve uygulaması
Pennisi, Pappalardo, Motta ve Cincotti[26]	2007	A Genetic Algorithm for Shortest Path Motion Problem in Three Dimensions	SP problemine üçüncü bir boyut olan yükseklik etkeninin de eklenmesi ve GA ile probleme çözüm bulunması
Behzadi, Alesheikh ve Poorazizi[27]	2008	Developing A Genetic Algorithm To Solve Shortest Path Problem On A Raster Data Model	Krokiler üzerinde GA ile en kısa yolun bulunması
Pires, da Silva ve Bertoni[28]	2008	Solving Shortest Path Problem Using Hopfield Networks And Genetic Algorithms	GA ile modifiye edilmiş bir Hopfield ağıyla SP problemlerine çözüm üretilmesi
Yussof, Razali ve See[29]	2009	A Parallel Genetic Algorithm for Shortest Path Routing Problem	Ağlardaki paket iletişimde gerçek zamanlı GA kullanılabilmesi amacıyla geliştirilmiş, paralel işlem ile zaman kazancı sağlayan GA tasarımı

Nallusamy, Duraismamy, Muthukumar ve Sathiyakumar[30]	2010	Energy Efficient Dynamic Shortest Path Routing İn Wireless Ad Hoc Sensor Networks Using Genetic Algorithm	Dinamik wireless paket taşımada sensörlerin enerji seviyesini de göz önüne alan GA optimizasyonu uygulaması
--	------	---	---

I.5 KARINCA KOLONİSİ OPTİMİZASYONU İLE EN KISA YOL BULMAYA YÖNELİK GEÇMİŞ ÇALIŞMALAR

Karınca kolonisi optimizasyonu kullanılarak gerçekleştirilen en kısa yol bulmaya yönelik çalışmalar Tablo I.2 deki gibi özetlenebilmektedir:

Tablo I.2 KKA ile en kısa yol bulmaya yönelik çalışmalar

Yazar	Yıl	Başlık	Açıklama
Yuhe,[31]	2003	An Improved Ant Colony Algorithm for the Shortest Path Problem	SP problemleri üzerine KKO uygulanması
Jinjiang ve Da[32]	2004	Solving a Shortest Path Problem by Ant-algorithm	Matematik modeli bilinen bir SP problemi üzerinde KKO algoritması uygulanması ve sonuçların değerlendirilmesi
Khan[33]	2004	Ant System To Find The Shortest Path	SP problemleri üzerine Dorigo'nun ilk geliştirdiği Ant System algoritmasının uygulanması ve eleştirilmesi
Li ve Jin-song[34]	2005	Multi-Thread Ant Colony Algorithm And Its Application İn The Shortest Path	Multi-Thread teknolojisi yardımı ile her bir karıncanın bir thread olarak belirlenmesi ve algoritmanın bu şekilde SP uygulamalarında kullanılması üzerine bir çalışma

Liu, Lin ve Lin[35]	2005	A Shortest-Path Network Problem Using An Annealed Ant System Algorithm	Ağlardaki SP problemi üzerine KKO algoritmasının uygulanması ve yön seçim işleminde kullanılan iki farklı yöntemin (karıncaların merkezi dağılımı veya başlangıçta dağılımı) karşılaştırılması
Min ve Yang[36]	2005	A Shortest-Path Routing Based On Ant Algorithm	Ağ üzerindeki paket yönlendirmesi için bir SP problemi ve KKO algoritması ile matematik model performanslarının karşılaştırılması
Shang ve Jingyu[37]	2006	Convergence Analysis of Ant Colony Algorithm for Solving Shortest Path Problem	Gerçek bir karınca kolonisi ile KKO algoritmasının sonuçlarının yakınlığının tartışılması üzerine bir çalışma
Chibaya ve Bangay[38]	2007	A Probabilistic Movement Model For Shortest Path Formation In Virtual Ant-Like Agents	KKO algoritmalarındaki feromon salgısına ilişkin kaynaktan hedefe, hedeften kaynağa ve sonuç ağırlığına göre olmak üzere üç farklı yöntem öne süren ve bunları değerlendiren bir çalışma
Jiang, Wang ve Zhao[39]	2007	Solving The Shortest Path Problem In Vehicle Navigation System By Ant Colony Algorithm	Araç rotalama problemi üzerinde feromon güncelleme algoritması değiştirilmiş bir KKO algoritması uygulanması ve sonuçların Dijkstra algoritması ile karşılaştırılması

Shuang-hua, Yang ve Liang[40]	2007	Study on Shortest Path Search Method Based on Chaos Ant Colony Optimization	KKO algoritması üzerinde, yerel optimum noktalarından kurtulmak amacıyla geliştirilmiş ve Chaos Ant Colony Optimization adı verilen bir algoritmanın trafikteki SP problemi üzerine uygulanması ve değerlendirilmesi
Wen-qun[41]	2008	Study On Advanced Ant Colony Algorithm Searching Shortest Path In Logistics Network Of Distribution	Lojistik dağıtım ağlarında KKO algoritması kullanımı ve sonuç iyileştirme için çözüm alanının kaynaktan hedefe doğru giderek daraltılmasını içeren bir uygulama
Gao, Liu, Chang ve Xiong[42]	2009	An Improved Ant Colony Algorithm for the Shortest Path Problem in Time-Dependent Networks	Zamana bağlı ağlarda SP problemine geliştirilmiş feromon güncelleme yöntemli bir KKO algoritması uygulaması ve sonuçların değerlendirilmesi.
Ghoseiri ve Nadjari[43]	2009	An Ant Colony Optimization Algorithm For The Bi-Objective Shortest Path Problem	İki kriterli bir SP problemi için KKO algoritması kullanılması ve sonuçların değerlendirilmesi
Ok, Seo, Ahn, Kang ve Moon[44]	2009	An Ant Colony Optimization Approach For The Preference-Based Shortest Path Search	Noktalar arası bağlantı özelliklerinin de göz önünde tutularak (yoğunluk vb.) sadece en kısa yolun değil yolun uygunluğu gibi özelliklerinde değerlendirildiği bir KKO algoritmasının SP problemlerinde uygulanması

Zhong-qiu, Zhang ve Zheng[45]	2009	Analysis and Simulation of Shortest Path Problem Based on Modified Ant Colony Algorithm	Lokal minimum noktasına takılma oranını azaltmayı amaçlayan geliştirilmiş bir KKO algoritmasının uygulaması ve analizini içeren bir çalışma
Yu, Li ve Wu[46]	2010	A Temporal Ant Colony Optimization Approach To The Shortest Path Problem In Dynamic Scale-Free Networks	Boyutu dinamik olarak değişen ağlar üzerinde değiştirilmiş bir KKO algoritmasının uygulanması ve analizi
Zakzouk, Zaher ve El-Deen[47]	2010	An Ant Colony Optimization Approach For Solving Shortest Path Problem With Fuzzy Constraints	Yol değerlerini ve uzaklıklarını ağırlık olarak belirleyen ve olasılıksal işlemler yapan geliştirilmiş bir KKO algoritması ve uygulaması

Bu çalışmada engellerin bulunduğu bir ortamdaki en kısa yol bulma çalışmalarına yönelik güncel yapay zeka algoritmaları değerlendirilerek, hybrid algoritma tasarımı sağlanacaktır.

Bölüm II’de problemin tanımlanması ve en kısa yolun tespiti işlemi öncesi ön hazırlıklar anlatılmıştır.

Bölüm III’de problemin çözümüne yönelik kullanılan GA ve KKA algoritmalarının irdelenmesinin yanı sıra avantaj ve dezavantajları aktarılmıştır.

Bölüm IV’de oluşturulan hybrid algoritma ve probleme uygulanmasının yanı sıra, hybrid algoritmanın performans değerlendirmeleri aktarılmıştır.

Bölüm V’de elde edilen sonuçlar irdelenmiş ve öneriler sunulmuştur.

BÖLÜM II EN KISA YOLU BULMA PROBLEMİ

II.1 GİRİŞ

Bu bölümde, en kısa yol bulma probleminin oluşturulduğu ortamın yanı sıra hybrid algoritma kullanımı öncesi yapılan işlemler kısaca açıklanarak olası problem tipleri örneklenecektir.

II.2 EN KISA YOLU BULMA PROBLEMİNİN GENEL TANIMI

En kısa yol bulma problemleri genel olarak verilen iki nokta veya düğüm arası mümkün olan yolların en kısa olanını bulmayı amaçlayan problem tipleridir. Sözü geçen başlangıç ve bitiş noktaları en kısa yolun bulunmasına yönelik veri tipi zaman yol uzunluğu veya düğümler arası ağırlıklar olabilmektedir[48].

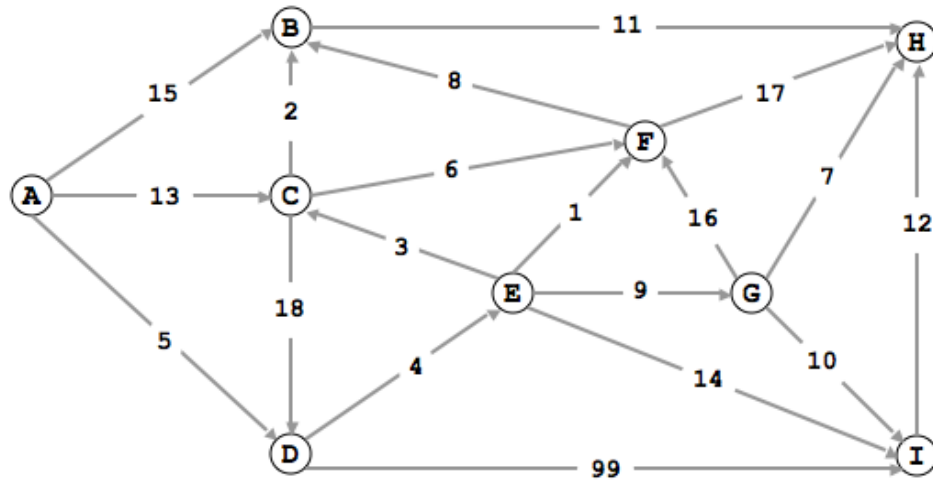
En kısa yol bulma problemlerinde matematiksel çözümler graphlar yardımı ile yapılmaktadır. İlgili problem graph adı verilen matematiksel ağırlıkların bulunduğu veri ağaçlarına aktarıldıktan veya bu şekilde açıklığa kavuşturulduktan sonra en çok kullanılan Dijkstra, Bellman-Ford ve Floyd algoritmaları ile çözülmektedir. Ancak bu çözüm yöntemleri düğüm sayısının yüksek olduğu durumlarda istenilen çözümü sunsa dahi optimum çözüm süresinden uzaklaşmaktadır. Genetik algoritmalar ve karınca kolonisi algoritması gibi sezgisel algoritmalar kullanılarak çözüme daha kısa sürede ulaşılmaya çalışılmaktadır[49].

En kısa yol bulma problemlerinin bir türevi olan engellerin bulunduğu bir ortamda en kısa yol bulma problemi ise günümüz araştırma ve uygulamalarında sıkça rastlanan ve çözümlenmeye çalışılan bir problem tipidir. Gerek robotlarda gerekse araç çizelgeleme problemlerinde karşımıza çıkan engellerin bulunduğu bir ortamda en kısa yol bulma problemi, olası tüm düğümler arası bağlantıların mümkün olmadığı, bazı düğümler arası bağlantıların engeller tarafından geçersiz kılındığı problem tipidir. Bu problem tipinde en kısa yol, engellerinde etkisi hesaplanarak bulunmakta ve yine engellere rağmen mümkün olan en kısa yol minimum zamanda ortaya koyulmaya çalışılmaktadır[50].

II.2.1 Mevcut En Kısa Yol Probleminin Yapısı

Mevcut en kısa yol bulma probleminde graph yapısı yerine dijital bir resimdeki pikseller düğümler olarak kabul edilmektedir. Genel graph yapılarındaki düğümler arası ağırlıklar yerine pikseller arası uzaklıklar ele alınarak işlem gerçekleştirilmektedir. Problemdaki engeller de yine pikseller olarak ele alınmıştır.

Genel en kısa yol bulma problemlerine ait graph yapısı ve ilgili probleme ait piksellerin düğümler kabul edildiği yapılar Şekil II.1 ve Şekil II.2 de gösterilmektedir.



Şekil II.1 Graph yapısı

(0, 0)	(0, 1)	(0, 2)	(0, 3)
(1, 0)	(1, 1)	(1, 2)	(1, 3)
(2, 0)	(2, 1)	(2, 2)	(2, 3)
(3, 0)	(3, 1)	(3, 2)	(3, 3)

Şekil II.2 Piksel yapısı ve piksel koordinatları

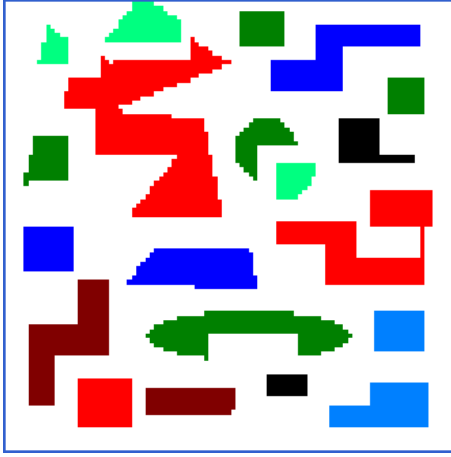
II.2.2 Mevcut Problem İin Grnt İřleme İle Ortam Oluřturulması

Problemin oluřturulması iin, genel tabir ile grnt iřleme olarak adlandırılan ve kullanıcıdan alınan grntlerin (problemin getiđi ortam resimlerinin bitmap formatındaki hali) matematiksel fonksiyonlar tarafından kullanılabilir hale getirildiđi bir dizi iřlem gerekleřtirilmektedir. Grnt iřleme iřlemi iin MATLAB 7.4.0 ortamı kullanılmaktadır.

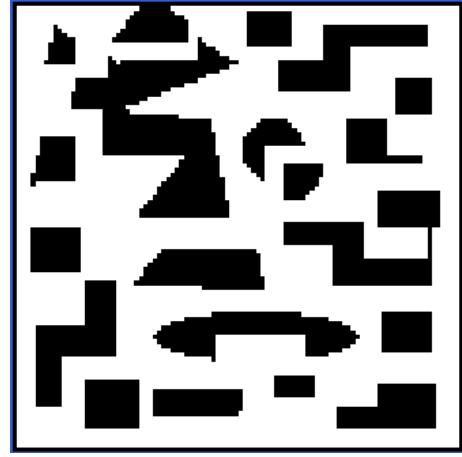
Problemin engellerin bulunduđu bir ortamda gerekleřmesinden dolayı ilk olarak engellerin ele alınan grnt üzerinde belirlenmesi gerekmektedir. Bu iřlem iin daha ncede sz edildiđi gibi piksellerden oluřan bir ortamda yine pikseller üzerinden engelleri tanımlama durumu sz konusudur.

Kullanıcıdan alınan grntlerdeki engel tanımlama iřlemi iin piksellere ait belli bir eřik deđerinin stndeki renk deđerlerinin (renk skalasında genel resim renk ortalamasında siyaha daha yakın renkler) engel olarak tanımlanması iřlemi gerekleřtirilmektedir. Renk deđerlerine gre engel tanımlama iřlemi iin MATLAB 7.4.0 programının hazır fonksiyonları kullanılmıřtır.

Grnt iřlemenin ilk basamađında kullanıcıdan alınan renkli grntnn, oluřturulan grnt iřleme formllerinde kullanılabilir hale getirilmesi amacı ile siyah-beyaz grntye dnřm bulunmaktadır. Őekil II.3'de rneđi grlen kullanıcıdan alınan grntnn Őekil II.4'de grlen siyah-beyaz grntye dnřtrlmesinin ardından istenilen boyutta bir ara iin siyah-beyaz grntnde zm uzayı oluřturma yani hareket edilebilir blgelerin belirlenmesi iřlemine geilmektedir.



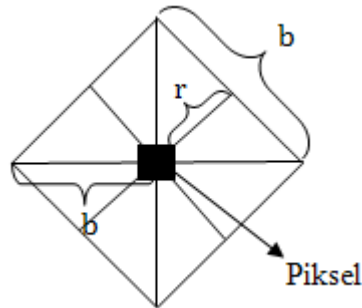
Şekil II.3 Orijinal görüntü



Şekil II.4 Siyah beyaz görüntü

Engel tanımlamada (çözüm uzayı oluşturma) gerçekleştirilen işlemlerde fonksiyonel olarak bir engelin ne gibi bir araç için engel olduğunun belirlenmesi gerekmektedir. Bu durum güncel problemler üzerinden örnek verilecek olunursa bir robotun geçebileceği yol ile bir ağır vasıtanın geçebileceği yol genişlikleri farklı olduğundan göz önüne alınmıştır.

Çözüm uzayı oluşturulması işleminde siyah-beyaz görüntü üzerinde çapı verilen aracın merkezinin bulunabileceği pikseller tespit edilmeye çalışılmaktadır. Bu işlem için kullanılan yöntem Şekil II.5'de ki diyagonal alan mantığı kullanılarak gerçekleştirilmektedir. Şekil II.5 de görülen diyagonal alanın merkezindeki pikselin aracın merkezi olarak kabul edilmesinin ardından merkez pikselin etrafındaki diyagonal alan üzerindeki piksellerin engellere ait olup olmadığı tespit edilmektedir. Kullanılan yöntemin sonucu olarak hesaplama yapılan (merkezde bulunan) bir piksel sonuca göre çözüm uzayına dâhil edilmekte veya engel olarak tanımlanmaktadır.



Şekil II.5 Diyagonal engel tanımlama

Burada kontrol edilen piksel Şekil II.5 de görülen ortadaki piksel ise bu pikselin şekilde görülen uzaklıktaki piksellerin tamamının kontrolü ile engel olup olmadığı tanımlanmaktadır. Kullanılan uzaklıklar Denklem II.1 deki gibidir;

$$b = \frac{\text{araç çapı}}{2} \quad \text{ve} \quad r = \text{round}(b\sqrt{2}) \text{ dir.} \quad (\text{II.1})$$

Bu mantıkla gerçekleştirilen çözüm uzayı oluşturma işleminin formülasyonu Denklem II.2 de gösterilmektedir;

$$\sum_{i=1}^{2r} im(x-r+i, y-r+i) + \sum_{i=1}^{2r} im(x-r-i, y+r-i) + \sum_{i=1}^{2b} im(x-b+i, y) + \sum_{i=1}^{2b} im(x, y-b+i) + \sum_{i=1}^b im(x-b+i, y+i) + \sum_{i=1}^b im(x+i, y+b-i) + \sum_{i=1}^b im(x+b-i, y-i) + \sum_{i=1}^b im(x-i, y-b+i) \quad (\text{II.2})$$

Buradaki değişkenler;

$$b = \frac{\text{araç çapı}}{2}$$

$r = \text{round}(b\sqrt{2})$ yani $b\sqrt{2}$ 'nin guantalanmış değeri

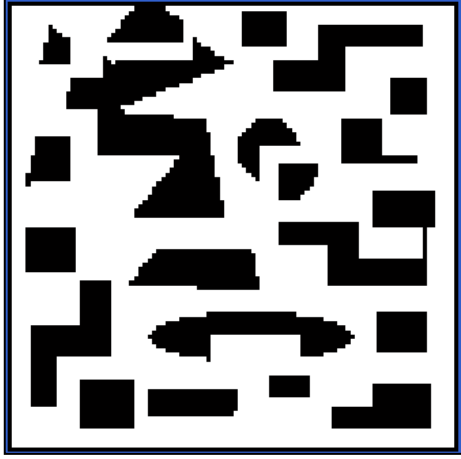
im = görüntü değişkeni

x = satir indeksi

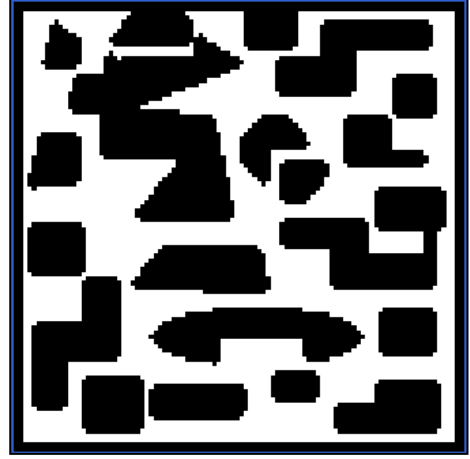
y =sütün indeksi

$im(x,y)$ = görüntü değişkenindeki pikselin değeridir.

Sonuç olarak siyah-beyaz görüntüden elde edilen çözüm uzayı görüntüsü (satır x sütun) boyutunda bir matris oluşturmaktadır. Yukarıdaki denklemler kullanılarak yapılan kontroller sonucu her bir piksel için kullanılabilir olma durumuna göre 0 (kullanılmaz) veya 1 (kullanılabilir, aracın merkezi bu noktada olabilir) değerleri çözüm uzayı görüntüsünün matrisinde piksel değerleri olarak atanmaktadır. Oluşturulan siyah-beyaz görüntü ve çözüm uzayı görüntüsü Şekil II.6 ve Şekil II.7 de gösterildiği gibidir.



Şekil II.6 Siyah beyaz görüntü



Şekil II.7 Çözüm uzayı görüntüsü

II.2.3 Çözüm Uzayının Noktalara İndirgenmesi

Bilindiği üzere iki nokta arasındaki en kısa yol bu iki noktanın birbirlerine dik olan doğrudur. İki nokta arasında bir engel olması durumunda ise bu yol engele teğet geçen bir yol şeklini alacaktır. Bu durum söz konusu engel bir daire olduğunda belirli bir yüzeyine teğet geçerken farklı şekiller için tek bir teğet noktası olabilmektedir. Bu mantıkla yola çıkılarak oluşturulan çözüm uzayının indirgenmesi ve yol için kullanılacak düğüm noktalarının elde edilmesi aşağıdaki gibi yapılmaktadır.

Teğet noktaları oluşturmak amacı ile çözüm uzayındaki cisimlerin etrafına bir piksel genişliğinde kaplama hatlar oluşturulmaktadır. Hatlar oluşturulurken çözüm uzayı üzerindeki her bir beyaz nokta için Denklem II.3 deki formülasyon uygulanmaktadır. Herhangi bir beyaz piksel için uygulanan fonksiyonun sonucunun maksimum yani 9 çıkmaması (etrafındaki tüm piksellerin engelsiz tanımlanmaması) durumunda kontrol edilen pikselin bir engele komşu olduğu belirlenmektedir. Komşuluk ilişkisi belirlenen pikseller için siyah (0) değeri geriye kalan engel nitelikli ve engellerle komşuluk ilişkisi bulunmayan cisimler için ise beyaz (1) değeri hat değişkeni ve görüntüsünde kullanılmaktadır.

$$\sum_{i=1}^3 \sum_{j=1}^3 (imc(x-2+i, y-2+j)) \quad (II.3)$$

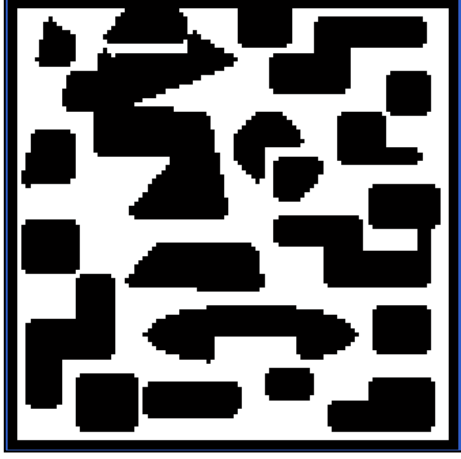
imc= çözüm görüntüsü değişkeni

x= satir indeksi

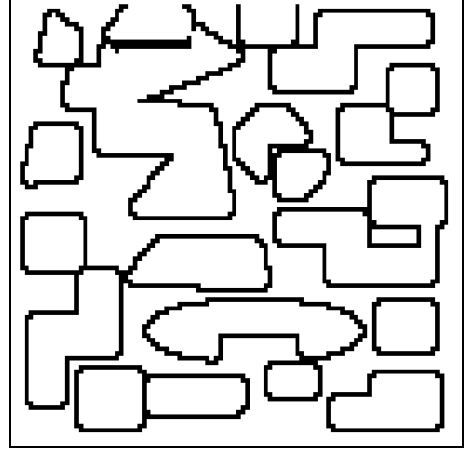
y=sütün indeksi

$imc(x,y)$ = çözüm görüntüsü değişkenindeki pikselin değeridir.

Yukarıda anlatılan mantıkla oluşturulan çözüm uzayı ve bu çözüm uzayına ait teğet görüntüleri Şekil II.8 ve II.9 da gösterildiği gibidir.

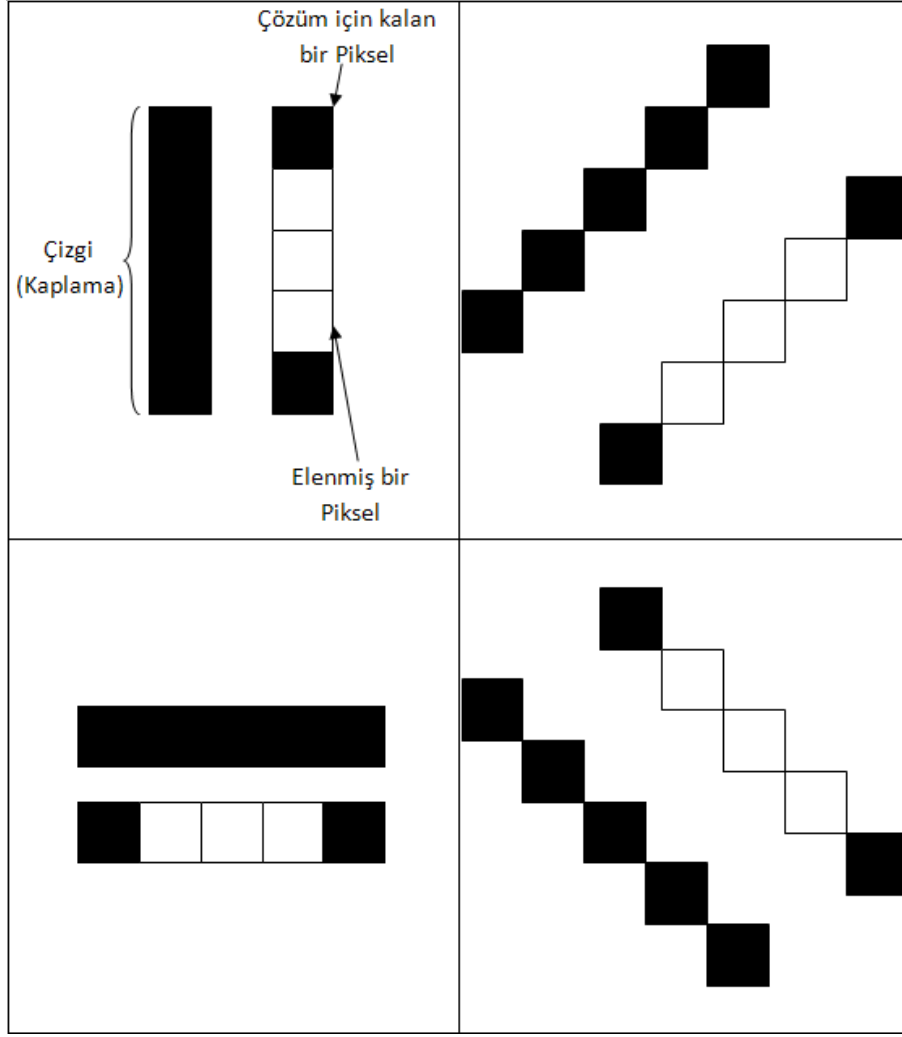


Şekil II.8 Çözüm uzayı



Şekil II.9 Teğet noktaları

İki nokta arasındaki en kısa yol, bu iki noktayı birleştiren doğru parçasıdır. Bu mantıkla düşünüldüğünde başlangıç ve bitiş düğümleri belli olan bir doğrunun üzerinde bulunan herhangi bir düğüm en kısa yolun oluşturulmasında etken değildir. Doğru üzerindeki düğüm olmadan da başlangıç ve bitiş düğümlerinin sayesinde doğru çizilebileceği için aradaki düğüm etkisiz eleman olarak görülebilir. Oluşturulan hatların piksellerden oluşan doğrular gibi davrandıkları bölgeler için bu etkisiz düğümleri bulmak ve hatlardan çıkarmak üzere işlem gerçekleştirilmiştir. MATLAB programının yapısından dolayı, 45 derece ve katlarındaki eğimlere sahip doğrular üzerinde gerçekleştirilebilen düğüm eleme işlemi ile çözüm uzayı üzerinde kullanılması muhtemel düğümler belirlenmiştir. Hatların doğrusal parçaları üzerinde gerçekleştirilen düğüm indirgeme mantığı Şekil II.10 da ki gibidir.



Şekil II.10 Çizgilerdeki düğüm sayısının azaltılması

Düğüm elemesi için kullanılan formülasyonlar Denklem II.4 deki gibidir. Denklem kaplama görüntüsünün tersi üzerinde kullanılmak üzere tasarlanmıştır. Kontrol edilen pikselin beyaz (1) olması kaydı ile formül sonucunun 3 çıkması durumunda kontrol edilen piksel siyah (0) değerlikli bir piksel haline getirilir ve sonuç olarak elde edilen görüntünün tekrar tersi alınarak nokta görüntüsü elde edilir.

$$\sum_{i=1}^3(imn(x-1+i, y-1+i)), \text{ sol alttan sağ üste çapraz bir çizgi,}$$

$$\sum_{i=1}^3(imn(x-1+i, y)), \text{ soldan sağa yatay bir çizgi,}$$

$$\sum_{i=1}^3(imn(x, y-1+i)), \text{ yukarıdan aşağıya dikey bir çizgi}$$

$$\sum_{i=1}^3(imn(x-1+i, y+1-i)), \text{ sol üstten sağ alta çapraz bir çizgi için. (II.4)}$$

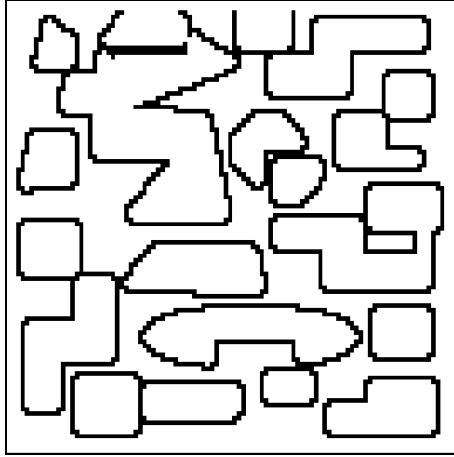
imn = kaplama görüntüsü değişkeni

x = satir indeksi

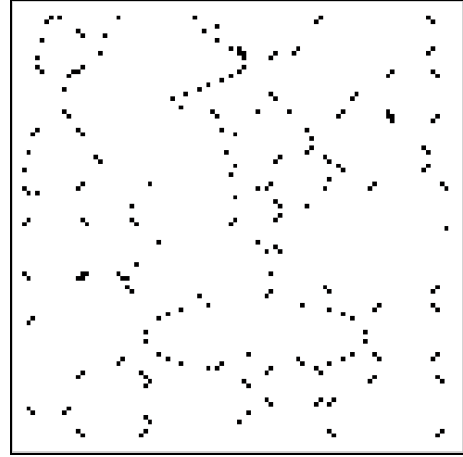
y =sütün indeksi

$imn(x,y)$ = kaplama görüntüsü değişkenindeki pikselin değeridir.

Gerçekleştirilen kaplamalar üzerindeki düğüm indirgemesi sonrası ve öncesi elde edilen sonuçlar Şekil II.11 ve Şekil II.12 de gösterildiği gibidir.



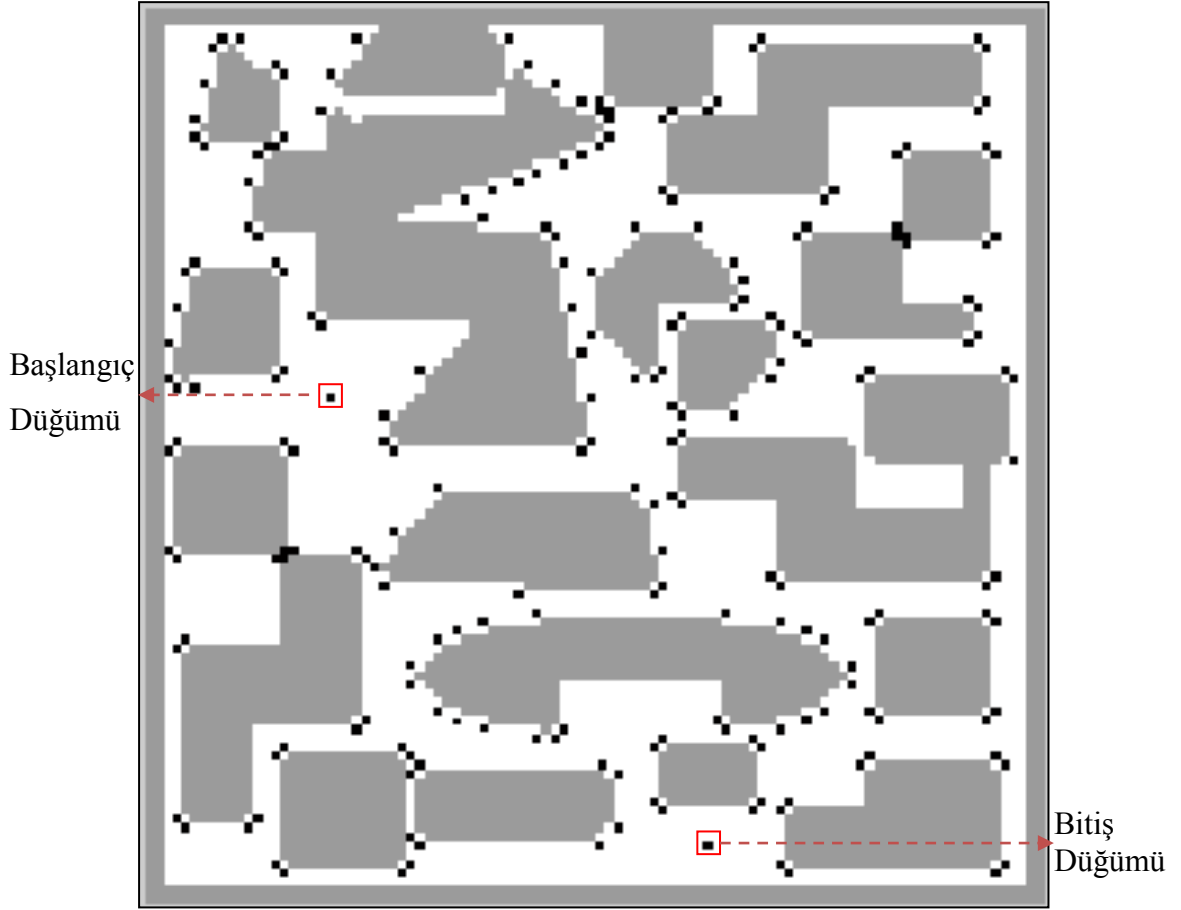
Şekil II.11 Teğet kaplamalar



Şekil II.12 İndirgenmiş düğümler

Yukarıda gösterilen işlemlerden sonra elde edilen düğümler kümesine başlangıç ve bitiş düğümlerinin de eklenmesi ile en kısa yol bulma problemine ait düğümlerin çözüm kümesi elde edilmiş bulunmaktadır.

İşlemler sonrası oluşturulan engeller, indirgenmiş çözüm düğümleri, başlangıç düğümü ve bitiş düğümünü barındıran görüntü Şekil II.13 de gösterildiği gibidir.



Şekil II.13 Genel kapsamlı görüntü

Daha sonra gerçekleştirilecek yol bulma işlemleri öncesi, düğüm çözüm kümesi üzerindeki düğümler arası görüş olanağının hesaplanması gerekmektedir. Buradaki amaç düğümler arası hareket olanağının hesaplanmasıdır.

Düğümler arası görüş olanağı, düğüm çözüm kümesi üzerindeki her ikili düğüm kombinasyonları arası oluşturulan sanal doğruların bir engelle kesişip kesişmediği tespit edilerek hesaplanmaktadır. İki düğüm arasındaki doğru üzerinde bir engel yoksa bu iki düğüm arasında doğrusal hareket gerçekleştirilebilir demektir ve bir sonraki aşamada doğrusal hareket gerçekleştirilebilen ikili düğüm kombinasyonları arası dik uzaklıklar Pisagor denklemi ile bulunmaktadır.

Düğümlerin birbirlerini görüş olanakları ve düğümler arası dik uzaklıklar gerekli değişkenlerde tutularak yol bulma işleminde kullanılmak üzere saklanmaktadır.

BÖLÜM III GENETİK ALGORİTMALAR VE KARINCA KOLONİSİ ALGORİTMASI

III.1 GİRİŞ

Bu bölümde hybrid algoritma oluşturmada kullanılan yapay zeka algoritmalarından Genetik Algoritmalar (GA) ve Karınca Kolonisi Algoritması (KKA) incelenecek, avantaj ve dezavantajları belirlenerek genel yapıları hakkında tartışma gerçekleştirilecektir.

III.2 GENETİK ALGORİTMALAR

III.2.1 Giriş ve Tanımı

Optimizasyon bir şeyin daha iyisini gerçekleştirme sürecidir. Bir mühendis veya bilim adamı yeni bir fikir ortaya koyar. Optimizasyon bu fikrin geliştirilmesine yardımcı olur. Optimizasyon; fikirleri ilerletmek için ortaya atılan bilgileri kullanarak, başlangıç kavramlarını (parametrelerini) değiştirmektir. Eğer fikri etkileyen parametreler elektronik formata dönüştürülürse bilgisayar mükemmel bir optimizasyon aracı olur. Bir takım bilgiler bilgisayara girilir ve bir çözüm elde edilir. Elde edilen bu çözümün değerlendirilmesi optimizasyon aracılığıyla gerçekleştirilir. Bir problemin birden fazla çözümü varsa, en iyi çözümü bulmak gerekir. Aslında “en iyi” tanımlaması izafi kavramdır. Optimal çözüm, problemi formülize eden kişiye bağlıdır. Eğitim, fikirler, ideolojiler ve sosyal durumlar “en iyi” tanımlamasında etkili olan faktörlerdir. Bazı problemlerin tam cevabı bulunurken, bazıları optimal noktalar olarak bilinen, değişik minimum ve maksimum noktalarına sahiptir. İşte burada “en iyi” izafi olabilir. Yer kürede yaşayan insanoğlunun hayatı, optimizasyon problemleri ve çözümleri ile doludur. Bununla ilgili birkaç örnek verilebilir. Hem işe zamanında gitmek, hem de uyku miktarını artırmak için ne zaman kalkılmalıdır? İşe giderken en iyi yol neresidir? Bir projeyi yürütürken ilk olarak nereden başlanmalıdır? Bir şeyin uzunluğunu kısaltmak, ağırlığını azaltmak bir ürünün cazibesini artırmak ve maliyetini düşürmek optimizasyon tekniğinin ilgi alanı içerisindedir[51].

Son yıllarda optimizasyonun önemi, bir çok büyük ölçekli kombinatorol optimizasyon (combinatorial optimization) problemlerinin ve yüksek kısıtlı mühendislik problemlerinin günümüz bilgisayarları ile yaklaşık olarak çözülebilmelerinden dolayı daha da artmıştır. GA'nın amacı böyle kompleks problemlerdir. Bu problemler, olasılıklı algoritmalar sınıfına ait olmakla birlikte rastgele algoritmalarından çok farklıdır[52].

GA, rastlantısal arama teknikleri kullanarak çözüm bulmaya çalışan, parametre kodlama esasına dayanan sezgisel bir arama tekniğidir[5]. GA, canlıların en iyi olanı yaşar prensibini örnek alır ve iyi bireylerin kendi yaşamlarını muhafaza edip kötü bireylerin yok olmasına dayanır.

GA, ebeveyn bireylerden oluşan yeni bireylerin şartlara uyum sağlayıp yaşamlarını devam ettirmesi prensibine sahiptir. Oluşan yeni bireylerin ebeveynlerinden iyi özellikli genleri alabilecekleri gibi kötü özellikteki genleri de alabilmeleri mümkündür. Bu durum söz konusu olunca kötü özellikteki genlere sahip bireyler yaşamlarını devam ettiremeyeceklerdir[6].

GA'nın bu özelliği probleme yeni çözümler üretmek için kopyalama, çaprazlama ve mutasyon gibi operatörlerle gerçekleştirilir. GA'nın önemli özelliklerinden birisi de bir grup üzerinde çözümü araması ve bu sayede çok sayıda çözümün içinden en iyi olanı seçmesidir [53].

GA, olasılığa dayalı çözümler sunar ve bir problemin en iyi çözümünü bulmak için garanti vermez. Ancak problemlere kısa süre içinde kabul edilebilir çözümler üretir. GA'lar problem hakkında bilgi olmadan da kullanılabilir için tercih edilmektedir. Özellikle matematiksel bir denklemin kurulamadığı problemlerde kabul edilebilir olumlu sonuçlar vermektedir. Bu durum karmaşık problemlerde zamandan ve maliyetten önemli miktarda kazanç sağlamaktadır. GA'nın özelliklerinden birisi de bir grup üzerinde çözümü araması ve bu sayede çok sayıda çözümün içinden en iyi olanı seçmesidir[54].

III.2.2 Tarihçesi

Michigan Üniversitesinde psikoloji ve bilgisayar bilimi uzmanı olan John Holland bu konuda ilk çalışmaları yapan kişidir. Mekanik öğrenme (machine learning) konusunda çalışan Holland, Darwin'in evrim kuramında etkilenecek canlılarda yaşanan genetik süreci bilgisayar ortamında gerçekleştirmeyi düşünmüştür. Tek bir mekanik yapının öğrenme yeteneğini geliştirmek yerine böyle yapılarda oluşan bir topluluğun çoğalma, çiftleşme, mutasyon, vb. genetik süreçlerden geçerek başarılı (öğrenebilen) yeni bireyler oluşturabildiğini görmüştür. Çalışmalarının sonucunu açıkladığı kitabının 1975'te yayınlanmasından sonra geliştirdiği yöntemin adı Genetik Algoritmalar (ya da kısaca GA) olarak yerleşmiştir. Ne var ki 1985 yılında Holland'ın öğrencisi olarak doktorasını veren David E. Goldberg adlı inşaat mühendisi, 1989 da, konusunda bir klasik sayılan kitabını yayınlamaya dek genetik algoritmaların pek pratik yararı olmayan bir araştırma konusu olduğu düşünülmüştür. Kitabında genetik algoritmalara dayalı tam 83 uygulamaya yer vererek GA'nin dünyanın her yerinde çeşitli konularda kullanılmakta olduğunu savunmuştur[7].

III.2.3 Temel Bileşenleri ve Yapısı

Birçok alanda uygulama imkânı ve uygulamaları olan genetik algoritmaların işlem adımları şöyle açıklanabilir[5]:

- Arama uzayındaki tüm mümkün çözümler dizi olarak kodlanır.
- Genellikle rastsal bir çözüm kümesi seçilir ve başlangıç popülasyonu olarak kabul edilir.
- Her bir dizi için bir uygunluk değeri hesaplanır, bulunan uygunluk değerleri dizilerin çözüm kalitesini gösterir.
- Bir grup dizi belirli bir olasılık değerine göre rastsal olarak seçilip çoğalma işlemi gerçekleştirilir.
- Yeni bireylerin uygunluk değerleri hesaplanarak, çaprazlama ve mutasyon işlemlerine tabi tutulur.
- Önceden belirlenen kuşak sayısı boyunca yukarıdaki işlemler devam ettirilir.

- İterasyon, belirlenen kuşak sayısına ulaşınca ya da önceden belirlenen bir başka sonlandırma kriteri sağlanınca işlem sona erdirilir. Amaç fonksiyonuna göre en uygun olan dizi seçilir.

Genetik algoritmalar bir çözüm uzayındaki her noktayı, kromozom adı verilen ikili bit dizisi ile kodlar. Her noktanın bir uygunluk değeri vardır. Tek bir nokta yerine, genetik algoritmalar bir populasyon olarak noktalar kümesini muhafaza eder. Her kuşakta, genetik algoritma, çaprazlama ve mutasyon gibi genetik operatörleri kullanarak yeni bir populasyon oluşturur. Birkaç kuşak sonunda, populasyon daha iyi uygunluk değerine sahip üyeleri içerir. Bu, Darwin'in rastsal mutasyona ve doğal seçime dayanan evrim modellerine benzemektedir. Genetik algoritmalar, çözümlerin kodlanmasını, uygunlukların hesaplanmasını, çoğalma, çaprazlama ve mutasyon operatörlerinin uygulanmasını içerir[55].

III.2.3.1 Çözümlerin Kodlanması

Bir problemin çözümü için genetik algoritma geliştirmenin ilk adımı, tüm çözümlerin aynı boyutlara sahip bitler dizisi biçiminde gösterilmesidir. Dizilerden her biri, problemin olası çözümler uzayındaki rastsal bir noktayı simgeler. Parametrelerin kodlanması, probleme özgü bilgilerin genetik algoritmanın kullanacağı şekle çevrilmesine olanak tanır[55].

III.2.3.2 İlk Popülasyonun Oluşturulması

Olası çözümlerin kodlandığı bir çözüm grubu oluşturulur. Çözüm grubu popülasyon, çözümlerin kodları da kromozom olarak adlandırılır. İkili alfabenin kullanıldığı kromozomların gösteriminde, ilk popülasyonun oluşturulması için rastsal sayı üreticileri kullanılabilir. Rastsal sayı üreticisi çağrılır ve değer 0,5'den küçükse konum 0'a değilse 1 değerine ayarlanır. Birey sayısının ve kromozom uzunluğunun az olduğu problemlerde yazı-tura ile de konum değerleri belirlenebilmektedir. Genetik algoritmalarda ikili kodlama yöntemi dışında, çözümü aranan probleme bağlı olarak farklı kodlama yöntemleri de kullanılmaktadır [55].

III.2.3.3 Uygunluk Değerinin Hesaplanması

Bir kuşak oluşturulduktan sonraki ilk adım, popülasyondaki her üyenin uygunluk değerini hesaplama adımıdır. Örneğin, bir maksimizasyon problemi için i.

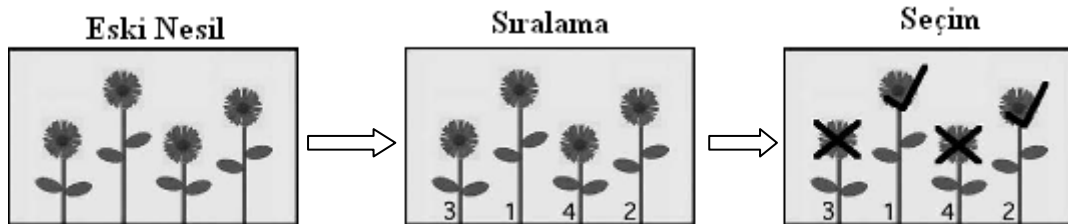
üyenin uygunluk değeri $f(i)$, genellikle o noktadaki amaç fonksiyonunun değeridir. Çözümü aranan her problem için bir uygunluk fonksiyonu mevcuttur. Verilen belirli bir kromozom için uygunluk fonksiyonu, o kromozomun temsil ettiği çözümün kullanımıyla veya yeteneğiyle orantılı olan sayısal bir uygunluk değeri verir. Bu bilgi, her kuşakta daha uygun çözümlerin seçiminde yol göstermektedir. Bir çözümün uygunluk değeri ne kadar yüksekse, yaşama ve çoğalma şansı o kadar fazladır ve bir sonraki kuşakta temsil edilme oranı da o kadar yüksektir[55].

III.2.3.4 Çoğalma İşleminin Uygulanması

Çoğalma operatöründe diziler, amaç fonksiyonuna göre kopyalanır ve iyi kalıtsal özellikleri gelecek kuşağa daha iyi aktaracak bireyler seçilir. Üreme operatörü yapay bir seçimdir. Dizileri uygunluk değerlerine göre kopyalama, daha yüksek uygunluk değerine sahip dizilerin, bir sonraki kuşaktaki bir veya daha fazla yavruya daha yüksek bir olasılıkla katkıda bulunması anlamına gelmektedir. Çoğalma, bireyleri seçme işleminden seçilmiş bireyleri bir eşleme havuzuna kopyalama işleminden ve havuzda bireyleri çiftler halinde gruplara ayırma işleminden oluşur.

Uygunluk değerinin hesaplanması adımı sonrasında mevcut kuşaktan yeni bir populasyon yaratılmalıdır. Seçim işlemi, bir sonraki kuşak için yavru üretmek amacıyla hangi ailelerin yer alması gerektiğine karar vermektedir. Bu doğal seçimdeki en uygunun yaşaması durumuna benzerdir. Bu yöntemin amacı, ortalama uygunluğun üzerindeki değerlere çoğalma fırsatı tanımadır. Bir dizinin kopyalanma şansı, uygunluk fonksiyonuyla hesaplanan dizinin uygunluk değerine bağlıdır. Seçim yöntemlerine rulet tekerleği seçimi, turnuva seçimi ve sıralama seçimi gibi seçim yöntemleri örnek verilebilir[55].

İlgili seçim işlemine ait benzetim Şekil III-1 deki gibidir[56]:



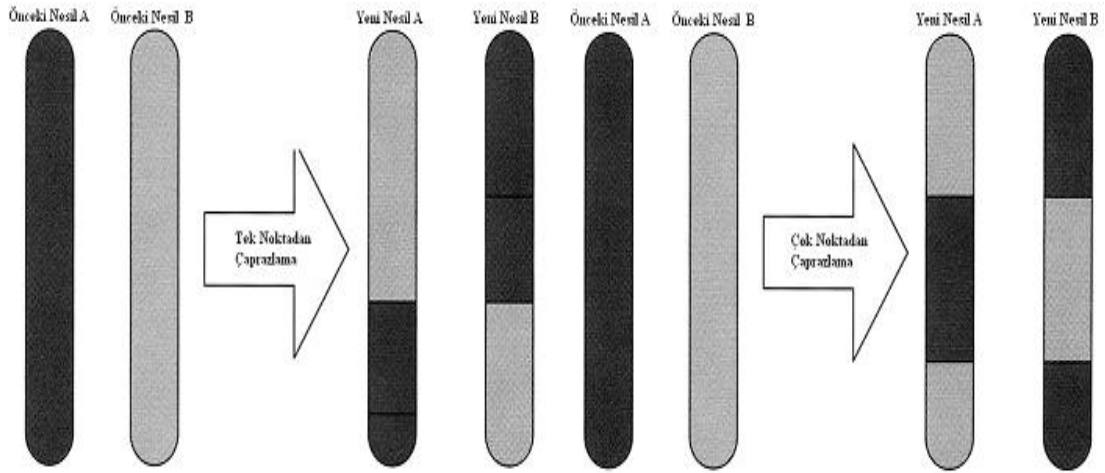
Şekil III-1 Seçim İşlemi

III.2.3.5 Çaprazlama İşleminin Uygulanması

Mevcut gen havuzunun potansiyelini arařtırmak üzere, bir önceki kuřaktan daha iyi nitelikler içeren yeni kromozomlar yaratmak amacıyla çaprazlama operatörü kullanılmaktadır. Çaprazlama genellikle, verilen bir çaprazlama oranına eşit bir olasılıkla seçilen aile çeřitlerine uygulanmaktadır.

Genetik algoritmanın performansını etkileyen önemli parametrelerden biri olan çaprazlama operatörü doğal populasyonlardaki çaprazlamaya karşılık gelmektedir. Coğalma işlemi sonucunda elde edilen yeni populasyondan rastsal olarak iki kromozom seçilmekte ve karşılıklı çaprazlama işlemine tabi tutulmaktadır. En basit çaprazlama yöntemi tek noktalı çaprazlama yöntemidir. Tek noktalı çaprazlama yapılabilmesi için her iki kromozomun da aynı gen uzunluğunda olması gerekir. İki noktalı çaprazlamada ise kromozom iki noktadan kesilir ve karşılıklı olarak pozisyonlar yer değıřtirilir[55].

İlgili çaprazlama işlemine ait benzetim Şekil III-2 deki gibidir[57]:



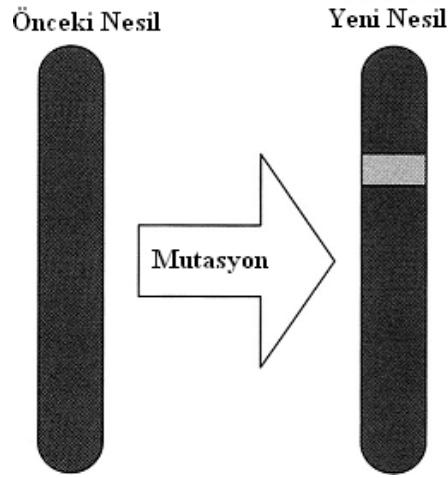
Şekil III-2 a-) Tek Noktadan Çaprazlama b-) Çok noktadan Çaprazlama

III.2.3.6 Mutasyon İşleminin Uygulanması

Çaprazlama mevcut gen potansiyellerini arařtırmak üzere kullanılır. Fakat populasyon gerekli tüm kodlanmış bilgiyi içermeyebilir ise, çaprazlama tatmin edici bir çözüm üretmez. Bundan dolayı, mevcut kromozomlardan yeni kromozomlar üretme yeteneğine sahip bir operatör gerekmektedir. Bu görevi mutasyon gerçekleştirir.

Yapay genetik sistemlerde mutasyon operatörü, bir daha elde edilemeyebilir iyi bir çözümün kaybına karşı koruma sağlamaktadır. İkili kodlama sisteminin kullanıldığı problemlerde mutasyon, düşük bir olasılık değeri altında bir bit değerini (0 veya 1 olabilir) diğer bit değerine dönüştürür. İkili kodlama sisteminin kullanılmadığı problemlerde ise daha farklı mutasyon yöntemleri kullanılmaktadır. Hangi yöntem kullanılırsa kullanılsın, mutasyonun genel amacı, genetik çeşitliliği sağlamak veya korumaktır[55].

İlgili mutasyon işlemine ait benzetim Şekil III-3 deki gibidir[57]:



Şekil III-3 Mutasyon İşlemi

III.2.3.7 Yeni Kuşağın Oluşması ve Döngünün Durdurulması

Yeni kuşak çoğalma, çaprazlama ve mutasyon işlemlerinden sonra tanımlanmakta ve bir sonraki kuşağın ebeveynleri olmaktadır. Süreç yeni kuşakla çoğalma için belirlenen uygunluk ile devam eder. Bu süreç, önceden belirlenen kuşak sayısı kadar veya bir hedefe ulaşıncaya kadar ya da başka bir durdurma kriteri sağlanana kadar devam eder. İstenen hassasiyet derecesine göre de maksimum iterasyon sayısı belirlenebilmekte ve iterasyon bu sayıya ulaştığında döngü durdurulabilmektedir. Durdurma kriteri iterasyon sayısı olabileceği gibi hedeflenen uygunluk değeri de olabilmektedir [55].

III.2.3.8 İlgili Parametre Seçimleri

Parametreler, genetik algoritma performansı üzerinde önemli etkiye sahiptir. Optimal kontrol parametreleri bulmak için bir çok çalışma yapılmıştır fakat tüm problemler için genel olarak kullanılacak parametreler bulunamamıştır. Bu parametreler, kontrol parametreleri olarak adlandırılmaktadır. Kontrol parametreleri popülasyon büyüklüğü, çaprazlama olasılığı, mutasyon olasılığı, kuşak aralığı, seçim stratejisi ve fonksiyon ölçeklemesi olarak sayılabilir. Bu parametreler aşağıda açıklanmıştır [55]:

- Popülasyon Büyüklüğü: Genetik algoritma kullanıcısı tarafından verilen en önemli kararlardan birisidir. Bu değer çok küçük olduğunda, genetik algoritma yerel bir optimuma takılabilmektedir. Popülasyonun çok büyük olması ise çözüme ulaşma zamanını arttırmaktadır. Bu konuda Goldberg 1985’de, yalnızca kromozom uzunluğuna bağlı bir popülasyon büyüklüğü hesaplama yöntemi önermiştir. Ayrıca Schaffer ve arkadaşları 1989’da çok sayıda test fonksiyonları üzerinde yaptıkları araştırmalar sonucunda, 20-30 arası bir popülasyon büyüklüğünün iyi sonuçlar verdiğini belirtmişlerdir.

- Çaprazlama Olasılığı: Çaprazlamanın amacı, mevcut iyi kromozomların özelliklerini birleştirerek daha uygun kromozomlar yaratmaktır. Kromozom çiftleri belli bir olasılık ile çaprazlamaya uğramak üzere seçilirler. Çaprazlamanın artması, yapı bloklarının artmasına neden olmakta fakat aynı zamanda bazı iyi kromozomların da bozulma olasılığını arttırmaktadır.

- Mutasyon Olasılığı: Mutasyonun amacı popülasyondaki genetik çeşitliliği korumaktır. Mutasyon bir kromozomdaki her bitte meydana gelebilir. Eğer mutasyon olasılığı artarsa, genetik arama rastsal bir aramaya dönüşür. Fakat bu aynı zamanda kayıp genetik malzemeyi tekrar bulmada yardımcı olmaktadır.

- Kuşak Aralığı: Her kuşaktaki yeni kromozom oranına kuşak aralığı denmektedir. Genetik operatörler için kaç tane kromozomun seçildiğini gösterir. Yüksek bir değer bir çok kromozomun yer değiştirdiği anlamına gelmektedir.

- Seçim Stratejisi: Eski kuşağı yenilemenin çeşitli yöntemleri mevcuttur. Kuşaksal stratejide, mevcut popülasyondaki kromozomlar tamamen yavrular ile yer değiştirir. Popülasyonun en iyi kromozomu da yenilediğinden dolayı bir sonraki

kuşğa aktarılamaz ve bu yüzden bu strateji en uygun (elitist) stratejisiyle beraber kullanılmaktadır. En uygun stratejisinde, popülasyondaki en iyi kromozomlar hiçbir zaman yenilenmemektedir bundan dolayı çoğalma için en iyi çözüm her zaman elverişlidir. Denge durumu stratejisinde ise, her kuşakta yalnızca birkaç kromozom yenilenmektedir. Genellikle, yeni kromozomlar popülasyona katıldığında en kötü kromozomlar yenilenir.

• Fonksiyon Ölçeklemesi: Doğrusal ölçekleme, üstsel ölçekleme gibi yöntemler mevcuttur. Probleme göre en uygun ölçekleme yönteminin seçilmesi genetik algoritmanın etkin işlemesi açısından önem taşımaktadır.

III.2.4 Avantajları

GA'ların avantajları aşağıdaki gibi derlenmiştir[55,58-69]:

GA'ların en büyük avantajlarından biri paralel çalışmasıdır. Bu paralellik çözüme ulaşmak için birden çok bireyin farklı bölgelerden çalışmasından kaynaklanmaktadır. Paralelliğin sonucu olarak ise GA lar çözüme daha hızlı sürede ulaşmakta ve bu durum kullanıcı için büyük avantaj teşkil etmektedir.

GA'lar geniş bir problem uzayına hitap etmektedir. Optimizasyon uygulanması gereken hemen hemen her problem için kullanılabilir olması GA'yı tercih edilen bir yapay zeka algoritması konumuna sokmuştur. Birçok algoritma belirli kıstaslarla belli problem uzayına hitap ederken GA'lar bu kıstasları, problem tipinin kullanıcı dizaynı ile belirlenmesinden dolayı minimuma indirmiştir.

GA'lar türev kullanmayan yapıdaki algoritmalarındandır. Bunun anlamı GA'lar uygunluk değeri hesaplamada kullandıkları fitness fonksiyonu sayesinde başka bir bilgiye ihtiyaç duymadan, yani türev kullanan algoritmalarda olduğu gibi matematiksel olarak problem yapısına ve işleyişine ihtiyaç duymadan çözüm üretebilmektedir.

GA'lar tasarımcısının inisiyatifinde dizayn edilen ve çözüm üreten algoritmalarındandır. Genel işlem mantığına sadık kalan tasarımcı GA'ları kullanırken bir çok yönden değişiklik yapma ve problemin çözümünde yapısal değişiklikler yapma işini çok basit bir şekilde gerçekleştirebilir.

GA'lar birçok yerel optimum noktası olan problemlerde de yüksek performans sergileyebilmektedir. Bunun nedeni GA'nın mutasyon ve çaprazlama operatörlerinin çözümün yerel optimum noktasından kurtarılmasında etkili olmasıdır. Bireyin ilgili yerel optimum noktaya takılması durumunda eğer hata toleransı istenilen bant aralığında değilse bir sonraki nesilde GA'nın operatörleri bireyi yerel optimum noktasından kurtarabilmektedir.

GA'nın bir diğer avantajı ise problem yapısı ve nedenlerinden bağımsız olmasıdır. Bir problemin yapısı ve kaynağı GA'yı ilgilendiren bir durum değildir. GA tasarımcının verdiği fitness fonksiyonu ve gen yapısı ile optimum çözümü bulmaktan başka bir işlem yapmaz.

GA'lar en karmaşık problemlerde dahi yüksek performans sergileyebilmektedir. Problemin gürültü oranı, problemin doğrusal olup olmaması veya çözüm uzayının büyüklüğü GA ile işlem yapılırken diğer algoritmalara oranla avantajlı bir ortam sağlar. Çoğu algoritma geniş çözüm uzayında arama yaparken uzun zaman harcar ve kimi zaman cevapsız kalır. Ancak GA için çözüm uzayının büyüklüğü bir kıstas olsa bile işlem yapısından dolayı tüm uzaya eşik uzaklıkta işlem yapar tabiri kullanılabilir. Bunun sağlanması için gerçekleştirilen mutasyon çaprazlama gibi işlemler GA'nın çözüm uzayının her bölgesini kullanmasını sağlamaktadır.

Bir problemin farklı bileşenlerinin olması ve her bir bileşenin ayrı bir optimum nokta için değerlendirilmesi GA'nın bir diğer avantajıdır. Bir sistemin parçalanması gerektiği düşünüldüğünde buna uygun fitness fonksiyonu ve popülasyon kodlamasının sağlanması durumunda GA problemin her bir parçası için aynı anda ayrık işlem yapacak ve sonucu bütüne göre değerlendirebilecektir. Bu durum GA'nın kompleks ve çok etkenli problemler içinde başarısını açıklayan bir durumdur.

Bir problemin giriş değerlerinin eksik veya bozuk olması durumunda dahi GA'lar istenilen hata toleransında doğru sonuç üretebilmektedir. Bunun nedeni girdi değerlerinin GA için çözüme ulaşmadaki ihtimaller havuzu olması ve diğer yapay zeka algoritmalarının kimisinde olduğu gibi çıktı değerine direkt olarak etki etmemesidir. Hatalı bir girdi olması durumunda GA sonuca o birey için

ulaşmayacaktır ancak birey çokluğu ve gelecek nesillerdeki değer farklılıkları GA'nın sonuca ulaşmasını sağlayacaktır.

GA bir problemin çözümü için gereken çok sayıda değişkeni kullanabilme özelliğine sahiptir. Bir problemin çözümü için yüz farklı değişkenin hesaba katılması gerekse bile GA doğru sonucu üretebilmekte ve diğer algoritmalar ile arasında bu noktada büyük bir farklılık doğmaktadır. Tüm probleme ait değişkenler bir bireyde toplanabilmekte ve bu bireyin uzunluğu kullanıcı ile bilgisayar kapasitesinin kısıtları ile belirlenmektedir.

III.2.5 Dezavantaj ve Kısıtlılıkları

GA'ların dezavantaj ve kısıtlılıkları aşağıdaki gibi derlenmiştir[55,58-69]:

GA'ların ürettiği sonuç her ne kadar istenilen hata toleransı bandı içerisinde olsa da istenilen değerden daha iyi bir çözüm olup olmadığı ancak kendi sonuçları ile karşılaştırılınca veya diğer optimizasyon teknikleri ile karşılaştırılınca bulunabilir. GA'ların çözüm için verdiği herhangi bir garanti söz konusu olmadığı gibi bu algoritmaların verdiği sonuçların güvenilirliği ve yeterliliği her zaman tartışmaya açık bir konudur.

GA'lar kimi zaman yerel optimum noktalarına takılıp kalabilirler. Bu durum GA'ların yapısından dolayı çok rastlanan bir durum olmamasına rağmen yine de göz ardı edilmemesi gereken bir hata olasılığıdır. İşlemin bir yerel optimum noktasında takılıp kalması belki istenilen hata toleransı bandında sonuç üretmesine engel olmayabilir fakat bu durum nihayetinde daha iyi bir sonucunda üretilmesine engel teşkil etmektedir. Bu durum ilk başlarda yerel bir optimum noktaya takılan bireylerin zamanla seçim yönteminin de iyi belirlenmemesi etkiyle gen havuzunu doldurması ve yeni nesillerin bu yerel optimum noktasına bağlı bireylerin etkisiyle bu durumdan kurtulamaması ile gerçekleşir.

GA'ların en can alıcı hataları, mutasyon, çaprazlama ve popülasyon büyüklüğü gibi parametrelere dayalı hatalardır. Bu parametrelerin iyi belirlenmesi belli kısıtlara bağlı olmamakla beraber tamamen hazırlanan fitness fonksiyonunun yapısı ile belirlenen ve kullanıcıya dayalı bir yapıdadır. Bu parametrelerin belirli aralıklar içinde tutulması hem performans zaman dengesini sağlamak için hem de GA'ların gerçekten istenilen işlemi gerçekleştirmesi için çok önemlidir.

Popülasyon sayısının doğru belirlenmemesi GA'nın yerel optimuma takılmasını veya işlem zamanında aşırı artışa neden olunmasını sağlayabilir. Popülasyon sayısının az seçilmesi yerel optimumlara takılma olasılığını arttırırken, çok büyük seçilmesi işlem zamanında gereksiz bir artışa neden olmaktadır. GA'ların popülasyon sayısına yönelik bir belirleme şu ana kadar belirlenmemiş olduğu gibi tamamen kullanıcının gözlem ve inisiyatifine kalmış bir değerdir.

Mutasyon oranının doğru belirlenmemesi GA'ların yerel optimum noktalara takılmasına neden olabileceği gibi işlem zamanında artışa da neden olabilmektedir. İlgili oranın küçük tutulması gelecek nesillerdeki bireylerin değişim oranını azaltacağı için yerel optimum noktalarda takınılmasına neden olabilmektedir. Tam tersine ilgili değer büyük tutulması ise algoritma mantığına aykırı şekilde eski nesillerdeki bireyler ile yeni nesil bireylerin genetik kodlamasının farklılaşmasında artışa ve bu nedenle de iyi sonuçların farklılaştırılarak çözüm süresinin uzamasına neden olabilmektedir. Benzer şekilde çaprazlama işleminin gerçekleştirilme oranı da aynı sonuçlara neden olacaktır.

Karmaşık bir problem için fitness fonksiyonu oluşturmak pahalı ve zor olabilmektedir. Bu gibi durumlarda başka optimizasyon tekniklerinin kullanılması gerekebilir. Oluşturulacak fitness fonksiyonunun belki de hiçbir zaman yeterli olmayacağı durumlarda bile sonuç önceden kestirilemeyeceği için boşa emek harcanmış olabilir.

III.2.6 Tartışma

GA'lar kimi kullanıcılar tarafından her derde deva olarak tasvir edilse ve kullanılsa da aslında bazı araştırmacılara göre bu yanlış bir yaklaşımdır. GA'ların da tüm diğer mevcut algoritmalar gibi kullanılması uygun olmayan alanlar ve özel problemler söz konusudur. Bazı durumlar ve problemlerde GA'lar kullanılabilir olmasına rağmen, probleme ait kullanılabilir diğer optimizasyon algoritma ve yöntemleri GA'lara göre daha iyi sonuçları aynı zaman kistaslarında verebilir. Hangi optimizasyon problemi için hangi algoritmanın kullanılacağı ise araştırılıp bulunmalı ve mümkün olduğu kadar doğrusal ve matematiğe dayalı bir optimizasyon seçeneği mevcut ise o kullanılmalıdır.

GA'ların özellikle doğru yanlış tipi problemlerde kullanımı tavsiye edilmemektedir. Bu gibi problemlerde rastgele arama algoritmaları gibi algoritmaların kullanılması GA'lara oranla daha hızlı yanıt verebilir. Benzer şekilde sonucun basit bir denklemde yerine koyularak bulunabileceği problemler GA'lar için basit kalabilmekte ve daha uygun yöntemler daha kısa sürede ve uğraşta istenilen sonucu üretebilmektedir.

GA'ların asıl hedef problem kitlesinin hiçbir çözüm tekniği (matematiksel teknik veya hata toleransı olmayan teknikler) olmayan problemler olduğu unutulmamalıdır. Bu tekniklerle çözüm söz konusu olabilecek iken GA'ların kullanılması kimi zaman avantaj sağlayacak olsa bile güvenilirlik ve sağlamlık yönünden her zaman GA'lar ikinci seçenek olarak düşünülmelidir. GA'ların kullanılması gereken problem tiplerinin genel özellikleri aşağıdaki gibi özetlenebilir:

- Probleme ait matematiksel bir modelin oluşturulmadığı problemler,
- Geleneksel optimizasyon algoritmalarının kullanılmadığı veya istenilen sonucu üretmediği problemler,
- Arama uzayının çok büyük olduğu problemler,
- Mevcut bilgiye dayalı arama uzayında çözüm için yeterli matematiksel alt yapının sağlanamadığı problemler,
- Gerçek zamanlı olmayan veya çözüme ait yolun ve verilerin dinamik olarak değişmediği problemler.

GA'ların uygulanacak probleme göre fitness fonksiyonunun zor oluşturulur durumda olması, oluşturulamıyor olması veya oluşturulan fitness fonksiyonunun istenilen işlemi gerçekleştiremeyecek kadar gürültü içeriyor olması gibi durumlar söz konusudur. Bu gibi durumlarda fitness fonksiyonunu parçalar halinde oluşturmak kullanıcı için kolaylık sağlayacağı gibi işlemin sonucunda da belirli bir iyileştirme gerçekleştirecektir. Oluşturulacak parçalı fitness fonksiyonunun her biri için ayrı değerlendirme ve hata toleransı oluşturulması problemin çözümü için esneklik ve başarı sağlayacaktır. Bu gibi benzer bir durum için fitness fonksiyonunun yerine aday yani benzer bir fitness fonksiyonu atanması da söz konusu olabilir. Böyle bir durumda başarı durumu tartışılabilir olacağı gibi kullanıcı açısından aynı sonucun üretilebilmesi durumunda zaman ve maliyetten kazanç sağlanacaktır.

GA'ların kaçınılan bir durumu olan yerel optimum noktalara takılması ihtimali birkaç farklı yöntem ile düzeltilebilmektedir. Bu yöntemlerin içinde daha öncede bahsedilen mutasyon, çaprazlama gibi operatörlerin iyi belirlenmesinin yanı sıra fitness fonksiyonunun değiştirilmesi ve seçim işleminde penaltı durumunun kullanılması gibi seçenekleri de barındırmaktadır. Seçim işleminde penaltı durumunun uygulanması şöyle özetlenebilir: GA' işlem basamağında eski bireylerden yeni bireylerin oluşturulmasından sonra, eski bireylerin tamamen gen havuzundan çıkartılmasının yanı sıra baskın olan bir gen varsa bu genin baskınlığının azaltılmasıdır. Bu sayede gen havuzunun yerel optimum noktalarda takılıp kalması yeni bireylerin çoğunluğunun sağlanması yolu ile sağlanmaktadır.

GA'ların kullanılmasının uygun görülmediği bir alan olan dinamik problemlerde, GA'ların başarılı olabilmesi amacı ile yöntemler geliştirilmiştir. Bu yöntemler yerel optimum noktaya takılmayı önleyici önlemlerde olduğu gibi mutasyon oranının artırılması ve eski bireylerin havuzdan tamamen çıkarılmasını içermektedir. Mutasyon oranının artırılması çözüm kalitesini düşürme ihtimali olan bir etken olmakla birlikte istenilen eksikliği gidermekte etkili olarak görülmektedir. Adı geçen bir diğer yöntem olan eski bireylerin gen havuzundan tamamen çıkarılması ise GA'ların mantık yapısına ayrı düşmekle birlikte iyi olan yaşar prensibine aykırı olarak gerçekleştirilmektedir. Ne var ki bu işleminde istenilen sonuç için gerekli olduğu varsayılmaktadır.

Şu ana kadarki bilgiler ve tartışma ile bir değerlendirme yapıldığında GA'ların aslında tüm optimizasyon problemlerinde kullanılabilir olduğu ancak kullanımının ne kadar yararlı ve karlı olduğu tartışılabilir bir durum olarak dikkat çekmektedir.

Bu algoritmanın güncel hayatta yüksek kullanımına neden olan en önemli etken karmaşık ve zor problemlerde kullanıcı tarafından diğer optimizasyon yöntemlerine oranla daha basit ve az işlemle istenileni elde edebilmesidir. Ancak unutulmamalıdır ki GA'ların verdiği sonucun netliği ve doğruluğu tartışılabilir değerlerdedir ve bu durum klasik optimizasyon yöntemlerinin yanında hiçte yadırganmayacak bir eksiklik olarak değerlendirilmektedir. Yaygın olan bir görüş ve değerlendirmeye göre bir sistemin veya problemin matematik modeli oluşturulabiliyorsa çözüm için ilk yöntem klasik optimizasyon teknikleri olmalıdır.

III.3 KARINCA KOLONİSİ ALGORİTMASI

III.3.1 Giriş ve Tanımı

Günümüzde gerek robotlar için olsun gerekse nakliye ve tasarruf gibi maliyet üzerine ihtiyaçlardan dolayı olsun minimum yol bulma algoritmaları önem kazanmıştır. Bu algoritmalar oluşturulurken kimi zaman matematiksel denklemler üzerinden yola çıkılmakta kimi zaman ise doğal olaylardan esinlenerek yeni yöntemler ortaya atılmaktadır.

Son yıllarda çok revaçta olan ve üzerine birçok uygulama yapılarak geliştirilen Karınca Kolonisi Algoritması (KKA) ve türevleri de bunlardandır.

KKA algoritmaları tamamen gerçek karıncaların yiyecek bulma yöntemlerinden esinlenerek geliştirilmiş bir yapıya sahiptir. Bu yapıya göre gerçek karıncaların yiyecek ararken ve bulduktan sonra yuvaları ile yiyecek arasındaki yola bıraktıkları feromon adı verilen kimyasal madde diğer karıncalar tarafından da takip edilebilmektedir. Feromon adı verilen bu maddenin uçucu bir madde olması ve yoldaki miktarının zaman ile uçuculuğundan dolayı azaldığı düşünüldüğünde karıncaların en kısa yoldan yuvalarına yiyecek götürme mantığı kopyalanabilir olmuştur.

III.3.2 Tarihçesi

Gerçek karıncalar ile ilgili ilk deneyler 1989 yılında Goss ve arkadaşları tarafından laboratuvar ortamında gerçekleştirilmiştir[70].

Doğal hayattaki karıncaların davranışlarının ve yiyecek bulma yöntemlerinin, başta en kısa yol bulma problemleri olmak üzere, birçok problem üzerine uygulanabileceği Dorigo ve arkadaşları tarafından 1991 yılında ortaya atılmıştır[14].

Dorigo tarafından geliştirilen bu algoritma Ant System adını almıştır ve küçük ölçekli (75 şehirden az) gezgin satıcı problemleri (TSP) üzerinde uygulanarak başarılı sonuçlar elde etmiştir[7].

Algoritmanın gelişim aşamalarında yine Dorigo ve arkadaşlarının etkisi vardır. 1995,1996 ve 1997 yıllarında Dorigo ve arkadaşları geliştirdikleri algoritma ile daha iyi performans eldesi sağlamışlardır[15-17,71].

III.3.3 Temel Bileşenleri ve Yapısı

Gerçek karıncalar yuvaları ile besin kaynağı arasında gerçekleştirdikleri yiyecek taşıma işlemi sırasında bilinçsiz olarak en kısa yolu oluşturma ve kullanma durumundadırlar. Bu durum daha öncede bahsedildiği gibi karıncaların salgıladığı ve izini takip ettiği feromon adlı maddeden kaynaklanmaktadır

Karıncalar diğer sürü bireylerinin takip ederken öncelikle feromon oranı yüksek olan yolları tercih etmektedirler. Feromon gibi uçucu bir maddenin takip edilmesi sırasında, bir yolun diğerlerine göre kısa olması o yoldaki feromon miktarını etkilemektedir. Bunun başlıca nedenlerinden biri yolun uzunluğu ile birim zamanda o yolu takip eden karınca sayısının ters orantılı olması ve bundan dolayı da o yol üzerinde tekrar salgılanan feromon miktarının yüksek olmasıdır. Yani bir yol ne kadar kısa ise o yola bırakılan feromon miktarı o kadar fazla olacaktır. Diğer bir neden ise yine yol uzunluğu ile bağlantılı olarak birim zamanda buharlaşan feromon miktarıdır. Bir yolun kullanımı yol uzunluğu ile ters orantılı olduğuna göre yoldaki tazelenen feromon miktarı da yol uzunluğu ters orantılı olacaktır ve karıncalar yüksek yoğunluklu feromon miktarını takip edecektir.

Karıncaların sürü zekası adı verilen etkileşimleri ile gerçekleştirdikleri bu işlem yola bir engel eklenmesi durumunda da geçerli olmakta ve karıncalar ilk başta en kısa yolu tercih etmese dahi daha sonra en kısa yola adapte olmaktadır.

Gerçek karıncaların özellikleri şu şekilde özetlenebilmektedir:

- Karıncalar arasında feromon aracılığı ile kurulan iletişim,
- Feromon miktarının fazla olduğu yolların öncelikle tercih edilmesi,
- Kısa yollar üzerinde feromon miktarının daha hızlı artması.

Gerçek karıncalardan alınan bu özelliklerin yanı sıra yapay karıncalara eklenen özellikler de aşağıdaki gibi özetlenebilir:

- Yapay karıncalar zamanın ayrık olarak hesaplandığı bir ortamda yaşarlar,
- Yapay karıncalar tamamen kör olmayıp, problem ile ilgili detaylara erişebilirler,

- Yapay karıncalar belli bir miktar hafıza ile problemin çözümü için oluşturdukları bilgileri tutabilirler.

III.3.3.1 İlk Değerlerin Oluşturulması

KKA'da işlemlerin gerçekleştirilmesi için iki temel yapının önceden oluşturulması gerekmektedir. Bunlar feromon matrisi ve sezgisel matris adı verilen, yolların feromon izlerinden bağımsız olarak sadece uzunlukları ile ters orantılı olan ağırlık değerlerinin tutulduğu matrislerdir.

Feromon matrisi karıncaların her tur sonunda gerçekleştirdikleri işlemler sonucu feromon güncellemelerinin sonuçların tutulduğu yani yollarda kalan feromon miktarlarının tutulduğu matristir. Bu matris karıncaların yol kullanımları, feromon buharlaşma işlemi ve bazı algoritmalarda ise global feromon güncellemesinden sonra güncellenen feromon miktarlarının saklandığı yerdir.

Sezgisel matrisin değeri işlemler boyunca hiç değişmemektedir. Yollara ait uzunlukların ters orantılı değerlerini barındıran bu matris, karıncaların feromon izinden bağımsız olarak yol seçme işlemi için veya hem feromon izine hem de uzunluklara bağlı olarak gerçekleştirilecek yol seçme işlemi için kullanılır. Sezgisel matrisin kullanımındaki temel amaç karıncaların en iyi yolu bulurken tamamen feromon değerlerine bağlı kalmamaları ve olası yerel minimum çözümden uzaklaşmalarını sağlamaktır.

Bir düğümden bir diğer düğüme geçecek karıncanın yol seçimindeki olasılık bu matrislerin bir veya ikisinin kullanımı ile belirlenmektedir.

III.3.3.2 Başlangıç Durumlarının Belirlenmesi

Algoritmada ilk olarak sistemde kaç adet karıncanın olacağını belirlemek gerekmektedir. Bu değer için kesin bir kıstas olmasa da Dorigo ve arkadaşları karınca sayısının belirlenmesinde kullanılacak düğüm sayısına eşit sayıda karınca kullanılmasının algoritmanın performansı için iyi olacağını bildirmişlerdir[72].

Algoritmada ikinci olarak feromon matrisinin başlangıç değerlerinin atanması gerekmektedir. Bu değer için Dorigo ve arkadaşları tüm düğümler arası değerler için aynı değeri önermişlerdir. Bu değer sıfırdan farklı bir değer olmak koşulu ile önerilen

değer karınca sayısının en yakın komşuluk değerlerinin hesaplanmasıyla oluşturulan tur değerine bölümü ile bulunabilir[72].

Bir diğer belirlenmesi gereken parametre ise buharlaşma katsayısıdır. Bu değer çok sayıda karıncanın tercih ettiği herhangi bir yoldaki feromon miktarının dominant bir değere gelerek yerel çözümlere takınılmasını engellemek amacı ile kullanılmaktadır. Buharlaşma katsayısı Dorigo ve arkadaşları tarafından 0.5 olarak tavsiye edilmiştir[72].

Belirlenmesi gereken diğer iki önemli parametre ise alfa (α) ve beta (β) değerleridir. Bu değerlerden alfa değeri feromon matrisinin olasılık karar denklemindeki değerini belirlemektedir. Bu değer yüksek veya düşük olması yolların feromon miktarına göre seçilme ihtimalinin yüksek veya düşük olmasını belirlemektedir. Diğer değer olan beta değeri ise sezgisel matrisin yani yol uzunluklarının olasılık denklemindeki değerini belirlemektedir. Aynı şekilde bu değerinde yüksek veya düşük olması yol seçim işleminde yol uzunluklarının etkisinin yüksek veya düşük olmasının belirlenmesini etkilemektedir. Bu değerler için genel olan kanı alfa değeri için bir değerinin, beta değeri için ise iki ila beş arası bir değer kullanımının gerektiğidir[73].

İlk değerlerin atamasının gerçekleştirilmesinden sonra sanal karıncaların hangi şehirlerden yola çıkacağı belirlenmesi gerekmektedir. Karıncaların başlangıç şehirleri yani düğümleri için Dorigo ve arkadaşları rastgele düğümleri önermektedir[72].

III.3.3.3 Karınca Turlarının Gerçekleştirilmesi

Karıncaların şehirlere yerleştirilmesinin ardından karıncaların turlarına başlaması aşamasına geçilir. Bu aşamada karıncaların buldukları şehirlerden sonraki seçecekleri şehir olasılık karar denklemi ve Rulet çemberi adı verilen bir seçim yöntemi vasıtası ile belirlenmektedir.

Seçim işlemi tüm karıncalar için aynı anda gerçekleştirilir ve tüm karıncalar aynı anda hareket eder. Gidecekleri şehri seçen karıncalar aynı anda hareket ederek turlarını da aynı anda tamamlamış olurlar[73].

III.3.3.4 Feromon Düzeylerinin Güncellenmesi

Feromon miktarlarının güncellenmesi işlemi birçok farklı algoritma yapısında farklı kullanılmaktadır. Bu işlemin ana hedefinin ve yapısının tüm algoritmalarda ortak olduğu düşünüldüğünde işlem iki temel aşamada özetlenebilir; feromon eklenmesi ve buharlaştırılması.

- Buharlaştırma:
Buharlaştırma işlemi, bir sabiti bulunan ve bu sabit oranında tur sonlarında yollardaki feromon miktarlarının azaltılmasını sağlayan bir işlemdir. Tüm karınca tabanlı algoritma yapılarında gerçekleştirilen bu işlem yerel optimum çözümlerden kaçınmak için kullanılmaktadır.
- Ekleme:
Buharlaştırma işlemi sonrası karıncaların yeni turlara başlamasından önce feromon ekleme işlemi gerçekleştirilir. Bu işlemde kimi algoritmalarda yerel ve global olmak üzere iki farklı feromon güncellemesi gerçekleştirilirken kimi algoritmalarda ise karıncaların tur uzunluklarına bağlı olarak tek bir güncelleme gerçekleştirilir[74].

Feromon güncellemesi işleminde birçok algoritmada birçok farklı yapı göze çarpmaktadır. Bunlar bir önceki algoritmanın eksikliklerini gidermeye veya performansını arttırmaya yönelik ufak farklılıklar içermektedir. Genel yapıya bakıldığında feromon buharlaştırma ve eklenme yöntemleri farklılık gösterse de temel amaç aynıdır. Bu amaç karıncaların yol seçim işleminde tercih edecekleri bir sonraki düğüm için feromon miktarının etkisinin daha etkin ve doğru kılınmasıdır.

III.3.4 Avantajları

KKA'nın avantajları aşağıdaki gibi derlenmiştir[75-79]:

KKA'nın en önemli avantajlarından biri paralel çalışmasıdır. Bu paralellik GA'lar da olduğu gibi fonksiyonel bir paralellik değil tamamen yapısından kaynaklanan bir paralelliktir. Paralelliğin nedeni olarak ise birden çok karıncanın aynı anda çalışması, aynı anda karınca sayısı kadar tur gerçekleştirilmesi düşünülebilir.

Feromon sayesinde gerçekleşen pozitif geri beslemenin işlem zamanı ve sonuç kalitesi üzerinde performans artışı sağlaması yine önemli avantajlardandır. Genel optimizasyon problemlerinden de bilindiği gibi pozitif geri besleme çözüm kalitesi üzerinde oldukça yararlı gösterilen ve kullanılan bir yöntemdir. Karıncaların yola bıraktıkları feromon maddesinin diğer turlarda pozitif geri besleme olarak değerlendirilmesi bu etkiyi KKA algoritmalarında sağlamaktadır.

KKA gezgin satıcı problemi ve türevlerinde olduğu gibi bizim örneğimizin bir türevi olan araç rotalama problemlerinde de oldukça yüksek performans göstermektedir. Algoritmanın keşfi üzerine düşünüldüğünde de zaten karıncaların bu işi mükemmel yaptığı ve bu işlemin bilgisayara aktarılmasının benzer problemler üzerinde de güzel sonuçlar vermesi yadırganmamaktadır.

KKA dinamik problemler üzerinde de kullanılabilir bir yapıya sahiptir. Karıncalar yollarının önüne bir engel konulduğunda, yollarını kısa süreliğine daha uzun bir yola çevirmelerine rağmen, bir süre sonra tekrar en kısa yola adapte olmaktadır. Bu durum algoritmadaki feromon güncellemesinin etkisinden dolayı sanal karıncalar içinde gerçekleştirilebilir durumdadır.

Algoritmanın en göze batan ve takdir gören taraflarından birisi de sürü mantığını etkin bir şekilde kullanabilmesidir. Bu mantık sayesinde, sürü içinde oluşabilecek bireysel hatalar, sürünün diğer bireyleri tarafından giderilebilmektedir. Bir karıncanın uzun bir yol seçmesi, genel çözüm kalitesi üzerinde, diğer karıncaların iyi turlarından dolayı daha az yanılısamaya yol açacaktır. Geniş bir popülasyona sahip sürüler içinde hataların giderilmesi veya etkisinin azalması düşük popülasyonlu sürü ve gruplara göre daha olasıdır.

KKA kaynağın ve hedefin belirli olduğu ve önceden tanımlandığı problem tipleri üzerinde daha başarılıdır. Tüm olasılıkların hesaplanmasa da ortada olduğu (TSP problemlerindeki şehirler gibi) ve amacın belli olduğu (TSP problemlerindeki en kısa tur istemi gibi) problemler kaynağın ve hedefin belli olduğu problemler olduğu için KKA yüksek performans sergileyebilmektedir.

III.3.5 Dezavantajları ve Kısıtlılıkları

KKA'nın dezavantaj ve kısıtlılıkları aşağıdaki gibi derlenmiştir[75-79]:

Teorik analizin zorluğu KKA'nın dezavantajlarından biridir. Bir problem için teorik olarak altyapıyı oluşturmak, gerekli parametre ve değerleri atamak problemin boyutu büyüdükçe zorlaşmaktadır. Sezgisel matrisin değerlerinin atanması, karınca sayısı, olasılık katsayıları gibi belirlenmesi gereken değerler teorik olarak belirlenmesi zorluk çıkaran değişkenlerdir.

Oluşturulan yolların, düğümler arası komşulukların rastgele dizilerinden oluşması ve bu dizilerin belli bir yakınlıklarının olmaması, algoritmanın dezavantajlarından birisidir. GA'lar da ki gibi dizinin belli bir bölümü belli bir düğüm veya çözüm elemanını temsil etmemektedir. Oluşturulan diziler her bir karınca için farklı ve rastlantısal bir düğümün seçimine ait bilgi içermektedir.

İterasyon sayısı olasılık dağılımının üzerinde yüksek etkiye sahiptir. Yüksek iterasyon sayıları bir yandan yollardaki feromon miktarları üzerinde daha kalıcı etkiler bırakılmasına yol açarak olumlu bir etki gösterebilecekken diğer yandan da hem zaman hem de hatalı yol seçimi durumunda çözüm kalitesi üzerinde olumsuz etki sağlayabilir.

KKA'da araştırmalar teorik den çok deneysel olmak zorundadır. Katsayılar, karınca sayıları ve ilk matris değişkenlerinin belirlenmesinde belli bir matematiksel denklem olmamasından dolayı teorik hazırlıktan çok deneysel hazırlık yapılmaktadır. Algoritmanın performansı deneye yanıla arttırılabilecek bir yapıdadır denilmesi daha uygun olur.

Bir diğer dezavantaj ise çözüm için geçmesi gereken zamanın ya da iterasyon sayısının belirsiz olmasıdır.

III.3.6 Tartışma

KKA da tüm diğer sezgisel algoritmalarda olduğu gibi öncelikle sonucun kesin olmadığı ve belli bir yakınlık ile doğruluğun kabul edildiği algoritmalarındadır. Algoritma matematiksel denklemlerin yüksek çözüm süresi sunduğu veya çözüm sunmadığı problemler için başarılı denebilecek çözümler sunmaktadır.

KKA'ya ait parametrelerin hiçbir belirliliğinin olmaması tamamen Dorigo ve bazı araştırmacıların deneyerek bulduğu ve problem yapısına göre değişebilecek durumda olması algoritmanın performansı açısından bir eksi olarak görülebilir. Tıpkı GA'da olduğu gibi KKA'da da parametrelerin yanlış ya da problemin çözümüne uygun değerlerden uzak seçilmesi durumunda algoritmanın işlevselliği düşecektir.

Algoritma feromon üzerine kurulu gibi görülse de aslında tavsiye edilen katsayılarla kullanılması durumunda bunun tam tersi bir durum söz konusudur. Yol seçme olasılık denkleminde feromon katsayısının 1 ve olasılığa bağlı yani en yakın düğüm seçimine bağlı katsayının değerinin ise 2 ila 5 arasında seçilmesi durumunda algoritma feromon ve karınca davranışlarından çok en yakın kenar ilişkisi ile çalışıyor olacaktır. Bu durum, Dorigo ve arkadaşlarının daha sonraki karınca algoritmalarında tamamen feromon miktarına yönelik bir seçimin sağlanması ile ortadan kaldırılmıştır[79].

Benzer şekilde algortmada tavsiye edilen buharlaşma katsayısı 0.5 ve tüm yollar için eşittir. Burada gerçekleştirilen işlemin tüm yollar için ayrıcalıksız olması ve her tur sonunda gerçekleşmesi bu işlemin gereksiz bir işlem olduğuna dair izlenimler uyandırabilir.

KKA'nın birçok türevi araştırmacılar tarafından oluşturulmuş ve kullanılmıştır. Geliştirilen bu algoritmalarda söz konusu kısıtlılıklar giderilmeye çalışılmış ve çoğu zamanda başarılı olmuştur. Ancak algoritmanın temel parametrelerine ait değerler hala deneysel olarak bulunmakta ve bu tüm problem tipleri ve boyutları için değişiklik göstermektedir. Bu durum sezgisel algoritmaların işleyişinin rastgele çözümler üzerinden hareket etmesi ve rastgele çözümlerin herhangi bir matematiksel denklemlle bağıntısının kurulamamasından kaynaklanmaktadır.

İlk bulunan algoritmanın (Ant system) 75 ve üstü düğüm sayısında başarısız sonuçlar verdiği belirtilmiştir. Daha sonra geliştirilen yapılar bu sayıyı biraz daha yukarılara taşısa da KKA algoritmalar hala yüksek ölçekli problemler için yeterli durumda değildir.

BÖLÜM IV HYBRİD ALGORİTMA VE UYGULAMA

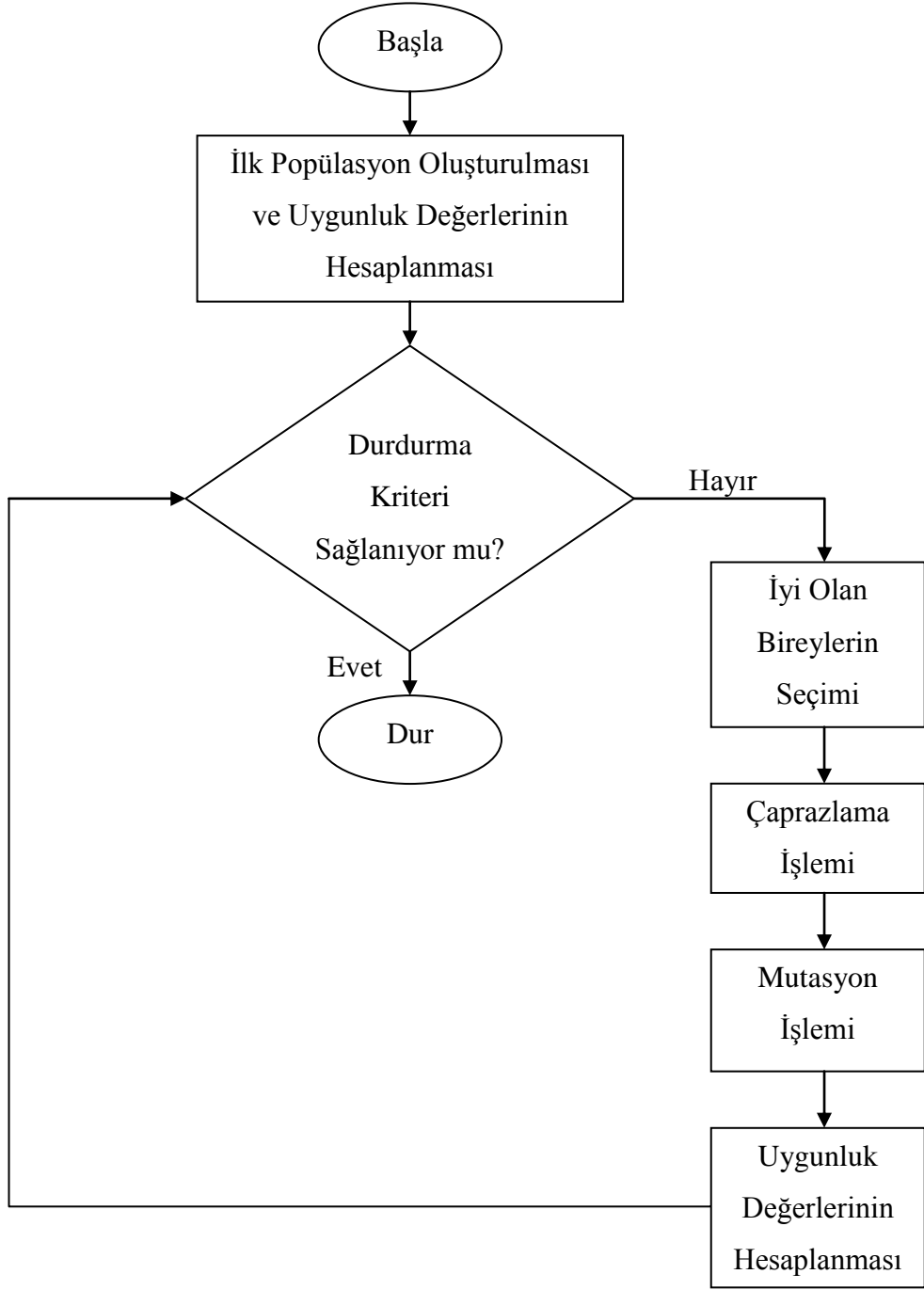
IV.1 GİRİŞ

Engellerin bulunduğu ortamda en kısa yol bulma problemine yönelik gerçekleştirilen hybrid algoritma, genetik algoritmaların ve karınca kolonisi algoritmasının belli başlı bazı prensiplerini alarak geliştirilmiş bir algoritmadır. Bundan dolayıdır ki bu bölümde algoritmanın tanıtımı ve işleyişi anlatılırken, sözü geçen algoritmaların hangi parçalarının kullanıldığı ve nedenler irdelenecektir.

Hybrid algoritma oluşturulurken kullanılan algoritma kısımlarının irdelenmesinden sonra algoritmanın akış şeması ile gösterimi yapılarak problem üzerindeki uygulamasının aktarımı sağlanacaktır.

IV.2 GENETİK ALGORİTMALARIN HYBRİD ALGORİTMADAKİ ETKİSİ

GA teorik olarak ele alındığında, iyi olan yaşar mantığı, mutasyon ile çeşitlilik sağlanması ve çaprazlama ile iyiden iyi birey üretimi gibi temel hususlar göze çarpmaktadır. Hybrid algoritma tasarlanırken bu algoritma aşamalarının her birinin elimizdeki problem tipi için uygunluğu belirlenerek ve uygun ise algoritmaya sağlayacağı katkı belirlenerek işlem yapılmıştır. GA'lara ait akış diyagramı kaba hatları ile Şekil IV.1 de görülmektedir.



Şekil IV.1 GA akış diyagramı

Yukarıdaki diyagramdan da görülebileceği gibi GA'ların temel mantığı üç işlem üzerinde gelişmektedir. Bu işlemlerin engellerin bulunduğu bir ortamdaki en kısa yol problemi üzerinde ki uygulanabilirliği değerlendirilmiştir. Değerlendirmenin sonunda hybrid algoritma içinde GA'nın hangi kısımlarının olacağı belirlenmiştir.

Değerlendirme yapılırken her bir bölüme ait göz önünde tutulan kısıtlılık ve avantajlar şöyledir:

İyi olan bireylerin seçimi: GA'lardaki iyi olan bireylerin gelecek nesillere aktarılması işlemi mevcut problem üzerinde de gerçekleştirilebilir bulunmuştur. Bunun nedeni bu işlemin aynı GA'larda olduğu gibi hybrid algoritma üzerinde de kuşaktan kuşağa daha iyi bireylerin yetişmesini sağlıyor olmasıdır.

Oluşturulan yolların iyilerinin seçilmesi ve bu yolların gelecek nesillere aktarılması bir nevi örnek teşkil etmesi hybrid algoritmanın performansında olumlu etki yaratmaktadır.

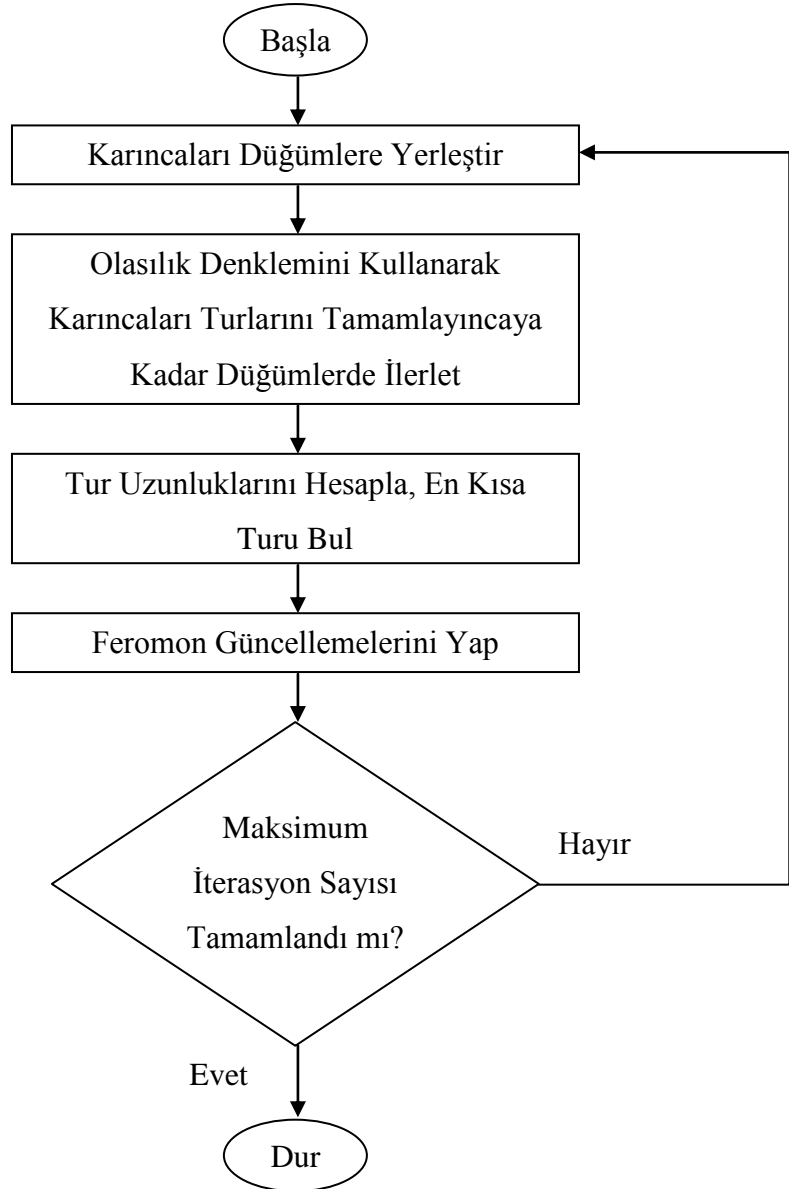
Çaprazlama işlemi: Çaprazlama işlemi GA'larda iyi olan bireylerin genlerinden yeni nesillerin üretimi için kullanılsa da oluşturulan hybrid algortmada kullanılmamaktadır. Bunun nedeni işlemin problem tipine uygun olmamasıdır. Ele alınan problemin engellerin bulunduğu bir ortamın üzerinde işleniyor olması olası tüm geçiş noktalarının (düğüm) kullanılabilirliğini kısıtlamaktadır. Bundan dolayıdır ki tüm düğümlerin kullanılmadığı bir ortamda yol uzunluğu için kullanılan düğüm sayısı yoldan yola değişiklik göstermektedir. Düğüm sayısı farklı olan yollar GA yapısı üzerinde düşünüldüğünde çaprazlama yapılarak çoğaltılamayacağı için bu işlem hybrid algortmada kullanılmamaktadır.

Ancak bu işlemin özü olan iyi olan bireylerin genlerinden yeni bireyler üretilmesi mantığı hybrid algoritma üzerinde farklı bir yöntem ile aktarılmıştır. Yolların bireyler, kullanılan düğümlerin ise genler olduğu düşünülecek olur ise; iyi olan yolların seçim işleminden sonra bu yollara ait düğümlerden yeni yollar üretimi sağlanarak çaprazlama benzeri bir mantık hybrid algortmaya aktarılmıştır.

Mutasyon işlemi: Mutasyon işlemi GA'larda çeşitlilik yaratmak amacı ile kullanılmaktadır. Yukarıda da bahsedildiği gibi problem tipinde tüm düğümlerin kullanılmıyor olması herhangi bir düğüm üzerinde değişiklik yapma olasılığını ortadan kaldırmaktadır. Ayrıca yine bireylerin yollar olduğunu düşünürsek ve bu yolların herhangi bir düğümünde tek düğümlük bir değişiklik yaptığımızı düşünürsek yolda kopukluklar ve bozukluklar oluşacaktır. Bu nedenle bu işlemin hybrid algortma üzerinde kullanımı olanaksızdır.

IV.3 KARINCA KOLONİSİ ALGORİTMASININ HYBRİD ALGORİTMADAKİ ETKİSİ

KKA'nın teorik yapısında göze batan noktalar sanal karıncaların feromon izini takip etmesi ve sürü mantığının aktif şekilde kullanılması olarak gösterilebilir. GA'nın kısımlarının değerlendirilerek işleme alınması gibi KKA'nın da kısımları değerlendirilmiş ve uygun görülen kısımları hybrid algoritma yapısına dahil edilmiştir. KKA'nın ana hatları ile akış diyagramı Şekil IV.2 de gösterildiği gibidir.



Şekil IV.2 KKA akış diyagramı

Yukarıdaki akış diyagramındaki işlem basamakları değerlendirilerek hybrid algoritma içinde kullanılacak kısımlar belirlenmiştir. KKA'ların temel işlem basamaklarından hangilerinin ilgili problemde uygulanabilir olduğu ve nedenleri şu şekildedir:

Karıncaların düğümlere yerleştirilmesi: KKA'da karıncalar tüm düğümlere yerleştirilerek her birinin kendi turlarını yapması sağlanmaktadır. Bu işlemin engellerin bulunduğu bir ortamda gerçekleştirilmesi gereksizdir çünkü başlangıç ve bitiş düğümlerinin belli iki düğüm olduğu bir ortamda daha öncede söz edildiği gibi tüm düğümler kullanılmamaktadır. Tüm düğümlerin kullanılmadığı bir ortamda kullanılmama ihtimali olan düğümlere karınca yerleştirilmesi işlem performansını olumsuz yönde etkileyecektir. Bundan dolayıdır ki hybrid algoritmada karıncalar tüm düğümlere yerleştirilmek yerine sadece başlangıç düğümüne yerleştirilmiş ve bu düğümden yola çıkarak yollar oluşturmaları sağlanmıştır.

Olasılık denklemi ile düğüm seçimi: Karıncalar düğümler üzerinde ilerlerken düğüm seçimi olasılık denklemi ile gerçekleştirilmektedir. Bu denklem daha öncede söz edildiği gibi düğümler arası feromon miktarını ve düğümler arası uzaklıkları kullanarak düğüm seçimini gerçekleştirmektedir. İlk turlar oluşturulurken yollardaki feromon miktarlarının ayırt edici bir özellik taşımadığı düşünüldüğünde ise yol seçim işlemi sadece sezgisel matris kullanılarak yani en yakın komşuluğun kullanımının ağırlıklı olduğu bir seçim ile gerçekleştirilmektedir. Bu durum rağmen karıncaların düğüm seçiminde kullandığı denklem en yakın komşuyu seçecek diye bir zorunluluk içermemektedir ve kimi zaman rastgelelik ağır basarak karınca tamamen herhangi kabul edilebilecek bir düğümü seçmektedir.

Bu durumlar göz önüne alındığında hybrid algoritma içinde düğüm seçim işlemi karıncalardakine benzer şekilde rastgele gerçekleştirilmektedir denilebilir. Karıncalar turlarını tamamlayıncaya kadar bir düğümden görüş olan (düğümler arası engel olmayan) bir diğer düğüme ilerleyerek yollarını tamamlamaktadır.

Tur uzunluklarının hesaplanarak en kısa turların bulunması: Bu işlem tıpkı KKA'da olduğu gibi hybrid algoritmalarda da gerçekleştirilmektedir. Karıncalar turlarını tamamladıktan sonra tur uzunlukları hesaplanarak en kısa turlar bulunmakta ve kayıt altına alınmaktadır.

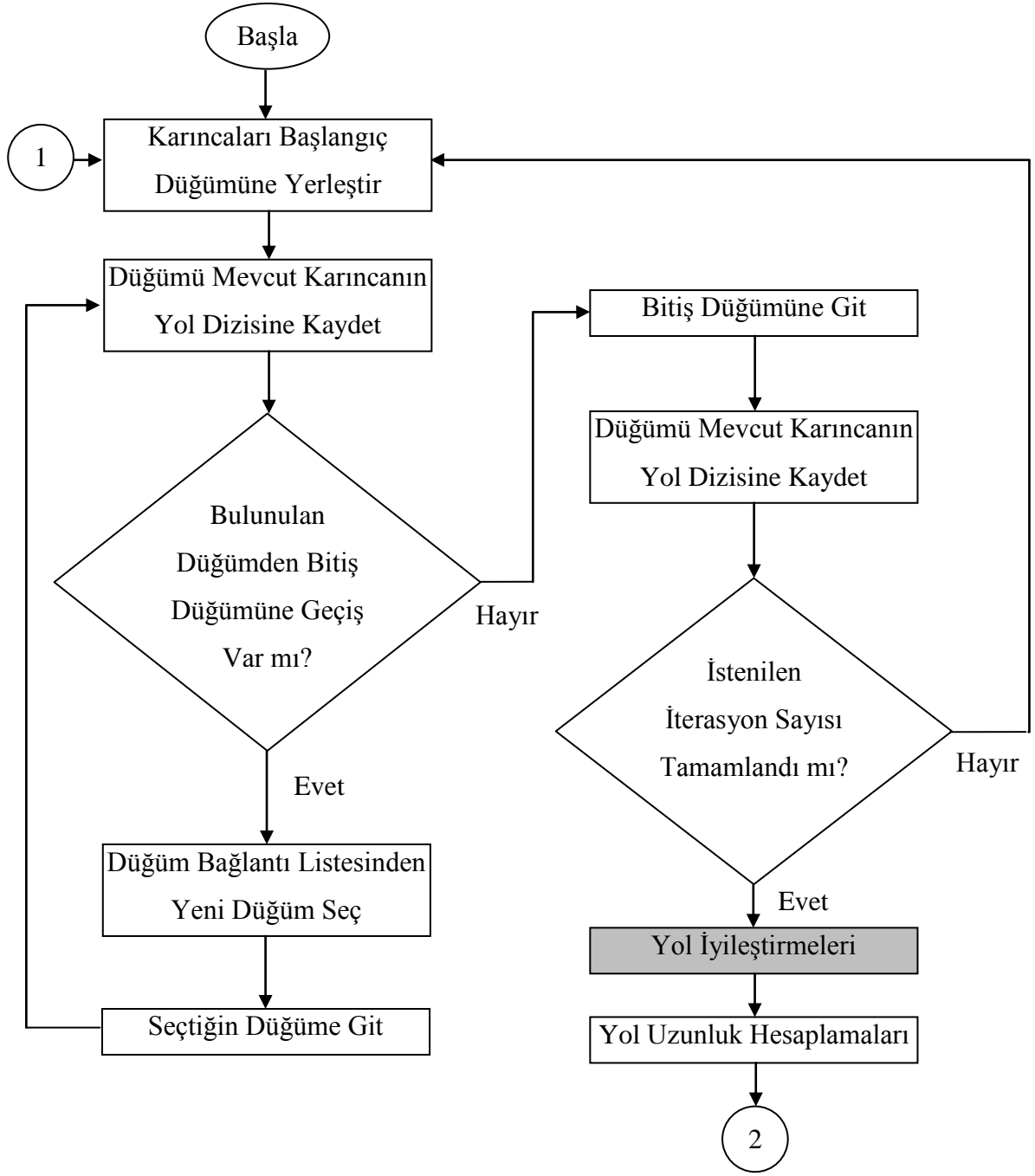
Feromon gncelleřtirmelerinin yapılması: Feromon gncellemeleri daha ncede bahsedildiđi gibi iki ařamadan oluřmaktadır. Bunlar feromon buharlařması ve eklenmesidir. Yine hatırlanacađı gibi feromon eklenmesi iřlemi farklı karınca algoritmalarında farklılık gsterebilmektedir. Kimi algoritmalarda feromon gncelleme iřlemi tamamlanmıř turlardaki tm dđmler iin eřit olarak gerekleřtirilirken kimi algoritmalar ise en iyi turların dđmleri iin ekstra bir gncelleme gerekleřtirmektedir.

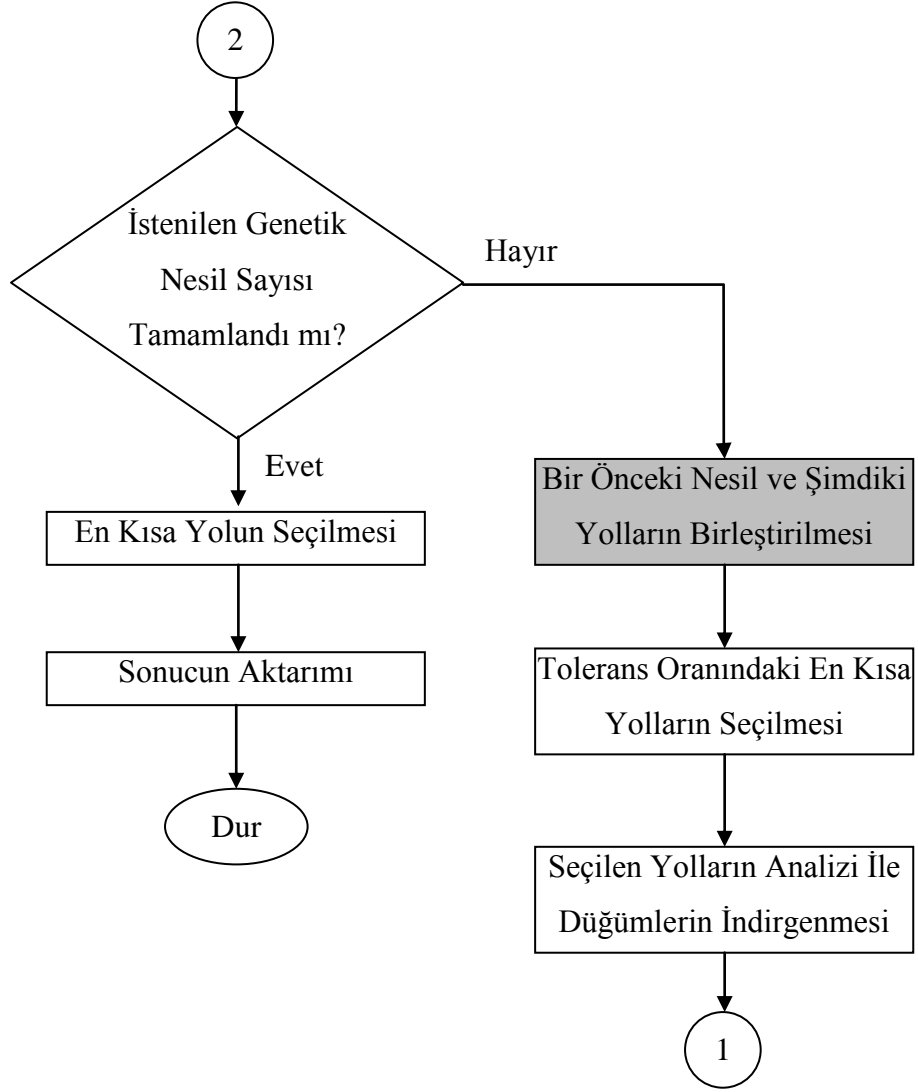
Feromon eklenmesi iřlemi hybrid algoritma iinde turlarını tamamlayan tm karıncaların yolları iin gerekleřtirilmektedir. Bunun nedeni hybrid algoritmada daha sonra kullanılacak GA’lardaki iyi olanların seimi iřleminin feromon gncelleme iřleminin etkisini arttıracak olmasıdır.

Feromon buharlařtırma iřlemi hybrid algoritmada kullanılmamaktadır. Kiřisel bir grř olarak feromon maddesinin tm yollarda deđerinin eřit oranda azaltılması oransal olarak bir deđeriklik sađlamayacađı iin bu iřlem kullanılmamaktadır.

IV.4 OLUŐTURULAN HYBRİD ALGORİTMA

Yukarıdaki blmlerde hybrid algoritmanın iřlem paralarının GA ve KKA’nın hangi iřlem basamaklarından geldiđi anlatılmıřtır. Bu blmde bahsi geen iřlem basamaklarının hybrid algoritma zerinde bir btn halinde sunumu yapılmaktadır. Őekil IV.3 de grlen akıř diyagramında ki gri blgeler genetik veya karınca algoritmalarının mantıđına ait olmayıp performans artıřı iin eklenmiřtir.





Şekil IV.3 Hybrid algoritma akış diyagramı

IV.5 ALGORİTMANIN PROBLEME UYGULANMASI

Algoritmanın ilgili problemde kullanımı öncesi yapılan işlemler (görüntü işleme ve düğüm tespiti işlemleri) ikinci bölümde anlatılmıştır. Bu bölümde gerçekleştirilen işlemler ile problemin teoriklikten daha uzak bir hal alması sağlanmış ve resimler üzerinde kullanımı sağlanmıştır. Gerçekleştirilen işlemler sonrası probleme kullanılmak üzere resim üzerindeki düğümler, düğümler arası bağlantılar ve bağlantılara ait uzunluklar oluşturulmuştur.

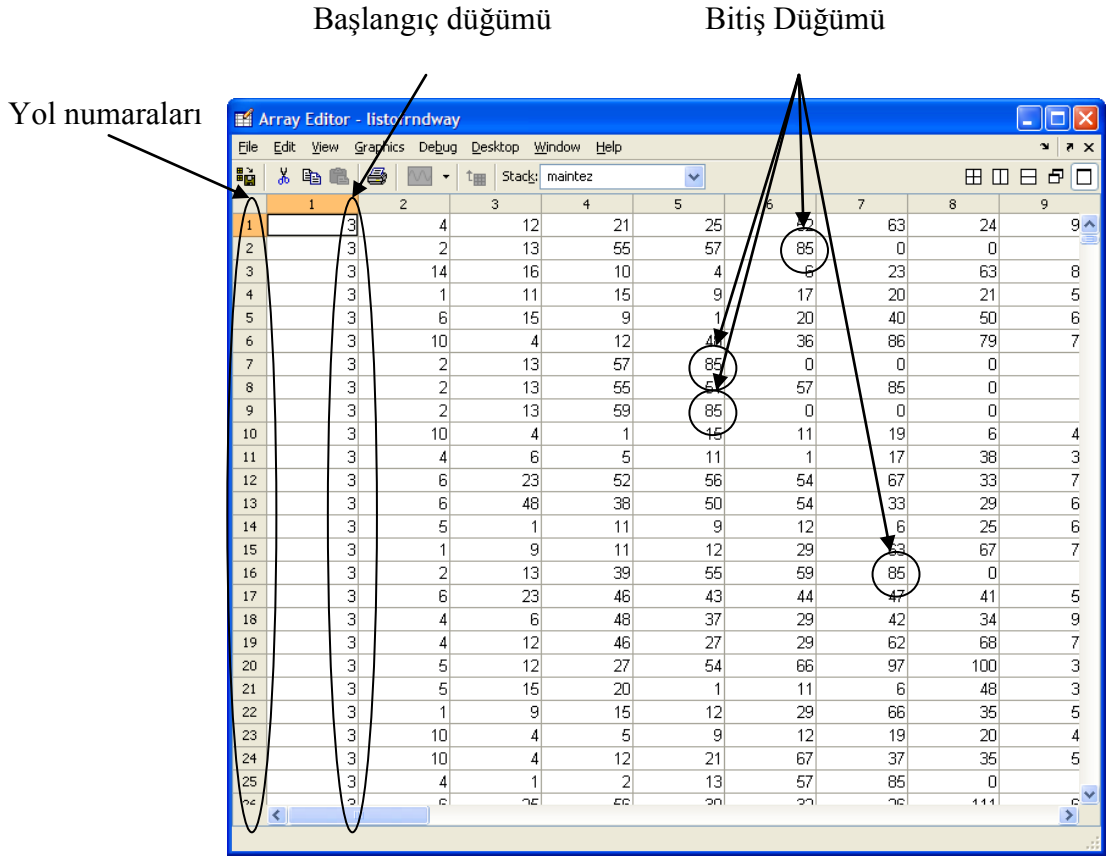
Aşağıdaki bölümlerde, sözü geçen değişkenlerinde kullanıldığı, algoritmanın eklenen ekstra performans artırıcı işlemler ile uygulanışı aktarılmaktadır.

IV.5.1 Hybrid Algoritmanın KKA'dan Alınan İşlemlerinin Gerçekleştirilmesi

Algoritmanın ilk aşamasında her bir yol bulma işlemi için karıncalar kullanıcıdan alınan başlangıç düğümüne yerleştirilir. Karıncaların oluşturacakları yollar düğüm dizileri halinde kayıt altında tutulmaktadır. Bundan dolayı karıncanın oluşturacağı yolun ilk düğümü olan başlangıç düğümü ilgili yolun ilk düğümü olarak kaydedilir.

Konumu başlangıç düğümü olan karınca hareket edebileceği (bulduğu düğüm ile arasında engel bulunmayan) düğümler arasında bitiş düğümü olup olmadığını kontrol eder. Karınca düğümler arası harekette kontrol işlemini gerçekleştirmek için önceden oluşturulmuş düğümler arası bağlantı matrisini kullanır. Bitiş düğümüne hareket olanağı bulunan karınca bitiş düğümüne giderek yolunu tamamlar. Bitiş düğümü ile bağlantı olanağı olmayan karınca ise bağlantı olanağı olan düğümlerin arasından olasılıksal bir düğüm seçerek seçtiği düğüme hareket eder. Bu işlem karıncanın bulunduğu düğümden bitiş düğümüne hareket olanağı bulması ve bitiş düğümüne hareket etmesine kadar gerçekleştirilir.

Yol bulma işlemini tamamlayan karıncaların oluşturdukları yollar daha öncede bahsedildiği gibi ardışık düğüm listeleri şeklinde kaydedilir. Her bir karıncaya ait yolun bir satırda ve her bir yola ait düğümlerin ilgili satırın hücrelerinde tutulduğu matrise ait örnek yollar Şekil IV.4 de gösterildiği gibidir.



Şekil IV.4 Yol kayıt listesi ve örnek yollar

Yol bulma işlemi ile tüm karıncaların yollarını tamamlamasının ardından kayıt altında tutulan yolların iyileştirme işlemi gerçekleştirilmektedir.

IV.5.2 Yol İyileştirme İşlemi

Yol iyileştirme işlemi, karıncaların oluşturdukları yolların kimilerinde daireler barındırıyor olmasından dolayı gerçekleştirilmektedir. Bir karınca düğümler arası ilerlerken daha önce uğradığı bir düğüme yeniden uğrarsa bu durum karıncanın daire çizdiğini göstermektedir. Karıncanın oluşturduğu yol daireler barındıran haliyle kötü bir çözüm olarak görülürken yoldan dairelerin çıkarılmasıyla iyi bir yola dönüştürülebilir. Bu nedenle karıncaların oluşturduğu yollarda iyileştirme işlemi yani yollardan dairelerin çıkarılması işlemi gerçekleştirilmiştir.

Daire barındıran bir yol ve bu yolun iyileştirilmesine yönelik örnek Tablo IV.1 de gösterilmektedir.

Tablo IV.1 Yol iyileştirme örneği

İyileştirilmemiş Yol	1	2	25	9	13	1	4	16
İyileştirilmiş Yol	1	4	16	0	0	0	0	0

İyileştirme işlemi gerçekleştirilen yolların uzunlukları hesaplanarak kayıt altında tutulmaktadır. Yol uzunluklarının hesaplanmasının ardından algoritmanın GA'lardan alınan işlemlerinin uygulanmasına geçilmektedir.

IV.5.3 Yeni Nesil Yollar İle Önceki Nesillerin Aynı Havuzda Tutulması

Turların tamamlanması ve yol uzunluklarının hesaplanması işleminin ardından algoritma genetik işlemlerin gerçekleştirildiği işlem basamağına geçmektedir. Ancak bu basamağına geçmeden önce tüm nesillere ait yolların bir araya getirilmesi işlemi gerçekleştirilmektedir. Bu işlem üretilen yolların ilk nesil yollar olmaması durumunda üretilen yollar ile bir önceki nesil yolların aynı havuzda toplanması ile gerçekleştirilmektedir.

Bu işlem algoritmanın genetik işlemlerde gerçekleştireceği seçim işleminin gelmiş geçmiş tüm nesiller içinden gerçekleştirilmesi amacını taşımaktadır. Bu sayede iyi bireylerin seçiminin yanı sıra bir sonraki nesle bozulmadan aktarılması sağlanmaktadır. Bu sayede olası iyi çözümler hiçbir zaman unutulmayacak ve daha iyisi üretilmemesi durumunda her zaman en tepede kalacaktır.

IV.5.4 Hybrid Algoritmanın GA'dan Alınan İşlemlerinin Gerçekleştirilmesi

GA'lardaki en iyi olan yaşar mantığının hybrid algortmada kullanıldığını daha önce belirtilmiştir. Bu mantık çerçevesinde yol havuzunda tutulan bireylerin, belirlenen yüzde diliminde iyi olanlarının seçimi gerçekleştirilmektedir. Seçim işlemi sonunda, GA'larda olduğu gibi iyi bireylerden yeni bireyler üretilmesi algortmaya özel farklılıklar ile sağlanmaktadır.

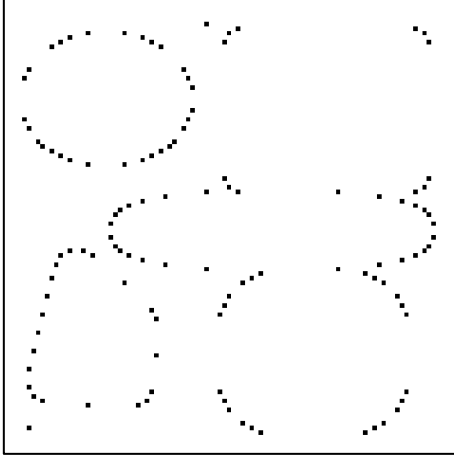
Daha öncede bahsedildiği gibi GA'lar iyi olan bireylerin genlerini çaprazlayarak daha iyi bireyler üretmeyi amaçlamaktadır. Ancak tüm düğümlerin

kullanılmadığı problem yapısından dolayı gerçekleştirilemeyen bu durum, iyi olan bireylerin genlerinin tek tek alınarak yeni bireyler üretilmesi ile sağlanmaktadır. Yollara ait düğümlerin genler kabul edildiği bu işlemde, iyi olan bireylerin genlerinin çözüm uzayı üzerinde işaretlenerek hayatta kalmaları sağlanmıştır.

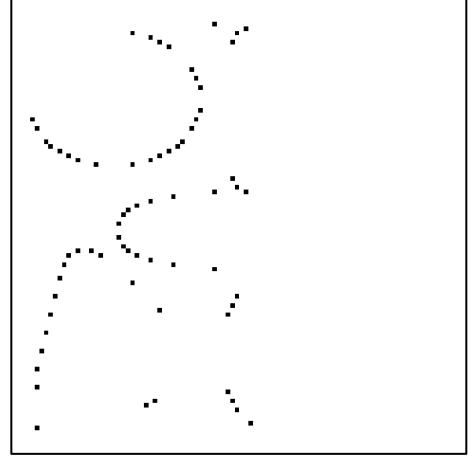
GA'larda olduğu gibi iyi bireylerin yaşaması kötü bireylerin ölmesi sağlanmasa da, iyi olan bireylerin genlerinin yaşaması ve kötü olan bireylerin genlerinin zaman ile unutulması sağlanarak hybrid algoritma performansında artış sağlanmıştır.

Örnek bir uygulama problemine ait şekiller ile ifade edilecek olursa:

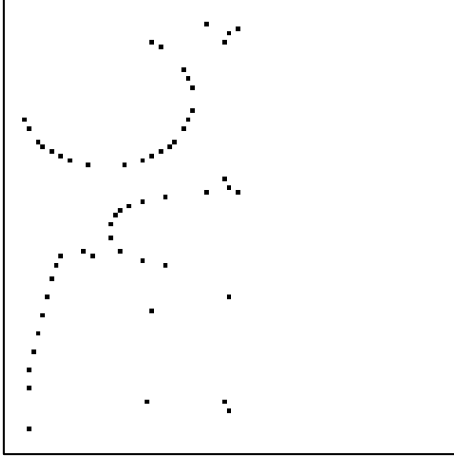
Şekil VI.5 de yollar üretilmeden önceki tüm olası çözüm düğümleri, Şekil VI.6 de ilk üretilen yolların en kısımlarının kullandığı çözüm düğümleri gösterilmektedir. Şekil VI.7 de birinci genetik işlemler sonucu kullanılan, Şekil VI.8 de ikinci genetik işlemler sonucu kullanılan, Şekil VI.9 de üçüncü genetik işlemler sonucu kullanılan çözüm düğümleri gösterilmektedir.



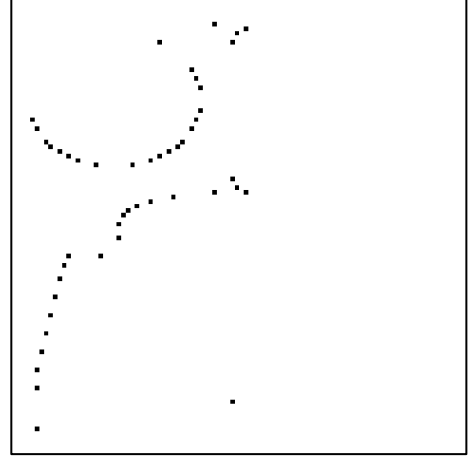
Şekil IV.5 Tüm olası çözüm düğümleri



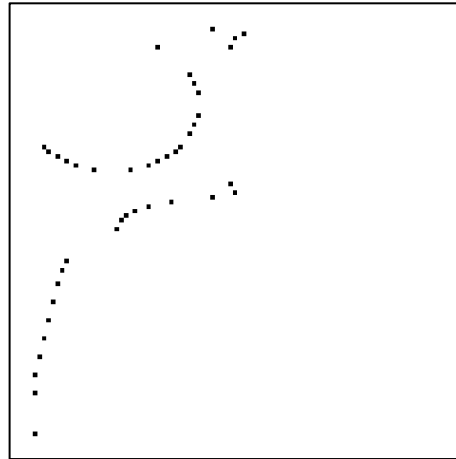
Şekil IV.6 İlk yolların kullandığı düğümler



Şekil IV.7 Birinci genetik seçim sonrası
kullanılan düğümler



Şekil IV.8 İkinci genetik seçim sonrası
kullanılan düğümler



Şekil IV.9 Üçüncü genetik seçim sonrası kullanılan düğümler

IV.6 UYGULANAN ALGORİTMANIN ÖRNEK SONUÇLARI

Engellerin bulunduğu ortamda en kısa yol bulma işlemi, genellikle araçlar için rota çizme ve robotlar için engellerden sakınma gibi problemlerin çözüme kavuşturulması işleminde sıkça kullanılmaktadır. Bu problemlerin çözümlerine yönelik simülasyonlar gerçekleştirilirken ise genellikle labirent, kroki ve grid benzeri ortamlar engellerin bulunduğu ortamlar için örnek teşkil etmektedir.

Geliştirilen hybrid algoritmanın performansının değerlendirilebilmesi amacı ile örneklerde ortam görüntülerine ait öznitelikler, başlangıç düğüm miktarları ve genetik işlemler sonrası son kalan düğüm miktarları değerlendirilmeye sunulmuştur.

Şekillerle birlikte verilen parametreler şu şekildedir:

X= Bitmap görüntüsünün piksel bazında genişlik değeri,

Y= Bitmap görüntüsünün piksel bazında yükseklik değeri,

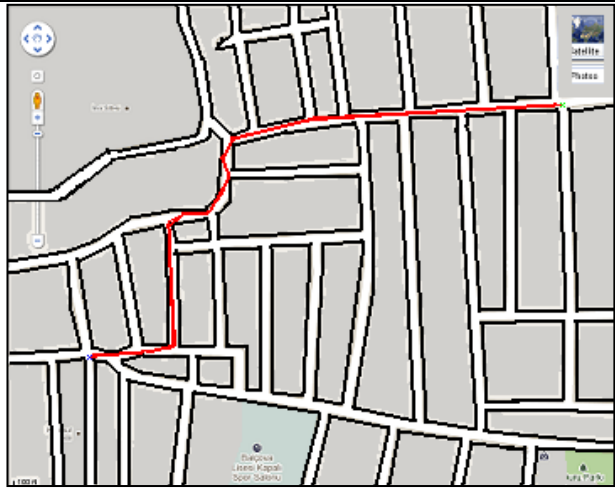
m= Görüntüden elde edilen başlangıç düğüm sayısı,

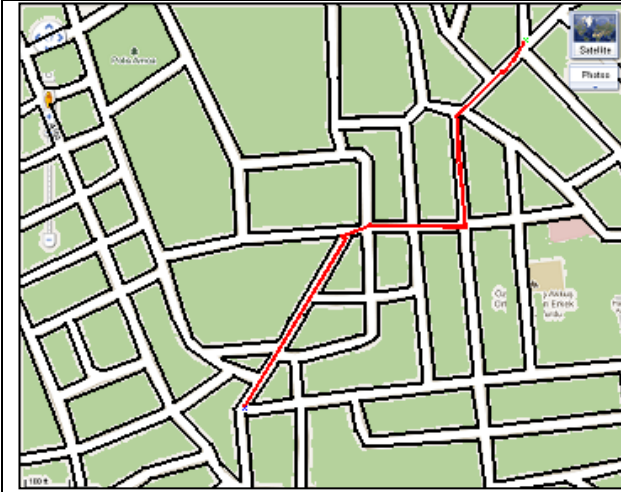
n= Genetik işlemler sonrası kalan düğüm sayısı,

YD= Bulunan en kısa yolun düğüm sayısı.

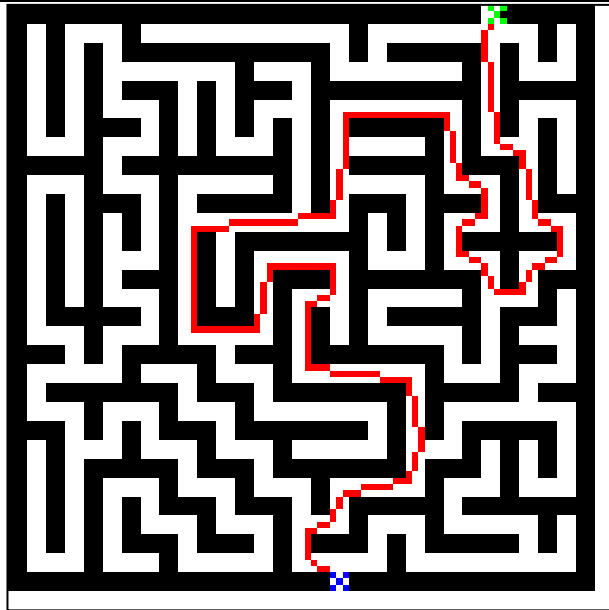
Tablo IV.2 de örnek çözüm görüntüleri ve performans kriterlerine yönelik değişkenler gösterilmektedir.

Tablo IV.2 Örnek çözüm görüntüleri

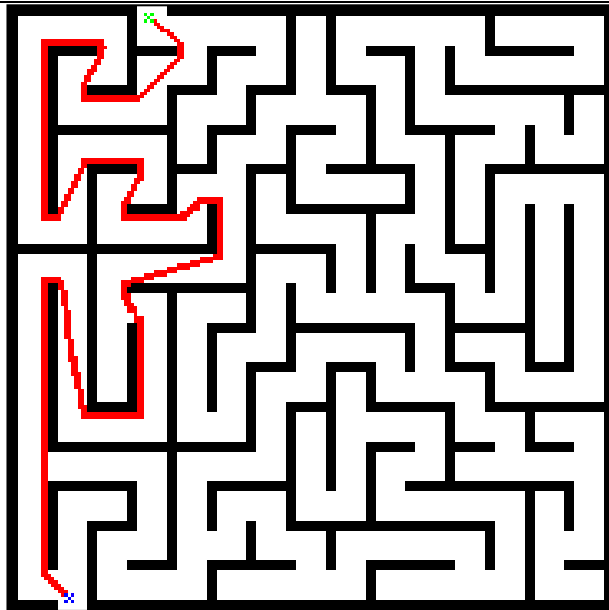
Çözüm Görüntüsü	İlgili Değerler
	X = 360 piksel Y = 286 piksel m = 839 düğüm n = 421 düğüm YD = 15 düğüm



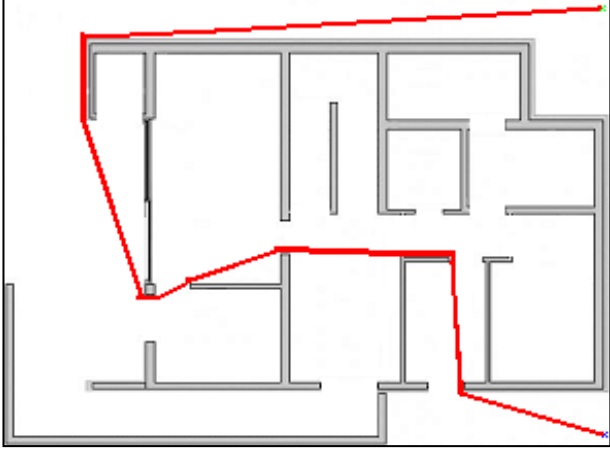
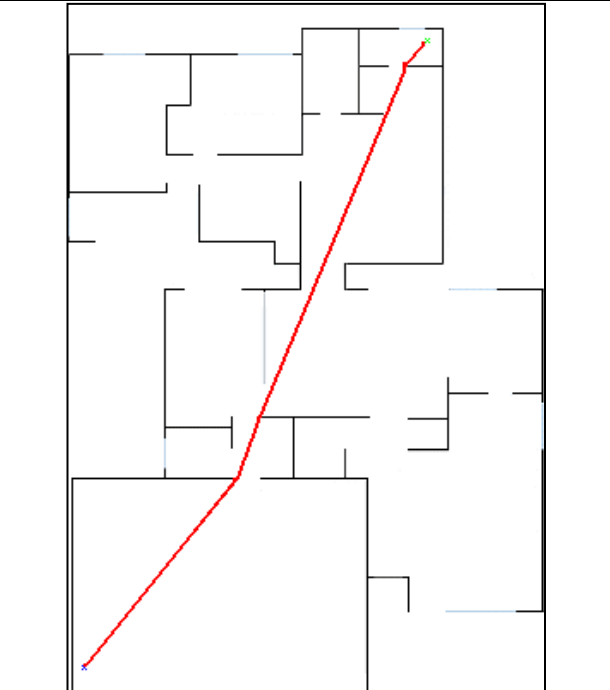
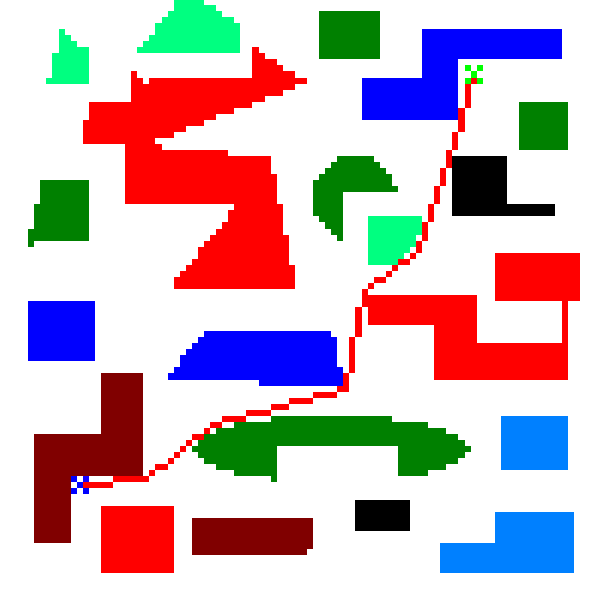
$X = 360$ piksel
 $Y = 288$ piksel
 $m = 1641$ düğüm
 $n = 151$ düğüm
 $YD = 9$ düğüm



$X = 96$ piksel
 $Y = 96$ piksel
 $m = 170$ düğüm
 $n = 54$ düğüm
 $YD = 36$ düğüm



$X = 183$ piksel
 $Y = 183$ piksel
 $m = 151$ düğüm
 $n = 36$ düğüm
 $YD = 29$ düğüm

	<p>X = 294 piksel Y = 218 piksel m = 61 düğüm n = 50 düğüm YD = 11 düğüm</p>
	<p>X = 294 piksel Y = 218 piksel m = 86 düğüm n = 15 düğüm YD = 5 düğüm</p>
	<p>X = 100 piksel Y = 100 piksel m = 155 düğüm n = 120 düğüm YD = 8 düğüm</p>

Geliştirilen hybrid algoritmanın performansına yönelik değerlendirmeler gerçekleştirilirken görüntü boyutu, düğüm sayısı ve genetik işlemler sonrası kalan düğüm sayıları göz önünde tutulmuştur. Bu kriterlerin neye göre göz önünde tutulduğu ve algoritmanın geri kalan nitelik değerlendirmeleri aşağıdaki gibidir.

IV.7.1 Görüntü Boyutu

Görüntü boyutu yada diğer bir ifade ile simülasyon ortamının boyutu, problemin büyüklüğü hakkında direk fikir sahibi olmaya yarayan birinci dereceden bir etmendir.

Şu ana kadar yapılan araştırmalarda, engelsiz geniş alanlar için herhangi bir indirgeme yapılmadığı ya da bu alanlar için referans düğümler atanmakta olduğuna dair izlenimler edinilmiştir. Bu durum büyük görüntüler için referans düğümler atanması, küçük boyutlu örnek teşkil edecek görüntüler için ise alan indirgemenin kullanılması olarak açıklanabilir.

Gerçekleştirilen görüntü işleme işlemleri ile ortam hazırlanması direk olarak hybrid algoritmanın bir parçası olmasa da, algoritma performansını arttırmaya yönelik bir ön adım olarak düşünülecek olursa görevini yerine getirmektedir denilebilir. Klasik yöntemlerde büyük ölçekli görüntüler için (örneğin 800x600 boyutlarında) referans düğümler belirlenirken bu referansların ne kadar mükemmel olduğu ve hangi kriterlere göre belirlendiği bilinmemektedir. Kullanılan görüntü işleme yönteminde ise düğümler doğrusal bir mantık ile oluşturulmakta ve olası tüm çözümlerin düğümlerini kapsamaktadır. Bu işlemin klasik yöntemlere göre performans artışı sağladığı söylenebilmesine rağmen mükemmel olduğu söylenememektedir.

Daha öncede söz edildiği gibi MATLAB ortamının kısıtlılıklarından kaynaklanan nedenlerden dolayı, düğüm indirgeme işlemleri 45 derece ve katlarındaki eğimlere sahip doğrular üzerinde gerçekleştirilebilmiştir. Reel hayat ve işlemler düşünüldüğünde bu derece kısıtlılığının ortadan kalkması sonucu, düğüm indirgeme işlemi çok daha başarılı bir şekilde gerçekleştirilebilecek ve hybrid algoritma kullanımı için çok daha elverişli bir ortam sağlanabilecektir. İkinci

bölümde anlatılan görüntü işleme işlemlerinin farklı ortamlarda gerçekleştirilmesi performans da artış sağlayabilecektir.

Tüm bu anlatılan durumlara rağmen, algoritma benzer kriterleri kullanmış görüntüler üzerinde değerlendirildiğinde çok daha büyük görüntü boyutlarında işlem yapabilmektedir. Şu ana kadar yapılmış gözlemlerde, engellerin bulunduğu ortamlara yönelik referans düğümler verilmeden gerçekleştirilen araştırmalarda, görüntü boyutlarının genellikle 5x5 veya 10x10, maksimum ise 20x20 veya 30x30 boyutlarında kullanıldığı gözlemlenmiştir[80-83].

Benzer şekilde görüntü boyutunun düğüm boyutu ile aynı olduğu (10x10 boyutlarındaki bir görüntü 100 düğüm barındırıyor demektir) çalışmalar ile karşılaştırılacak olursa uygulanan yöntem ve algoritma çok yüksek performans veriyor denilebilir.

Sonuç olarak değerlendirmek gerekirse, algoritmanın yüksek çözünürlükteki görüntüler üzerindeki performansı geçmiş örneklerine göre yüksektir denilebilir.

IV.7.2 Engel Çeşitliliği

Daha önceden gösterilen örnek görüntüler ile de hatırlanabileceği gibi, görüntü işleme ve hybrid algoritmanın birlikteki performansı herhangi bir engel tipi ile sınırlandırılmamıştır.

Geçmiş çalışmalara göz atıldığında engellerin bulunduğu ortamlara yönelik engel tespiti belirli şekiller ile kısıtlandırılmaktadır. Bu şekiller kimi çalışmalarda daireler iken kimi çalışmalarda düzgün şekiller olarak tabir edilebilecek köşeli cisimlerdir[84-86]. Bunun nedeni bu gibi engel tiplerinde belli başlı formülasyonlara sağdık kalınarak engellerin etrafından dolaşılmasının sağlanabilmesidir.

Gerçekleştirilen yöntemler ve algoritma ile engel tipi kısıtlılığı tamamen ortadan kaldırılmış ve birçok alanda uygulanabilirliği kanıtlanmıştır.

IV.7.3 Düğüm Sayısı

Algoritmanın performansı düğüm sayısı üzerinden değerlendirilecek olursa yine benzer örneklere göre daha yüksek performans sağlamaktadır denilebilir.

Hatırlanacak olursa ilk ortaya atılan karınca algoritmaları 75 ve üstü düğüm sayısında başarısız sonuçlar üretmiştir. Ancak daha sonra geliştirilen KKA ve benzeri algoritmalar düğüm sayısı konusunda çitayı biraz daha yükseğe çekebilmiştir. Kesin bir limit olmamakla birlikte, genel de görülen en yüksek örnekler 100 ila 200 düğüm arası problem büyüklüğüne sahip örneklerdir denilebilir. Bunun yanı sıra yüksek kapasiteli bilgisayarlar ve yüksek iterasyon sayıları ile gerçekleştirilen işlemler ile bu sayının biraz daha yüksek değerlere ulaşabileceği tahmin edilebilmektedir.

Gerçekleştirilen hybrid algoritma ile verilen örneklerde de görülebileceği gibi düğüm sayısına bağlı performans ölçütünden algoritma başarı ile geçmiş denilebilir. Örneklerdeki düğüm sayısı değerlerine dikkat edilecek olursa özerlikle krokiler üzerinde çok yüksek düğüm sayıları ile işlem yapılmaktadır. Ancak kroki gibi yüksek düğüm sayısı barındıran örneklerde iterasyon sayısı artırılarak çözümlerin iyileştirildiği belirtilmelidir. Diğer örnek çeşitlerindeki oranla 4-5 kat daha fazla iterasyon sayısı ile gerçekleştirilen yol bulma işlemi sonucunda göz ardı edilebilecek bir zaman kaybı oluşsa da yol bulmada istenilen yakınlık sağlanmıştır.

IV.7.4 Genetik İşlemler Sonrası Düğüm Sayıları

Bu kriter gerçekleştirilen işlemlerin performans kriteri olarak değerlendirilmeyebilir ancak işlemlerin performansındaki etkisi göz ardı edilmemelidir. Problemin başlangıcındaki düğüm sayıları ile genetik işlemlerin belli tekrarları sonrası kalan düğüm sayıları değerlendirildiğinde, problemlerin ölçeklerinin bir nevi küçültüldüğü düşünülebilir.

İlk yola çıkan karıncaların çözüm için elinde bulunan düğüm sayısı ile bu düğümlerin zamanla indirgenmesi sonucu, son yolları oluşturan karıncaların çözüm için elinde bulunan düğüm sayısı farklılık göstermektedir. Bu farklılığın bir sonucu olarak ise en iyi yolun oluşturulma ihtimali arttırılmakta, karıncalar için daha az seçenek sunularak aynı zamanda işlem zamanından da kazanç sağlanmaktadır.

Bu durum, kroki gibi yüksek düğüm sayılı örneklerde, ilerleyen safhalardaki yol bulma işleminin daha kaliteli ve kısa sürede olmasına neden olmaktadır. Daha öncede bahsedildiği gibi GA'lardan alınan bu mantık ile iyi olan düğümlerin etkin, kötü olan düğümlerin etkisiz kalması sağlanmaktadır.

IV.7.5 İşlem Zamanı

Algoritmanın işlem zamanına yönelik gerçekçi bir değerlendirme yapılamamaktadır. Bunun başlıca nedenleri problem boyutuna göre iterasyon sayısı gereksiniminin değişimi ve karşılaştırma yapılacak örneklerin azlığıdır.

Daha önce yapılan engellerin bulunduğu ortamlara yönelik çalışmalar daha çok sonuç kalitesi üzerine yoğunlaşmakta ve ele aldıkları problem boyutları ise bizim kullandığımız problem boyutuna göre küçük kalmaktadır.

Ancak bütün bunlara rağmen çözüm süresinin geçmiş araştırmalarda yapılan küçük boyutlu örneklere oranla daha başarılı olduğu söylenebilir.

IV.7.6 Sonuç Kalitesi

Sonuç kalitesi değerlendirilecek olursa, tüm diğer sezgisel algoritmalarda olduğu gibi, sonuç en kısa yoldur diye kesin bir tabir kullanılamamaktadır. Buna rağmen ele alınan örnekler incelendiğinde ve sezgisel algoritmadaki sonuçların belli bir yakınsal değer elde edilmesi durumunda başarılı kabul edildiği göz önüne alındığında algoritma başarılıdır denilebilir.

Örneklerin incelenmesi ile kişisel görüş olarak bulunan tüm sonuçlar en kısa yola en yakın sonuçlardır denilebilir. Ancak en kısa yolun bulunmasının iterasyon sayısı ve genetik işlem sayısı ile bağlantılı olduğu belirtilmelidir.

BÖLÜM V SONUÇ VE ÖNERİLER

Bu bölümde bu çalışma ile elde edilen sonuçlara ve gelecek çalışmalar için önerilere yer verilmektedir.

Sonuç:

Çalışmada en kısa yol bulmaya yönelik yapay zeka algoritmaları incelenmiştir. İnceleme işlemi gerçekleştirilirken algoritmaların çalışma mantığı aktarılmaya çalışılmış ve bunun ardından değerlendirilmeler yapılmıştır. Değerlendirmeler yapılırken yapılan araştırmalar sonucu algoritmalara ait avantajların yanı sıra kısıtlılıklar derlenmiştir. Bu kısıtlılıklar ve avantajlar üzerine, kişisel yorumlarında etkin olduğu tartışmalar sonucu hybrid algoritma gereksiniminin nereden geldiği aktarılmaya çalışılmıştır.

Yapılan çalışma sonucunda görülmüştür ki, matematiksel denklemlere bağlı olmayan optimizasyon teknikleri kullanım olarak pratik olsa da sonuç olarak kesin ve iyidir denilememektedir. Özellikle sezgisel algoritmaların performansları değerlendirildiğinde, sonuçların tamamen dizayn yapan kişinin performansı ile alakalı olduğu gözlemlenmiştir.

Algoritmalara ait parametrelerin yanlış, eksik veya yetersiz girilmesi durumunda kaliteli çözümler oluşmayacağı kesindir. Ancak bu durum tüm matematiksel denklemler için de geçerlidir. Herhangi bir denklemdeki değerin yanlış verilmesi durumunda denklem doğru sonuç üretmemektedir. Ne var ki bu durum sezgisel algoritmaların performans değerlendirmesindeki rolü tamamen kullanıcının üzerine aktarmaya yetmemektedir. Bunun yegane nedeni ise herhangi bir matematiksel denklemde değişkenlerin hangi aralıklarda ve neye bağlı olduğu net olarak belirlenirken sezgisel algoritmaların parametrelerinde bu sağlanamamaktadır.

Algoritmaları bulan kişilerin dahi deneme yanılma yöntemleri ile elde ettikleri parametre değerleri problemden probleme farklılık göstermektedir. Tamamen tavsiyelere bağlı olarak belirlenen parametre değerleri yanlış belirlenmiş ise, bu yanlışlığı düzeltmenin tek yolu deneyerek daha iyisini bulmaktır.

Sonuç olarak denilebilir ki, sezgisel algoritmaların iyi sonuç üretme ihtimalleri yüksektir. Ancak bu algoritmalar ne doğru çalışmayı nede iyi sonuç üretmeyi garanti etmemektedir.

Oluşturulan hybrid algortmada netice de sezgisel algoritmaların mantığını kullanan bir yapıdadır. Algoritma GA ve KKA'dan farklı olarak daha az parametre kullanan bir yapıda dizayn edilmeye çalışılmıştır. Bu nedenle kullanıcıya düşen risk ve sorumluluk azaltılmıştır denilebilir.

Algoritmanın iki parametresi kullanıcıdan istenmektedir. Bu parametreler iterasyon sayısı (karınca yol bulma sayısı) ve yol bulma işlemleri ardından gerçekleştirilecek genetik işlem (düğüm indirgemesi ile yeni nesle hazırlık işlemi) sayısı parametreleridir. Bu parametrelerin kullanıcıya bırakılmasının nedeni, küçük ölçekli problemler için gereksiz zaman kaybını önlemek ve büyük ölçekli problemler için yetersiz iterasyon sayısı gibi durumları ortadan kaldırma ihtimalini olanaklı kılmaktır.

Oluşturulan hybrid algoritma ile alakalı olarak iyi performans sergiliyor denilebilir Ancak ne olursa olsun algoritma sezgisel algoritmaların karakteristik yapısındadır ve sonucun en iyi olmasını garantilemez. Sonucun kalitesi hala iterasyon oranı ve genetik işlem sayısına bağlı kalmaktadır.

Öneriler:

Bundan sonra bu konu ile ilgili yapılacak çalışmalarda simülasyon gerçekleştirmek için kullanılan programın daha esnek yapıda olması tavsiye edilebilir. Daha öncede söz edildiği gibi düğüm elemsi gibi işlemlerde MATLAB programının kısıtlılıkları ile karşılaşmıştır. Bu kısıtlılıklar başka programlar ile ortadan kaldırılırsa performans artışı sağlanacağına inanılmaktadır.

Oluşturulan algoritma ne kadar işlevsel olursa olsun, kullanımı ne kadar pratik olursa olsun hiçbir zaman matematiksel yöntemler ile bulunan en kısa yolun yerini tutamaz. Bu bütün optimizasyon teknikleri için geçerlidir. Sezgisel optimizasyon tekniklerinin işlem zamanı, maliyeti ve pratikliği açısından avantajları olduğu göz ardı edilemez ancak bütün bu avantajlar doğruluk kıstası ile karşılaştırıldığında çokta önemli sayılmamalıdır.

Bir problemin çözümlü için binlerce yol olabilir, binlerce farklı çözümlü de olabilir ama optimum çözümlü yolu bir tanedir ve en büyük kriteri her zaman çözümlü dođruluđudur.

KAYNAKLAR

- [1] Nabiyev, V. V.: *Yapay Zeka: Problemler-Yöntemler-Algoritma*.(2005)
- [2] Antsaklis, P.; Stiver, J. ; Lemmon, M.: *Hybrid system modeling and autonomous control systems*. Vol. 736 Springer Berlin / Heidelberg,(1993)
- [3] Ahn, C. ; Ramakrishna, R.: "A Genetic Algorithm For Shortest Path Routing Problem And The Sizing Of Populations". *IEEE Transactions on Evolutionary Computation* 6, (2002).
- [4] Colorni, A.; Dorigo, M. ; Maniezzo, V.: "Distributed Optimization By Ant Colonies". *European Conference On Artificial Life*, (1992), 134-142.
- [5] Engin, O.: "Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma ile Çözüm Performansının Arttırılmasında Parametre Optimizasyonu", İTÜ Fen Bilimleri Enstitüsü, Yayınlanmamış Doktora Tezi, İstanbul, (2001).
- [6] Paçacı, Y. ; Pekdoğan, E.: "İki Nokta Arasındaki En Kısa Yolun Görüntü İşleme Yöntemi ve Yapay Zeka Algoritmaları Kullanılarak Bulunması", Marmara Üniversitesi Teknik Eğitim Fakültesi, İstanbul, (2008).
- [7] Kayıcı, S.: "Yapay Zeka Ders Notlarının Hazırlanması" *Lisans* thesis, Marmara Üniversitesi Teknik Eğitim Fakültesi, İstanbul, (2007).
- [8] Goldberg, D.: *Genetic Algorithms In Search, Optimization, And Machine Learning*. Addison-Wesley Reading Menlo Park,(1989)
- [9] Braun, H.: "On Solving Travelling Salesman Problems By Genetic Algorithms". *Parallel Problem Solving from Nature*, (1991), 129-133.
- [10] Merz, P. ; Freisleben, B.: "GeneticLocal Search For The TSP: New Results". (1997), 159-164.
- [11] Freisleben, B. ; Merz, P.: "A Genetic Local Search Algorithm For Solving Symmetric And Asymmetric Traveling Salesman Problems ". (1996), 616-621.
- [12] Freisleben, B. ; Merz, P.: "New Genetic Local Search Operators For The Traveling Salesman Problem". *Parallel Problem Solving from Nature-PPSN IV*, (1996), 890-899.
- [13] Uğur, A. ; Aydın, D.: "Ant System Algoritmasının Java İle Görselleştirilmesi". (2006).

- [14] Dorigo, M.; Maniezzo, V. ; Colorni, A.: "The Ant System: An Autocatalytic Optimizing Process". *TR91-016, Politecnico di Milano*, (1991).
- [15] Gambardella, L. ; Dorigo, M.: "Ant-Q: A Reinforcement Learning Approach To The Traveling Salesman Problem". (1995), 252-260.
- [16] Dorigo, M.; Maniezzo, V. ; Colorni, A.: "The Ant System: Optimizasyon By A Colony Of Cooperating Agents". *IEEE Transaction On Systems Vol.26*, (1996), 12.
- [17] Dorigo, M. ; Gambardella, L.: "Ant Colonies For The Travelling Salesman Problem". *BioSystems* 43, (1997), 73-82.
- [18] Gen, M.; Cheng, R. ; Wang, D.: "Genetic Algorithms For Solving Shortest Path Problems". (1997), 401-406.
- [19] Soehodho, S.: "Development Of Genetic-Taxonomy Evaluator For Finding Shortest Path In Transportation Planning Systems". *Urban Transport And The Environment For The 21st Century III*, (1997), 177.
- [20] Yu, Y. ; Wu, J.: "Loads Combination Method Based Core Schema Genetic Shortest-Path Algorithm For Distribution Network Reconfiguration". 3, (2002).
- [21] Selvanathan, N. ; Tee, W.: "A Genetic Algorithm Solution To Solve The Shortest Path Problem In Ospf And Mpls". *Malaysian Journal of Computer Science* 16, (2003), 58-67.
- [22] Abeyesundara, S.; Giritharan, B. ; Kodithuwakku, S.: "A Genetic Algorithm Approach to Solve the Shortest Path Problem for Road Maps". (2005), 272-275.
- [23] Kan, Z.: "Shortest Path Analysis Based On Genetic Algorithm". *Journal of Shangqiu Teachers College*, (2007).
- [24] Lin, F.; Lee, M. ; Fuh, C.: "A Two-Stage Genetic Algorithm for Solving Shortest Path Problem with Fuzzy Arc Lengths". (2007), 249-249.
- [25] Lin, L. ; Gen, M.: "A Bicriteria Shortest Path Routing Problems By Hybrid Genetic Algorithm In Communication Networks". (2007), 4577-4582.
- [26] Pennisi, M.; Pappalardo, F.; Motta, A. ; Cincotti, A.: "A Genetic Algorithm for Shortest Path Motion Problem in Three Dimensions". *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, (2007), 534-542.

- [27] Behzadi, S.; Alesheikh, A. ; Poorazizi, E.: "Developing A Genetic Algorithm To Solve Shortest Path Problem On A Raster Data Model". *Proc. of Journal on Applied Sciences* 8, (2008), 3289-3293.
- [28] Pires, M.; da Silva, I. ; Bertoni, F.: "Solving Shortest Path Problem Using Hopfield Networks And Genetic Algorithms". (2008), 643-648.
- [29] Yussof, S.; Razali, R. ; See, O.: "A Parallel Genetic Algorithm for Shortest Path Routing Problem". (2009), 268-273.
- [30] Nallusamy, R.; Duraiswamy, K.; Muthukumar, D. ; Sathiyakumar, C.: "Energy Efficient Dynamic Shortest Path Routing In Wireless Ad Hoc Sensor Networks Using Genetic Algorithm". (2010), 1-5.
- [31] Yuhe, B.: "An Improved Ant Colony Algorithm for the Shortest Path Problem". *Computer Engineering and Applications* 3, (2003).
- [32] Jinjiang, F. ; Da, Y.: "Solving a Shortest Path Problem by Ant-algorithm". *Computer Engineering and Applications* 33, (2004).
- [33] Khan, M.: "Ant System To Find The Shortest Path". (2004).
- [34] Li, Y. ; Jin-song, H.: "Multi-Thread Ant Colony Algorithm And Its Application In The Shortest Path". *Logistics Technology* 2, (2005).
- [35] Liu, S.; Lin, J. ; Lin, Z.: "A Shortest-Path Network Problem Using An Annealed Ant System Algorithm". (2005).
- [36] Min, L. ; Yang, J.: "A Shortest-Path Routing Based On Ant Algorithm". 2, (2005), 67-69.
- [37] Shang, G. ; Jingyu, Y.: "Convergence Analysis of Ant Colony Algorithm for Solving Shortest Path Problem". (2006).
- [38] Chibaya, C. ; Bangay, S.: "A Probabilistic Movement Model For Shortest Path Formation In Virtual Ant-Like Agents". (2007), 18.
- [39] Jiang, Y.; Wang, W. ; Zhao, Y.: "Solving The Shortest Path Problem In Vehicle Navigation System By Ant Colony Algorithm". (2007), 188-192.
- [40] Shuang-hua, W.; Yang, F. ; Liang, G.: "Study on Shortest Path Search Method Based on Chaos Ant Colony Optimization". *Journal of Chongqing Jiaotong University (Natural Science)*, (2007).
- [41] Wen-qun, B.: "Study On Advanced Ant Colony Algorithm Searching Shortest Path In Logistics Network Of Distribution". *Computer Engineering and Design* 19, (2008).

- [42] Gao, H.; Liu, Y.; Chang, Q. ; Xiong, H.: "An Improved Ant Colony Algorithm for the Shortest Path Problem in Time-Dependent Networks". **(2009)**.
- [43] Ghoseiri, K. ; Nadjari, B.: "An Ant Colony Optimization Algorithm For The Bi-Objective Shortest Path Problem". *Applied Soft Computing*, **(2009)**.
- [44] Ok, S.; Seo, W.; Ahn, J.; Kang, S. ; Moon, B.: "An Ant Colony Optimization Approach For The Preference-Based Shortest Path Search". *Communication and Networking*, **(2009)**, 539-546.
- [45] Zhong-qiu, Y.; Zhang, Y. ; Zheng, Z.: "Analysis and Simulation of Shortest Path Problem Based on Modified Ant Colony Algorithm". *Journal of Shenyang University of Chemical Technology* 2, **(2009)**.
- [46] Yu, F.; Li, Y. ; Wu, T.: "A Temporal Ant Colony Optimization Approach To The Shortest Path Problem İn Dynamic Scale-Free Networks". *Physica A: Statistical Mechanics and its Applications* 389, **(2010)**, 629-636.
- [47] Zakzouk, A.; Zaher, H. ; El-Deen, R.: "An Ant Colony Optimization Approach For Solving Shortest Path Problem With Fuzzy Constraints". **(2010)**, 1-8.
- [48] "Shortest path problem",
<http://en.wikipedia.org/wiki/Shortest_path_problem> **(12.04.2011)**.
- [49] Nilsson, N. J.: *Principles Of Artificial Intelligence*. Springer Verlag,**(1982)**
- [50] De Rezende, P. J.; Lee, D. T. ; Wu, Y.: "Rectilinear Shortest Paths With Rectangular Barriers". **(1985)**, 204-213.
- [51] Çunkaş, M.: "Genetik Algoritmalar ve Uygulamaları - Ders Notları". *Selçuk Üniversitesi*, **(2006)**.
- [52] İşçi, Ö. ; Korukoğlu, S.: "Genetik Algoritma Yaklaşımı ve Yöneylem Araştırmasında Bir Uygulama". *Yönetim ve Ekonomi* 10, **(2003)**.
- [53] Herrero, J.; Blasco, X.; Martinez, M. ; Salcedo, J.: "Optimal PID Tuning with Genetic Algorithms for Non-Linear Process Models". **(2002)**.
- [54] Ünal, M.: "PID Kontrolörün Karınca Kolonisi / Genetik Algoritma Tabanlı Optimizasyonu Ve Gunt Rt 532 Basınç Prosesinin Kontrolü", Marmara Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, **(2008)**.
- [55] Emel, G. ; Taşkın, Ç.: "Genetik Algoritmalar Ve Uygulama Alanları". *İktisadi ve İdari Bilimler Fakültesi Dergisi* 21, Uludağ Üniversitesi, Bursa, **(2002)**.

- [56] Bourg, D. ; Seeman, G.: *AI for Game Developers* O'Reilly Media Inc, USA,(2004)
- [57] Jones, T.: *AI Application Programming*. Vol. 2 Charles River Media,(2003)
- [58] Ursem, R.: "Models For Evolutionary Algorithms And Their Applications In System Identification And Control Optimization". (2003).
- [59] Thenorio, W. A.: "Genetic Algorithms", <<http://www.cs.chalmers.se/Cs/Grundutb/Kurser/algsem/Projects2007/GeneticAlgorithms.pdf>> (12.05.2010).
- [60] Russel, J. S. ; Norvig, P.: *Artificial Intelligence. A Modern Approach*. Printice Hall,NJ,(1995)
- [61] Marczyk, A.: "Genetic Algorithms and Evolutionary Computation", <<http://www.talkorigins.org/faqs/genalg/genalg.html#limitations>> (18.05.2010).
- [62] Koza, J.: *Genetic Programming: On The Programming Of Computers By Means Of Natural Selection.*, The MIT press, London, England,(1992)
- [63] Buseti, F.: "Genetic Algorithms Overview". (2010).
- [64] "Genetic Algorithms", <<http://www.tjhsst.edu/~ai/AI2001/GA.HTM>> (16.05.2010).
- [65] "Genetik Algoritmalar", <http://tr.wikipedia.org/wiki/Genetik_algoritma> (15.05.2010).
- [66] "Genetic Algorithm", <http://en.wikipedia.org/wiki/Genetic_algorithm> (10.05.2010).
- [67] Weise, T.: *Global Optimization Algorithms-Theory and Application*.(2009)
- [68] Davarynejad, M.: "Fuzzy Fitness Granulation in Evolutionary Algorithms for complex optimization" *M.Sc. thesis*, Ferdowsi University of Mashhad Department of Electrical Engineering, Mashhad, Iran, (2007).
- [69] "Genetic Programming Notebook", <<http://www.geneticprogramming.com/>> (16.05.2010).
- [70] Beckers, R.; Goss, S.; Deneubourg, J. L. ; Pasteels, J.: "Colony size, communication, and ant foraging strategy". *Psyche* 96, (1989), 239-256.
- [71] Dorigo, M. ; Gambardella, L.: "Ant Colony System: A Cooperative Learning Approach To The Traveling Salesman Problem". *IEEE Transactions on Evolutionary Computation* 1, (1997), 53.

- [72] Dorigo, M.; Birattari, M. ; Stutzle, T.: "Ant Colony Optimization". *IEEE Computational Intelligence Magazine* 1, (2006), 28-39.
- [73] Özdemir, Y. S.: "Karınca Kolonisi Algoritması İle Bilgisayar Ağlarının Topolojik En İyilenmesi", Başkent Üniversitesi Fen Bilimleri Enstitüsü, Ankara, (2008).
- [74] Timur Keskinurk ; Söyler, H.: "Global Karınca Kolonisi Optimizasyonu". *Gazi Üniversitesi Mühendislik Ve Mimarlık Fakültesi Dergisi* 21, (2005).
- [75] Selvi, V. ; Umarani, R.: "Comparative Analysis of Ant Colony and Particle Swarm Optimization Techniques". *International Journal of Computer Applications IJCA* 5, (2010), 1-6.
- [76] Rook, C.: "Investigation Into The Use Of Ant Algorithms For The Travelling Salesman Problem And Game Playing". *MSc Computing 1999/2000*, (2000).
- [77] Blum, C.: "Ant Colony Optimization: Introduction And Recent Trends". *Physics of Life Reviews* 2, (2005), 353-373.
- [78] Anirudh, S.; Poddar, P. ; Boswal, D.: "Ant Colony Optimization Algorithms: Introduction and Beyond". *Artificial Intelligence Seminar*, (2009).
- [79] Dorigo, M. ; Stützle, T.: *Ant colony optimization*. the MIT Press,(2004)
- [80] Sanem Sariel ; Akin, L.: "Özerk Robotlar İçin Bir Geniş Alan Taraması Planlama Yöntemi ". *Havacılıkta İleri Teknolojiler ve Uygulamaları Sempozyumu*, (2004).
- [81] Göksu, S.: "Kendi Kendine Yol Bulan Araç". *TMMOB Elektrik Mühendisleri Odası Proje Yarışması*, (2006).
- [82] Goforth, M. D.: *Informed Searching Lecture Notes*. Laurentian Üniversitesi, Sudbury Canada,(2003)
- [83] Ruhrop, S. ; Schindelbauer, C.: "Competitive Time And Traffic Analysis Of Position-Based Routing Using A Cell Structure". (2005).
- [84] Peer, S.; Sharma, D. K. ; Sharma, R. K.: "Finding Shortest Path In The Presence Of Barriers: An Alternate Approach". *Applied mathematics and computation* 185, (2007), 333-345.
- [85] Rezende, P. J.; Lee, D. ; Wu, Y.: "Rectilinear shortest paths in the presence of rectangular barriers". *Discrete and Computational Geometry* 4, (1989), 41-53.
- [86] Atallah, M. ; Chen, D.: 270-279 (ACM).

ÖZGEÇMİŞ

1984 yılında Malatya'nın Yeşilyurt ilçesinde doğdu. İlkokulu İzmir'de Mustafa Urcan İlköğretim Okulunda, ortaokulu Vali Kutlu Aktaş İlköğretim Okulunda okudu. Lise öğrenimine Malatya Akçadağ Anadolu Öğretmen Lisesinde başlayıp hazırlık ve lise birinci sınıfı burada okuduktan sonra, İzmir Konak İnönü Lisesine geçiş yaptı.

2004 yılında Marmara Üniversitesi Teknik Eğitim Fakültesi Bilgisayar ve Kontrol Öğretmenliği bölümüne başladıktan sonra 2008 yılında bu bölümden mezun olarak aynı yıl Marmara Üniversitesi Fen bilimleri Enstitüsü Bilgisayar-Kontrol Eğitimi bölümünde yüksek lisans eğitimine başladı ve hala devam etmektedir.


T.C.
MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KABUL ve ONAY BELGESİ

Yunus PAÇACI'nın "GÜNCEL YAPAY ZEKA TEKNİKLERİNİN İNCELENMESİ VE EKSİKLİKLERİNİN GİDERİLMESİ İÇİN HYBRİD ALGORİTMA TASARIMI" başlıklı Lisansüstü tez çalışması, M.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 09.09.2011 tarih ve 2011/21-10 sayılı kararı ile oluşturulan jüri tarafından Elektronik Bilgisayar Eğitimi Anabilim Dalı Bilgisayar Kontrol Eğitimi Programında YÜKSEK LİSANS Tezi olarak Kabul edilmiştir.

Danışman : Doç.Dr.Hasan ERDAL
1. Üye : Prof.Dr.Ahmet Fevzi BABA
2. Üye : Doç.Dr.Haluk KÜÇÜK

Marmara Üniversitesi
Marmara Üniversitesi
Marmara Üniversitesi



Tezin Savunulduğu Tarih : 11/10/2011

ONAY

M.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 13.10.2011 tarih ve 2011/23-02 sayılı kararı ile Yunus PAÇACI'nın Elektronik Bilgisayar Eğitimi Anabilim Dalı Bilgisayar Kontrol Eğitimi Programında Y.Lisans (MSc.) derecesi alması onanmıştır.

Fen Bilimleri Enstitüsü Müdürü
Prof. Dr. Meral ÜNAL

