
Araştırma Makalesi / Research Article

Çoklu Hedeflerin Çoklu Robotlara Paylaştırılması İçin Bir Yük Dengeleme Sistemi

Emrah DÖNMEZ*, Adnan Fatih KOCAMAZ

*İnönü Üniversitesi, Bilgisayar Mühendisliği, Malatya
(ORCID: 0000-0003-3345-8344) (ORCID: 0000-0002-7729-8322)*

Öz

Günümüzde robotik alanında verilen görevi icra etme, yol planlama, kontrolör tasarımı gibi konularda yaygın olarak tek robot ile yapılan sistemlere odaklanılmaktadır. Çoklu robotlar ve çoklu hedef/görev paylaşımı üzerine ise daha az sayıda çalışmalar bulunmaktadır. Ancak bu alandaki yöntemlerin geliştirilmesi ve kolektif çalışma modelleri üzerine derinlemesine çalışmalar yapılması gerekmektedir. Bu çalışmada, birden fazla robot ile çok sayıda hedefin gezilmesi için görev paylaşımı ve yük dengeleme sistemi (YDS) geliştirilmiştir. Çalışma çok sayıda hedefin asgari maliyet ile gezilmesi bakımından çoklu gezgin satıcı (Çoklu-GSP) problemine de benzemektedir. Görev paylaşımı sisteminde YDS pasif veya aktif durumlarına göre görev dağılımları yapılmıştır. Yük dengelemede amaç bir robota gereğinden fazla görev verilmesinin önüne geçerek enerji ve maliyeti gözetmektir. İlgili robotlara görev dağılımı yapıldıktan sonra robot sayısı kadar hedef küme oluşturulur. Her bir küme için robot konumu ve mevcut hedefler birer çizge düğümü olarak kabul edilmiştir. Oluşan bu çizge düğümleri tam bağlantılı hale getirilerek mesafe matrisi oluşturulmuştur. Daha sonra yol planı başlangıç düğümü olan robotun ilk pozisyonundan hedef düğümlere en yakın komşu (NN) ve genetik algoritma (GA) yöntemleri ile yapılmıştır. Gidilen bir sonraki düğüm yeni başlangıç pozisyonu olarak kabul edilirken, gezilen her bir düğüm ise çizge bağlantı matrisi içerisinden çıkarılmıştır. Hedef ve robotlar renkli etiketler ile etiketlenmiş ve nesnelerin konumları renk tabanlı nicemleme ve eşikleme yöntemleri ile hesaplanmıştır. Yapılan deneyler sonucunda tasarlanan sistemin değişken sayıda ve/veya farklı hedef dağılımlarında iyi bir şekilde görev paylaşımı yaptığı ve elverişli yol planı oluşturduğu gözlemlenmiştir.

Anahtar kelimeler: Yük Dengeleme Sistemi, Çoklu Görev Paylaşımı, Yol Planlama.

A Load Balancing System for Sharing Multiple Targets to Multiple Robots

Abstract

Today in the field of robotics, single robot systems and topics such as fulfilling the given tasks, path planning and controller design are generally focused. There are fewer studies on multiple robots and multiple target/task sharing. However, it is necessary to develop methods in this field and to make in-depth studies on collective working models. In this study, task sharing and load balancing system (LBS) have been developed for traveling multiple robots with multiple targets. The study is similar to the problem of multi-travelling salesman problem (Multi-TSP) in terms of visiting multiple targets with minimum cost. In the task sharing system, tasks are distributed according to LBS passive or active status. The purpose of load balancing is to observe energy and cost by preventing over-assignment tasks to a robot. After the distribution of the task to the related robots, the target cluster is created up to the number of robots. For each cluster, the robot position and the current targets are considered as a graph node. A fully connected graph is created with these nodes and the distance matrix is formed. Then, from the initial position of the robot that was the starting node, the path plan to the target nodes has been composed by the nearest neighbor (NN) and the genetic algorithm (GA) methods. The next node is considered to be the new starting position, while each visited node is removed from the graph adjacency matrix. Target and robots are labeled with colored labels and the positions of the objects are calculated by color-based quantization and thresholding methods. As a result of the conducted experiments, it has been observed that the designed system has made an efficient task sharing and create a suitable path plan in variable number of targets and different target distributions.

Keywords: Load Balancing System, Multi Task Sharing, Path Planning.

*Sorumlu yazar: emrah.donmez@inonu.edu.tr
Geliş Tarihi: 05.10.2018, Kabul Tarihi: 12.03.2019

1. Giriş

Mobil robot kontrolü için günümüzde pek çok araştırma yapılmakta ve önemli sayıda yöntemler ve teoremler geliştirilmektedir. Geleceğin teknolojileri arasında gösterilen karar alma seyrüseferine sahip (otonom) araçların yaygınlaştırılması bakımından da mobil robot kontrolü önemli bir konudur. Çünkü her bir otonom araç temelinde otonom bir mobil robot ile benzer kavramlarla geliştirilir. Otonom olarak tasarlanan robot için belirlenen hedeflerdeki görevleri yapmak ve bu görevleri diğer otonom robotlarla iletişim halinde olarak yerine getirmek geleceğin mobil sistemlerinde en temel tasarım unsurları olacaktır. Otonom bir mobil sisteme sadece görev koordinatlarını ve görev listesini vererek belirlenen bu hedeflerdeki görevleri yerine getirmesi talimatı verilebilir. Bu bakımdan tasarlanan sistemin doğru, kararlı ve maliyet gözetilen bir yapıda çalışması gerekmektedir. Belirlenen hedefler görevlendirilen robotlara elverişli ve dengeli bir biçimde atanmalıdır. Hedeflerin robotlara olan uzaklık durumları bu denge durumunun göz ardı edilmesini de gerektirebilir. Tüm bu hususlar dikkate alındığında, aslında robotların mevcut konumları ve kendilerine atanan hedeflerin konumlarına göre bir yol planı oluşturulmalıdır.

Robotu bir başlangıç konumundan başka bir hedef konuma uygun bir yol planı üzerinden götürmek robotik alanında üzerine odaklanılan temel alanlardan biridir. Bu kapsamda tek robot kontrolü üzerine çalışmalar yapıldığı gibi kolektif hareket kabiliyetine sahip işbirlikçi, görev paylaşımı yapan çoklu robot kontrol sistemleri için de çalışmalar yapılmaktadır. Bu sistemler birden fazla görevin icra edilmesine yönelik olarak bir görev paylaşım tablosu ve görevlerin yapılacağı sıra bilgisini içeren bir görev sıralama listesi gibi modüler tasarımlar gerektirebilir. Bu işlemleri gerçekleştirmek için de konfigürasyon uzayının harita bilgisi, robotların mevcut pozisyonları, hedeflerin tespit edilmesi ve pozisyonları gibi ön-bilgilerin elde edilmesi gerekmektedir. Ön-bilgiler sisteme önceden verilebileceği gibi görüntü işleme, lazer tarama ve sensör bilgileri gibi haritalama yöntemleri ile de elde edilebilecek verilerdir. Ön-bilgiler sisteme önceden verilse de hareketli dinamik engeller için kontrol sistemlerini gerçek zamanlı çalışacak şekilde yeniden modellemek gerekir. Görüldüğü üzere, ortam değişkenliğine ve tasarlanan sisteme bağlı olarak kullanılacak ön-bilgi çıkarımı yöntemi de değişebilir. Bu çalışma kapsamında robot ve hedeflerin küresel konumları görüntü üzerindeki ayırt edici özelliklerin kullanılması yoluyla elde edilmiştir.

Çok sayıda hedefin robotlara paylaştırılmasından sonraki aşamada uygun maliyetle gezilmesi bir gezgin satıcı problemine (GSP) benzemektedir. GSP' de her bir hedefe tek bir kez uğrayarak gezinmeyi gerçekleştirmek istenir. Deterministik olmayan polinomal bir problem olması ve problem boyutundaki hedef sayısı arttıkça çözümün eksponansiyel yakın büyümesi sonucu GSP için en iyi çözümü bulmak zorlaşmaktadır. Diğer taraftan, robot tarafından çok sayıda hedefin gezilmesinde en önemli konular yolun elverişliliği ve maliyetidir. GSP' de en iyi çözüme yaklaşıldıkça yol maliyeti ve elverişliliği de daha iyi bir oranda iyileşecektir. Hangi hedeften gezinmeye başlanıldığı da ortaya çıkacak yol maliyetini etkileyen bir husustur. Asgari gezinme devinimi ile istenen tüm hedefleri gezmek; zaman, yol maliyeti, robotun enerji durumunun gözetilmesi vb. temel etkenlerde daha verimli performans değerlerinin ulaşılmasına olanak verecektir.

Bölüm 2'de ilişkili çalışmalar irdelenmiştir. Problem tanımı ve ön-bilgiye bölüm 3'te değinilmiştir. Konfigürasyon uzayı, kullanılan materyal ve metot bölüm 4'te verilmiştir. Bölüm 5'te deney sonuçları açıklanmış ve son bölümde ise sonuçlar ve gelecek çalışmalar verilmiştir.

2. İlişkili Çalışmalar

Dağıtık sistemlerin tarihsel süreci 1980'li yıllara dayanır. Bu tarihten önce, araştırmacılar tekli robotlar üzerinde çalışmışlardır [1]. Çoklu robotlar ile yapılan ilk çalışmalardan hücre robot sistemleri (CEBOT) Fukuda ve diğerleri [2] tarafından, çoklu robot hareket planlama ile trafik kontrol uygulaması amacıyla yürütülmüştür. Çalışmalarında hücre yapılarından esinlenerek kendi kendini organize edebilen kolektif robotlar için bir kontrol altyapısı geliştirmişlerdir. Premvuti ve Yuta tarafından [3], çoklu robot işbirliği mimarisi ile ilgili çalışmalar yürütülmüştür. Bu kapsamda, otonom çoklu robot sistemleri için işbirliği mimarisine yönelik geliştirmeler yaptılar. Asama ve diğerleri [4] ACTRESS olarak literatüre giren erken araştırmalar ile bu konuya ışık tutmuşlardır. Çalışmaları kapsamında dağıtık robotik sistemlerin otonom olarak birbirleriyle iletişim kurma, çevreyi algılama, görevleri yerine getirme gibi bileşenlerin tasarımına yönelik araştırmalar yaptılar. Araştırmacılar sonraki çalışmalarında ise işbirlikçi

yol planlama ve iletişim protokolü çoklu ajan sensör sistemler ile hataya karşı dayanıklı bir yapı [5] ve çoklu robot ile parçacık filtresi kullanılarak bilinmeyen alanda keşif uygulaması [6] geliştirmişlerdir. Holonomik olmayan tekerlekli çoklu robot sistemler için dağıtık algoritma çalışması [7], çoklu robot görev paylaşımını akıllı market yaklaşımı [8] gibi yaklaşımlar da geliştirilmiştir. Temizlik amaçlı geliştirilen robotlar, biyolojik esintili sinirsel ağ yapısını kullanarak engelden kaçan yapılar [9] ve yapay potansiyel alan tabanlı mobil sensör ağı ile komşuluk ilişkisi voronoi ile çözülmüştür [10]. Çayırpunar [11], yayınladığı “Çoklu Robot Sistemlerinde Robotlar Arası Haberleşme Ve İşbirliği Kullanılarak Arama Verimliliğinin Artırılması” yüksek lisans tezinde çoklu robot sistemlerinin kontrolü üzerine son yıllarda yoğun kuramsal çalışmalar yapılırken, deneysel ve pratik çalışmaların nispeten daha az yoğunlukta olduğunu belirtmiştir. Bununla birlikte kuramsal çalışmalarda çoğu zaman belirli varsayımlar üzerinden sonuç elde edildiği tespitini yapmıştır.

Deneysel çalışmalarda ise çoğu zaman araştırmacılar laboratuvarlarında var olan robotların özellikleri ile sınırlı kalmak durumunda kaldıklarını ve çoğu zaman bu özelliklerin kavramsal çalışmalardaki “ideal” koşullardan farklı olduğunu belirtmektedirler. Akıllı denetleme yöntemleri şeklinde günümüzde robotlar üzerinde uygulanmış pek çok yöntem bulunmaktadır. Bulanık mantık, sinir ağları, genetik ve melez yöntemler akıllı denetim sistemlerine verilebilecek örneklerdir. Tiemin Hu ve arkadaşları [12] yapay sinir ağları yöntemini kullanarak gezgin robot için hem kinematik hem de dinamik denetlemeden bileşenlerinden oluşan denetleyiciyi geliştirilmişlerdir. Gezgin robotun engelden kaçınması için sinir ağları ve bulanık mantık yöntemleri ile kontrol sistemi Wang ve Yang [13] tarafından geliştirilmiştir. Çalışmada holonomik olmayan robotun teker hızlarının farklı büyüklükte olmasını sağlayarak robot kontrolünü sağlamışlardır. Haritalama probleminde, keşif amaçlı kullanılan tek robot ile çoklu robotların keşif süresinin karşılaştırılması üzerine bir çalışma Burgard ve arkadaşları [14] tarafından geliştirilmiştir. Harvard Üniversitesinin robotik çalışanları [15], sürü zekâsı sergileyen canlıların davranışları modelleyerek 1024 adet mobil robot ve kontrol sistemini tasarlamış ve robotu “Kilobot” olarak isimlendirmiştir. Alman Fraunhofer Enstitüsü araştırmacıları [16] sürü zekâsı kullanarak birbirinin aynısı olan ve birbirlerinin işlerini kolaylıkla devralabilecek depo robotlarını işbirliği yapacak şekilde geliştirmişlerdir. Literatürde çoklu robot kapsama algoritmaları ile ilgili birçok deneysel çalışma yapılmıştır.

Dönmez ve diğerleri [17], [18] tekli robot ve tek hedef üzerinde yapay potansiyel yöntemi ile yol planı çıkarılması ve görü tabanlı kontrolör tasarımlarını gerçekleştirmişlerdir. Çalışmada kullandıkları yapay potansiyel alan metodundaki parametreleri ortam değişkenlerine göre yeniden ayarlayan bir uyarlamalı yöntemi önermektedirler. Vizyon tabanlı yol planlaması alanında; Dirik ve diğerleri [19], tepe kamera sistemi ile statik ortamlarda yapay potansiyel alan algoritması ile gezgin robot rota planlaması gerçekleştirmiş, güçlü dinamik performansından dolayı sensörsüz görü kontrol uygulaması önemine işaret etmişlerdir. Kocamaz ve diğerleri [20], farklı kombinasyon ve sayıdaki pinpon topu dizilimleri ve farklı mobil robot konumlamalarını tepe kamera ile görüntüleyerek, görüntü örnekleri elde etmiş, mobil robotun gezinme sırasında tüm topları en kısa yolu tercih ederek en optimum mesafeyi gezerek tüm topları toplayabilmesi için gezgin satıcı problemi için NN-GSP algoritması kullanmıştır. Dönmez ve diğerleri [21], tepe kameralardan elde edilen görüntüler üzerinde başlangıç ve hedef konum arasındaki en az maliyetli yol tespiti yaparak, robotun bir başlangıç konumundan hedef konuma ilerletilmesini sağlayacak Gaussian kontrolör yöntemi önermişlerdir. Dönmez ve Kocamaz [21] çoklu robotlarda hedef dağılımı üzerine bir yaklaşımı bildirmişlerdir.

Literatürde yapılan çalışmaların çoğunda robotlar bir nokta kütlesi olarak ele alınmış, dinamik ve kinematik kavramları ise ihmal edilmiştir. Diğer taraftan yapılan çalışmaların büyük bölümünde robotlar arası kolektif çalışmada atanacak iş yüklerinin paylaşılmasına yönelik optimize edilmiş herhangi bir algoritma önerilmemiştir. Kolektif çalışma sistemlerinde genellikle sürü zekâsı çalışmaları bulunmakta ve merkezi bir zekâ yönetimi üzerine detaylı bir çalışma bulunmamaktadır. Çalışmalarda genellikle yol maliyetine odaklanılmakta robotlara atanan iş yükü sayısında bir denge aranmamaktadır. Bugüne kadar pek çok çalışma çoklu robot işbirliği alanında yapılmıştır. Bu araştırmalar; koordinasyon, lokalizasyon, araştırma, haritalama, yol planlama gibi konular üzerine olmuştur. Ancak bu çalışmaların çoğu çoklu robot ve çoklu hedef konularında hedef paylaşımının gerçekleştirilmesine yönelik herhangi bir model önermemişlerdir. Bu çalışma kapsamında ise çok sayıda hedefi birden fazla robot arasında dengeli şekilde paylaşılacak bir yük dengeleme sistemi önerilmiştir. Bu sayede görevleri birlikte kolay, hızlı, etkili, düşük maliyetli bir şekilde yürütmeleri sağlanabilir. Sistem robotların hedeflere yakınlığı, hedef sayısı ve yol maliyeti parametrelerini gözeterek iş yükü dağılımını gerçekleştirmektedir.

3. Problem Tanımı ve Ön-Bilgi

Her bir hedefi gezmek yoluyla en az maliyetli hedefe ulaşmak için tüm olasılıkları denemek gerekecektir. Bu durumda problem bir kaba kuvvet (Brute Force) analizi ile muhtemel çözümleri incelemeyi gerektirir. Ancak gerçekte, kaba kuvvet analizi özellikle çok sayıda hedef için pratik ve verimsizdir. Problemin hesaplama karmaşıklığının $O(n!)$ seviyesinde genişlemesi bu duruma sebebiyet vermektedir. Örneğin; 7 eleman içeren bir GSP problemini tam olarak çözmek istiyorsak, çözümün hesaplama karmaşıklığı $7!$ ' e kadar çıkabilir (5040 farklı yol senaryosu). Bu hedefler grubuna sadece üç hedef daha eklersek, hesaplama karmaşıklığı yaklaşık 720 kat artabilir. Hedef sayısı arttıkça kesin çözümü bulmak daha da elverişsiz hale gelmektedir. Hedef kümesi içerisinde uygun bir hedeften başlayarak sürekli en yakın komşuya ilerlemek yoluyla problem çözülebilir. Ancak sürekli olarak en yakın komşuyu seçerek ilerlemek kaba kuvvet gibi kesin bir çözüm sunmayabilir. Diğer taraftan, genetik algoritma tasarımı ile daha elverişli çözümlere yaklaşmak mümkündür. Ancak genetik algoritma çözümü de iteratif olarak çalıştığından oldukça yavaş bir yöntemdir. İterasyon sayısını artırmak daha iyi çözümler sunsa da zaman ve maliyet parametrelerini olumsuz olarak etkileyecektir.

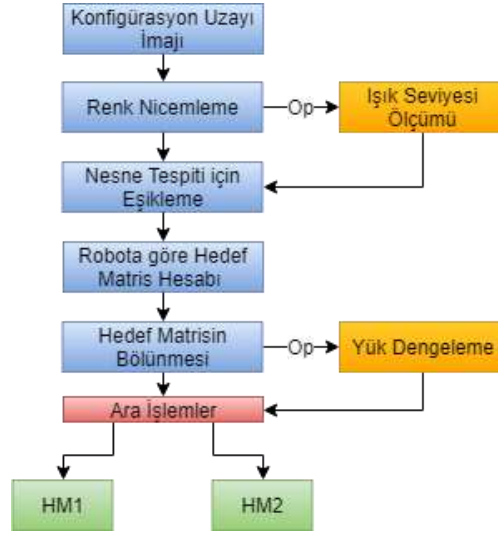
4. Materyal ve Metot

4.1. Konfigürasyon ortamı

Deney ortamında hedeflerin sayısı 8, 16 ve 24 olarak seçilmiştir, robotların sayısı ise iki olarak belirlenmiştir. Robotlar mevcut yön durumunun ve tekerlerin pozisyonunu gösterecek şekilde farklı renklerle etiketlenmiştir. Hedefler ise tek bir renk ile etiketlenmiştir. Robotların yönü, konumu ve hedeflerin konum bilgileri renk nicemleme ve renk eşikleme yöntemlerinin birlikte kullanılmasıyla bulunmaktadır. Renk nicemleme birbirine yakın tondaki renkleri tek bir sınıfa atarak renk derinlik değerini düşürmede kullanılmaktadır. Bu durumda daha az renkle eşikleme yapılması renk maskesinin daha kararlı çalışmasını sağlamaktadır. Renk tabanlı eşikleme nesne tespitini daha hızlı gerçekleştirmek için kullanılmıştır. Ortam aydınlanma değerlerindeki değişime karşı da ortam ışık değeri LDR sensörler aracılığıyla ölçülmekte ve sensörlerden gelen ortalama değerlere göre maske kanallarındaki eşik aralığı değiştirilmektedir. Bu eşik değeri ampirik olarak ayarlanmış ve sensör değerleri renk maskesindeki kanal değerlerine haritalandırılmıştır. Gerçek ortam görüntüsünü almada kullanılan kamera standart bir web kamerasıdır ve SVGA (800x600) çözünürlüğe ayarlanmıştır. Sistemin çalıştırıldığı bilgisayar ise İ3 işlemciye, 6GB Ram ve 5400RPM HDD sahiptir. Herhangi bir hızlandırıcı donanım (GPU vb.) veya yazılım (OpenMP vb.) kullanılmamıştır.

4.2. Sistem tasarımı

Sisteme girdi olarak gelen konfigürasyon uzayı görüntüsü üzerinde yer alan robot ve hedeflerin küresel pozisyonları renk nicemlemesi ve eşiklemesi yöntemleri ile belirlenmektedir. Belirlenen hedeflerin (veya çizge düğümlerinin) koordinatları bir matriste tutulmaktadır. Bu düğüm koordinatları ve robotları koordinatları ile robotlardan düğümlere olan mesafe vektörleri hesaplanır. Mesafe vektörleri ilgili hedef ID ve koordinat değerleri ile birlikte bir vektör matris tablosuna yerleştirilir. Sistemdeki robot sayısı kadar vektör matris tablosu hesaplanacaktır. İki robot için iki ayrı vektör matris tablosu oluşturulur. Daha sonra hedef sayısına göre oluşturulacak matris uzayının boyutları belirlenir. Robot konumları ve her bir vektör matris tablosundaki vektör büyüklüklerine göre her bir hedefin her iki robota olan uzaklıkları karşılaştırılır. Hedef hangi robota yakınsa o robota ait gezinme sınıfına atanır ve her iki vektör matris tablosundan ilgili ID değerine sahip hedef çıkarılır. Bu işlem YDS hem kapalıyken hem de açıkken tekrarlanmıştır. Böylece yük dengelemenin gezinme rotası üzerindeki etkisi incelenmiştir. Oluşacak yeni gezinme sınıfları YDS' nin açık/kapalı olması durumuna göre de farklı sayıda hedef elemanları içerebilir. Şekil 1' de sistemin her bir robot için hedef bilgilerini içeren matrislerin (HM) oluşturulması aşamasına kadar olan genel çalışma adımları verilmiştir. Bu hedef matrisleri her bir robot için gezinme işleminin uygulanacağı hedefleri içermektedir.



Şekil 1. Hedef bilgilerini tutan matrislerin oluşturulması

Ortam imgesi renk maskeleri kullanılarak eşiklemeye işlemine tabi tutulur. Bu renk maske değerleri dinamik veya durağan olarak belirlenebilir. Bu seçenek isteğe bağlı bir modül olarak sisteme entegre edilmiştir. Dinamik seçenekte, ortam ışık şiddetini tespit etmede kullanılan ve dört farklı örnekten alınan LDR çıktı değerlerine göre maske değerleri belirlenir. Durağan seçenekte ise varsayılan maske değerleri kullanılmaktadır.

Maske değerlerinin belirlenmesinden sonra renkli görüntü RGB' den HSV alanına dönüştürülür. HSV uzayındaki imgeye denklem (1) ve (2) uygulanarak imge uzayı renk derinlik skalası 16 bit derinlikten 6 bit derinliğe (65536 - 64) daraltılır. Bu işlem HSV uzayında 'Hue' kanalına göre yapılmaktadır.

$$E(d, q) = \int_c d(x, q(x)) d\mu = \sum \int_{c_j} d(x, q(x)) d\mu \geq \sum \int_{c_j} d(x, q_j) d\mu \quad (1)$$

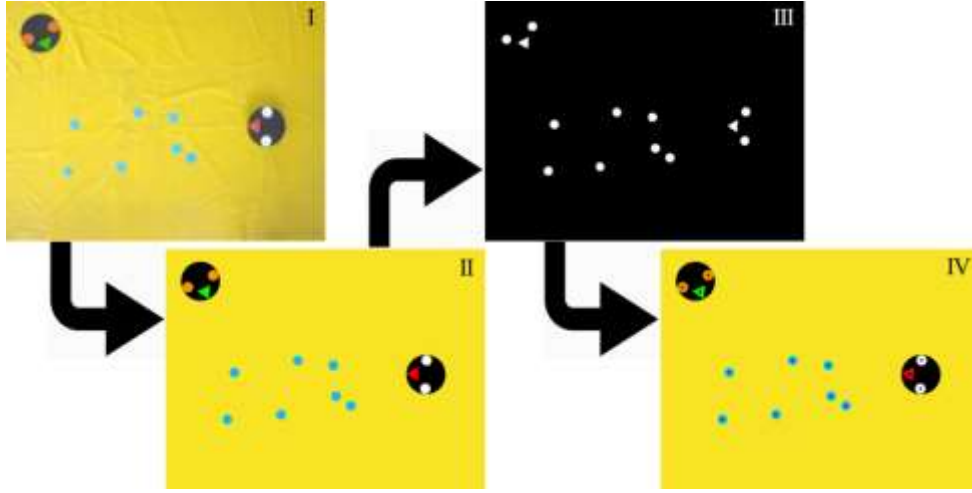
$$q_j = \arg \min_{c \in C, x \in C_j} E(d(x, c)) = \arg \min_{c \in C} \int_{c_j} d(x, c) d\mu \quad (2)$$

Q içerisinde $q_j, j = 1, \dots, n$ olarak nicemleme seviyeleri olduğu varsayılırsa, nicemleme hücre değerleri hesaplanabilir. C_i parametresi Q için Voronoi diyagramında hücreler olmak üzere, (1) ifadesi beklenen hatayı minimize etmede kullanılır. Bu eşitlikte d çarpıklık/distorsiyon parametresi q ise nicelleştiricidir. Girdi parametresini x değeri temsil etmektedir. C_j değerinin nicemleme hücreleri olduğu varsayılarak, q_j için en iyi değer (2) ifadesi ile bulunur.

Daha sonra ilgili nesnelere bölütleme için bir maske fonksiyonu uygulanır. Maske işlevi görüntünün Ton, Doygunluk ve Renk değerlerine bağlı olarak çalışır. Tüm görüntü matrisi taranır ve bu üç kanalda eşik değerleri olan her hücre '1' e dönüştürülür ve kalan hücreler '0'a dönüştürülür. Bu süreç "ikili görüntü elde etme" olarak bilinir. Maske fonksiyonunda tanımlanan kanal aralıklarını çıkarmak için Denklem (3) kullanılır.

$$f(p_k) = \frac{1}{M} \sum_{i=1}^M h(p_i) K_\sigma(p_k - p_i), \quad p_k \in \{H, S, V\} \quad (3)$$

Bu denklemde $h(p_i)$ renk kanallarının histogramını temsil eder. p_k kanal k^{th} seviyesi ve M kanalın toplam seviyesini ifade eder. Tanımlanmış eşik altında piksel alanı değerine sahip gürültülü alanlar varsa, bir çıkarma işlemi gerçekleştirilir ve bu gürültülü bölgeler '0' a dönüştürülür. Tespit edilen alanlar için merkez koordinat değerleri hesaplanır. Her bir merkez koordinat değerleri ilgili görüntü çerçevesinde saklanır ve işaretlenir. Örnek bir bileşen algılama işlemi aşağıda Şekil 2' te verilmiştir.



Şekil 2. Renk tabanlı bileşen algılama işlemi: (I) Gerçek ortam görüntüsü, (II) Nicemleme işlemi uygulanan görüntü, (III) Ortamın ikili harita görüntüsü, (IV) Tespit edilen bileşenler

4.3. Yük dengeleme sistemi (YDS)

Yük dengeleme (Load Balancing) ortamda bulunan hedefleri (iş yükleri) mevcut robotlar arasında dengeli ve etkili bir şekilde paylaşmak için kullanılmaktadır. Yük dengeleme ile amaç, robotların verilen hedef bölgelerini daha hızlı ve daha az maliyetle dolaşmalarını sağlamaktır. Önerilen yük dengeleme metodu iki kıstası göz önünde bulundurarak hedeflerin paylaşılması işlemini yürütmektedir. Bu kıstaslardan birincisi dengeli sayıda hedeflerin dağıtılmasıdır. Örneğin; ortamda bulunan iki robota '9' adet hedefin bir robota '4', diğerine '5' hedef düşecek şekilde paylaşılması verilebilir. Robotlara benzer oranlarda iş yüklerinin verilmesi amaçlanır. Bu kontrol için her bir hedefin robotlara olan mesafesi kontrol edilir ve ilk kıstas olan denge durumu gözetilerek yakınlık durumuna göre atama prosedürü gerçekleştirilir. Aşağıda Tablo 1'de yük dengeleme algoritmasının hedef parametrelerine göre uygulanan ilk aşaması verilmiştir. Bu işlem yol planı çıkarılmadan önce gerçekleştirilmektedir.

Tablo 1. Hedef sayılarına göre yük dengeleme algoritması

Girdi-1: Hedef konumları matrisi – KM
Girdi-2: Mesafe vektör matrisleri – $MR1, MR2$
Girdi-3: Hedef sayısı – HS
 Denge sınırı $DS: HS/2$
 Eğer $HS \leq 0$ ise,
BİTİR
 Eğer $HS > 0$ ise,
Döngü: $i=0; i < HS; i++$
 Eğer $MR1_i < MR2_i$ ise,
 Yeni KM Matrisi: $KM_{R1} \leftarrow KM_i$
 Eğer $MR1_i > MR2_i$ ise,
 Yeni KM Matrisi: $KM_{R2} \leftarrow KM_i$
 Değilse $MR1_i = MR2_i$ ise,
 Eğer $n(KM_{R1}) || n(KM_{R2}) \geq DS$ ise ve
 Eğer $n(KM_{Rx}) < n(KM_{Ry})$ $x, y \in \{1, 2\} \wedge x \neq y$
 $KM_{Rx} \leftarrow$ Kalan hedefler
Koşul Bitir
Döngü Bitir
Koşul Bitir
Çıktı: KM_{R1}, KM_{R2}

Hedef sayılarına göre YD algoritması ilk aşamada hedeflerin konum bilgisini içeren KM matrisini, her bir robotun tüm hedeflere olan mesafe verilerini içeren $MR1$ ve $MR2$ matrislerini ve hedef sayısı HS parametrelerini girdi olarak almıştır. Hedef sayısının (HS) bir ve üzeri olması durumunda, HS parametresi değeri kadar döngüye girilir. Her bir aşamada her bir hedefin robotlara olan mesafelerine bakılır ve hangi robota yakınsa hedef o robota atanır. Mesafe değerlerinde eşitlik olması durumunda ise hedef rastgele bir robota atanır. Hedeflerin en yakın mesafede olan robota atanması işlemi robotlardan birine atanan hedef sayısının hedef sayısının yarısı (DS) olmasına kadar devam eder. Bu değere ulaşıldığında ise kalan tüm hedefler atanan hedef sayısı az olan robota atanır. Böylece hedef sayısı dikkate alınarak robotlar arasında bir hedef atama işlemi gerçekleştirilir. Son durumda ise robotlardan birine diğerinden en fazla bir fazla, bir eksik ya da diğeriyle eşit sayıda hedef atanmış olur. KM_{R1} ve KM_{R2} yeni oluşturulan hedef konumları matrisleridir. $n(KM_{Rx})$ matrisin içerisinde yer alan hedef sayısıdır. x ve y parametreleri tam sayılardır.

YD için ikinci kıstas ise robotların gezinti mesafeleridir. YD algoritması bu kıstası yol maliyetinin en yakın komşu veya genetik algoritma ile çıkarılmasından sonra kontrol eder. Bu kapsamda dengeli sayıda dağıtılan hedefler üzerinde yol planı çıkarılmasının ardından, her bir robot için oluşan maliyetler kontrol edilir. Bu maliyete göre bir robota atanan iş yüklerinden bir ya da birkaçı diğer robota atanabilir. Aşağıda Tablo 2’de verilen algoritma bu işlemi yol planı çıkarıldıktan sonra gerçekleştirmektedir.

Tablo 2. Yol maliyetlerine göre yük dengeleme algoritması

Girdi-1: Paylaşılan hedef konumları matrisi – KM_{R1}, KM_{R2}
Girdi-2: R1 ve R2 için yol planı maliyetleri – YP_{R1}, YP_{R2}
Girdi-3: Mesafe vektör matrisleri – $MR1, MR2$
 Yol planı maliyetleri mutlak farkı – $YP_{MF} = |YP_{R1} - YP_{R2}|$
 Eğer $YP_{MF} < \min(YP_{R1}, YP_{R2}) * 0,20$ ise,
BİTİR
Döngü: $YP_{MF} \geq \min(YP_{R1}, YP_{R2}) * 0,20$ olduğu sürece,
 Eğer $\max(YP_{R1}, YP_{R2}) = YP_{R1}$ ise,
 $KM_{Rx} = KM_{R1} \wedge x = 1$
 $KM_{Ry} = KM_{R2} \wedge y = 2$
 $MR_x = MR1 \wedge MR_y = MR2$
 Eğer $\max(YP_{R1}, YP_{R2}) = YP_{R2}$ ise,
 $KM_{Rx} = KM_{R2} \wedge x = 2$
 $KM_{Ry} = KM_{R1} \wedge y = 1$
 $MR_x = MR2 \wedge MR_y = MR1$
Koşul Bitir
 $H_{Rx} = \max_d(KM_{Rx}),$
Yeni KM Matrisleri: $KM_{Ry} = KM_{Ry} \cup H_{Rx}, KM_{Rx} = KM_{Rx} \setminus H_{Rx}$
Yeni Yol Maliyeti: $YP_{Ry} = d(KM_{Ry}) \in \{MR_y\}, NN||GA$
Yeni Yol Maliyeti: $YP_{Rx} = d(KM_{Rx}) \in \{MR_x\}, NN||GA$
Döngü Bitir
Koşul Bitir
Çıktı: YP_{Ry}, YP_{Rx} ve KM_{Ry}, KM_{Rx}

Yol maliyetlerine göre YD algoritmasına girdi olarak paylaşılan hedef konumları matrisleri KM_{R1} ve KM_{R2} , her bir robot için yol maliyetleri YP_{R1} ve YP_{R2} ve her bir robotun tüm hedeflere olan mesafe verilerini içeren $MR1$ ve $MR2$ matrisleri verilmektedir. Eğer iki yol planı maliyeti arasındaki farkın mutlak değeri olan YP_{MF} en küçük yol maliyetinin %20 değerinden küçük ise algoritma sonlandırılır. Büyük veya eşit olması durumunda ise, yol maliyetlerinden büyük olana ait parametreler KM_{Rx} , MR_x ve x parametrelerine, küçük olana ait parametreler KM_{Ry} , MR_y ve y parametrelerine gelecek biçimde atamalar yapılır. Bu atama işlemlerinden sonra büyük olan yol planında robota en uzak düğüm değeri belirlenir ve H_{Rx} parametresine atanır. Bir sonraki adımda, yeni hedef konum matrisleri KM_{Ry} matrisine H_{Rx} hedefi dâhil edilir ve KM_{Rx} matrisinden ise H_{Rx} hedefi çıkarılır. Yeni yol maliyetleri ise eklenildiği KM_{Rx} ve KM_{Ry} hedefler arası mesafe matrislerinin GA ve NN ile yeniden hesaplanması ile bulunur. Bu hedef ekleme/çıkarma ve yeniden atama işlemleri yol maliyetleri

arasındaki eşik değere ulaşıncaya kadar devam eder. Bu eşik değere ulaşılmaması halinde algoritma sonlanır ve eşik değere yakın bulunan en yakın değer çözüm kabul edilir.

Çalışma kapsamında yük dengelemenin verimliliğini görebilmek için YDS modülü isteğe bağlı olarak çalıştırılabilen bir eklenti olarak sisteme entegre edilmiştir.

4.4. En yakın komşu yöntemi (NN)

En yakın komşu yöntemi basit bir şekilde yol oluşturmak için hedef düğümler arasındaki mesafeyi kullanır. Başlangıç konum düğümü algoritma tarafından veya dışarıdan seçimle belirlenebilir. Bir başlangıç konumu seçtikten sonra bu düğüm ile diğer hedef düğümler arasındaki mesafeler hesaplanır. Bu mesafe hesaplaması (**Hata! Başvuru kaynağı bulunamadı.**)' de verilen Öklid mesafe hesabı denklemi kullanılarak yapılır.

$$\text{Öklid} = \sqrt{\sum_{i=1}^k (x_i - y_i)^2} \quad (4)$$

Bir sonraki adımda, başlangıç düğümünden diğerlerine olan tüm mesafeler karşılaştırılır ve başlangıç noktasından en küçük mesafe değerine sahip düğüm, ziyaret edilecek ikinci hedef (nokta) olarak seçilir. Ziyaret edilen bu yeni düğüm başlangıç düğümü olarak yeniden atanır. Bir önceki gelinen düğüm ise gezinti matrisinden çıkarılır. Önceki tüm adımlar, bu yeni düğüm için tekrarlanır. Bu yol çıkarma işlemi, veri kümesindeki tüm değişkenler aynı şekilde işlenene kadar devam eder. Tüm düğüm ziyaretleri tamamlandığında, tüm düğüm noktalarını ziyaret eden bir yol elde edilir. Elde edilen bu yol en az maliyet yol olmayabilir, ancak en yakın komşu yöntemi genellikle GSP' de kabul edilebilir düzeyde maliyetli yolu üretir. Bu çalışma kapsamında en yakın komşu yöntemi varsayılan haliyle kullanılmıştır. Şekil 3' te NN yöntemi için çalışma diyagramı verilmiştir.

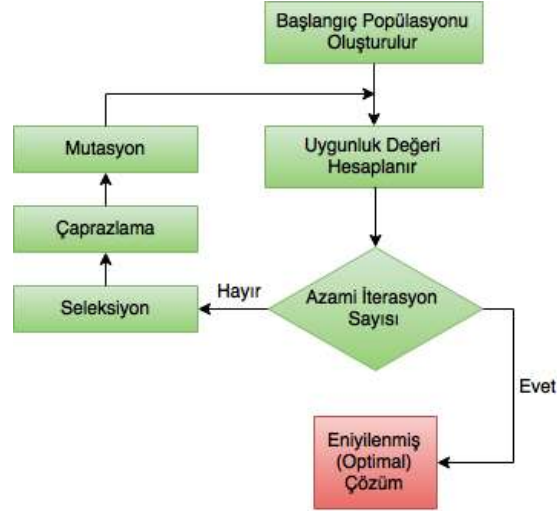


Şekil 3. En yakın komşu (NN) çalışma diyagramı

4.5. Genetik algoritma yöntemi (GA)

Genetik algoritmalar, doğadan esinlenerek oluşturulan ve hedef veri kümesi üzerinde bir arama ve en iyileme işlemi yürüterek en iyi çözüme yaklaştırma yapan bir yöntemdir. Arama işlemini çok boyutlu ve karmaşık uzayda yürütürken en iyi olanın hayatta kalması ilkesine göre bütünsel çözüme ulaşmaya çalışır. Genetik algoritma probleme tek bir çözümden farklı olarak bir çözümler kümesi üretir. Doğa esinli olarak çalıştığından seleksiyon, çaprazlama ve mutasyon adımları ile veri popülasyonunu daha da iyileştirmeye çalışır. Belirlenen azami iterasyon sayısına ulaşıldığında algoritma durdurulur. Yöntem uygunluk fonksiyonu üzerinden o anki durum için en iyi seçimi yapmaya çalışır. Geleneksel en iyileme yöntemlerinin iyi veya beklenen sonucu vermediği durumlarda etkili, verimli ve kullanışlı çözümler üretir. Şekil 4'de algoritmanın çalışma şeması gösterilmiştir. Gezgin satıcı probleminin çözümü için yaygın olarak kullanılır. Uygunluk değeri hesabında ise bu nesne düğümleri arasındaki uzaklık değerleri

temel alınarak hesaplama yapılmaktadır. Azami iterasyon için testler sonucunda bir üst sınır ampirik olarak belirlenmiştir.



Şekil 4. Genetik Algoritma (GA) çalışma diyagramı

5. Bulgular ve Tartışma

Önerilen sistemin performans değerlerini görmek için 8, 16 ve 24 adet hedef içeren toplamda üç farklı sayıda hedef içeren konfigürasyon uzayı kullanılmıştır. Sistemde iki robot (R1 – Kırmızı, R2 – Yeşil) görev paylaşımını yapmaktadır. Robot pozisyonları ise robotlar karşılıklı olacak şekilde ayarlanmıştır. Hedeflerin dağılımı rastgele (R), bir tarafa yığılı (Y) ve toplu (T) olarak üç biçimde ayarlanmıştır. Toplamda 9 farklı konfigürasyon uzayı düzeninde deneyler gerçekleştirilmiştir.

Görev paylaşımında hedeflerin robotlara olan yakınlığı gözetilmiştir. Bu durumda yük dengelemenin göz ardı edildiği söylenebilir. YDS aktifleştirildiğinde ise robotların izledikleri yolların toplam mesafe değerlerinin birbirine yaklaştırılması sağlanmaktadır. Bazı çalışmalarda hedeflerin dengeli sayılarda robotlara dağıtılması yoluyla yük dengeleme sağlanmaya çalışılmaktadır. Ancak sadece hedef sayısına bakmak tek tarafa yığılma olması gibi durumlarda robotların izledikleri yol maliyetlerinin birbirinden oldukça farklılaşmasına yol açacağı bilinmektedir. Bu yüzden bu çalışma kapsamında yük dengeleme için hem hedef sayısı hem de yol maliyeti kriterleri göz önünde bulundurulmaktadır.

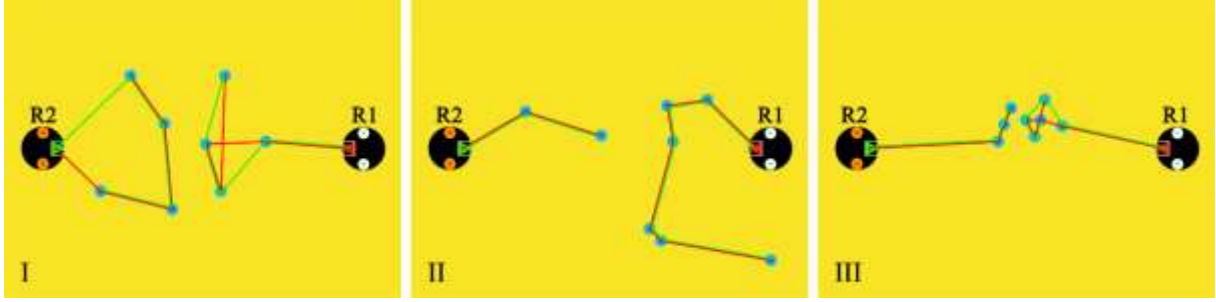
Deney ortamları 8 hedef içeren konfigürasyon için Şekil 5’te gösterildiği gibidir. Şekilde I numaralı kısımda rastgele dağılım, II numaralı kısımda sağa yığılı dağılım ve III numaralı kısımda ise toplu dağılım görülmektedir.



Şekil 5. Farklı konumlardaki 8 hedefin farklı dağılım konfigürasyonları

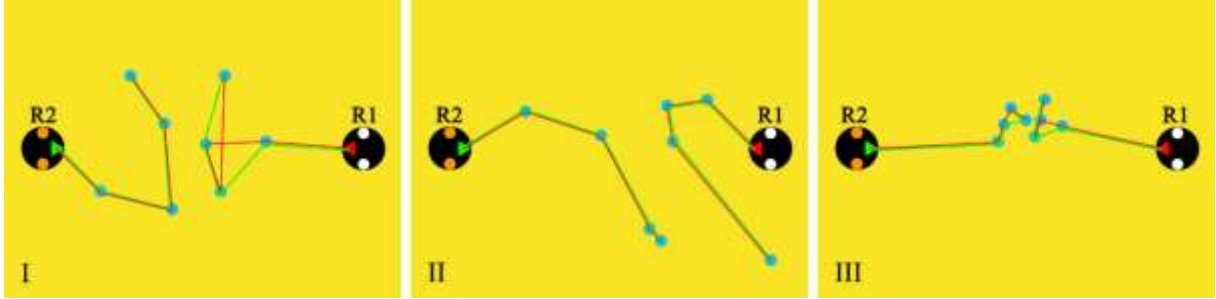
En yakın komşu ve genetik algoritma yöntemleri ile 8 hedef için elde edilen yol planları Şekil 6’da verilmiştir. Kırmızı ile gösterilen yol planı en yakın komşu yöntemi ile elde edilirken, mavi ile gösterilen yol planı genetik algoritma yöntemi ile elde edilmiştir. Elde edilen yol planlarına incelendiğinde genetik algoritmanın en yakın komşu ile benzer yol planları oluşturduğu bazı bölümlerde ise farklılaştığı gözlemlenmiştir. Bu farklılık yol maliyetlerinde de farklılaşmaya neden olmaktadır. Bu

durum yöntemlerin bir sonraki gidilecek hedefi değerlendirme süreçlerinin farklı olmasından kaynaklanmaktadır. Diğer taraftan, 8 hedef konfigürasyonlarında GA için iterasyon sayısı 30 olarak ayarlanmıştır.



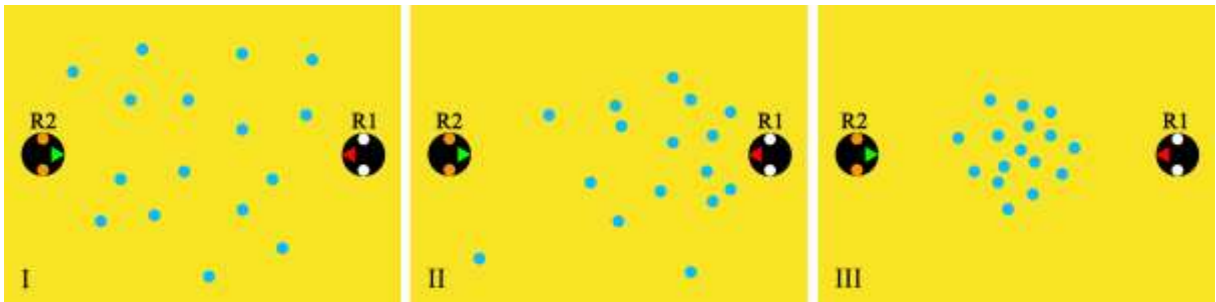
Şekil 6. NN (kırmızı) ve GA (mavi) yöntemleriyle 8 hedef için elde edilen yol planları

Şekil 7’de YDS açıkken oluşturulan yol planları görülmektedir. Görüldüğü üzere tüm dağılımlarda her bir robota eşit sayıda hedef atanmıştır. Bu atamadan sonra NN ve GA ile hedefler arası gezinti planı çıkarılmıştır. Sonraki aşamada yol planı uzunlukları arasında eşik uzunluk farkı kadar veya daha fazla fark olması durumunda algoritmanın ikinci kısmı çalıştırılmıştır. Bu dağılımlarda bu eşik değeri aşılmadığından yol planları verimli kabul edilmiştir.



Şekil 7. ‘8’ hedef için elde edilen yol planları (YDS açık)

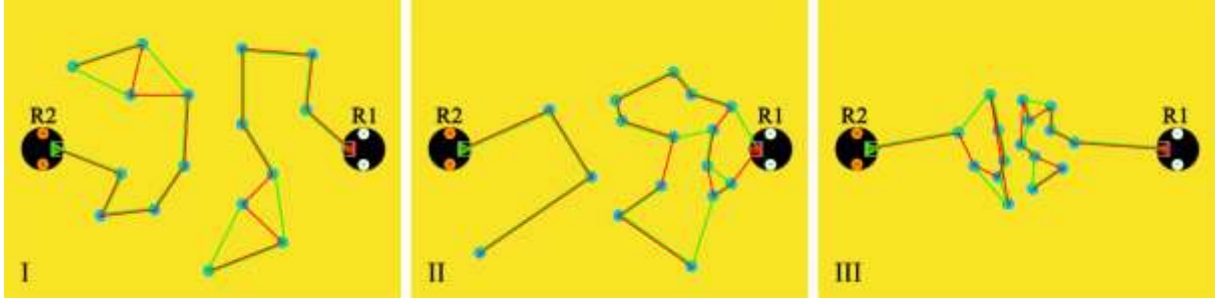
Deney ortamları 16 hedef içeren konfigürasyon için Şekil 8’de gösterildiği gibidir. Şekilde I numaralı kısımda rastgele dağılım, II numaralı kısımda sağa yığılı dağılım ve III numaralı kısımda ise toplu dağılım görülmektedir.



Şekil 8. Farklı konumlardaki 16 hedefin farklı dağılım konfigürasyonları

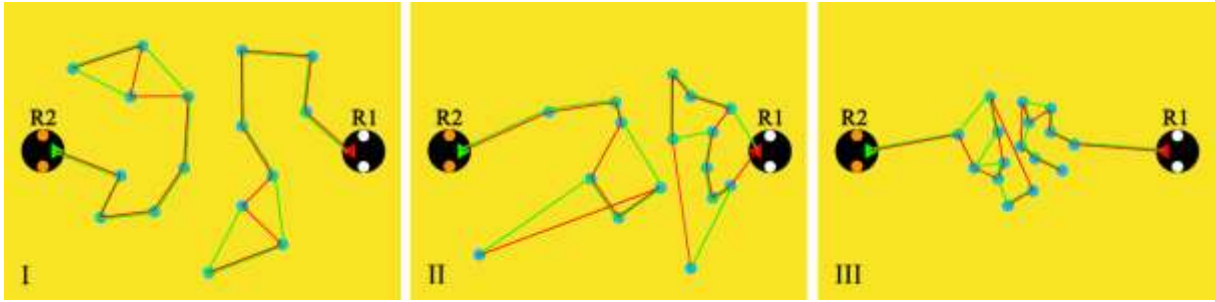
En yakın komşu ve genetik algoritma yöntemleri ile 16 hedef için elde edilen yol planları Şekil 9’da verilmiştir. Kırmızı ile gösterilen yol planı en yakın komşu yöntemi ile elde edilirken, mavi ile gösterilen yol planı genetik algoritma yöntemi ile elde edilmiştir. Yöntemler çoğu bölümde benzer yol örüntüsü çıkarmışlardır. (I) ve (III) numaralı dağılım incelendiğinde YDS kapalı olsa da eğer dağılım homojen ya da homojene yakın bir karakteristiğe sahipse, robotlara atanan hedef sayıları birbirine yakın çıkabilmektedir. Bu deneyde bir tarafa yığılım (II) ortamında R2 robotuna 3 hedef atanmışken R1 robotuna 13 hedef atanmıştır. Bu durum dengesiz sayıda görev atanması ve sistem için maliyetin daha

yüksek çıkması anlamına gelir. Diğer taraftan, 16 hedef konfigürasyonlarında GA için iterasyon sayısı 45 olarak ayarlanmıştır.



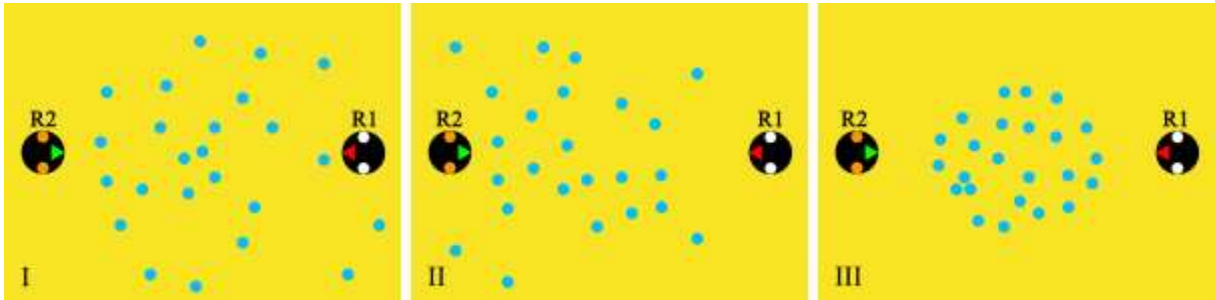
Şekil 9. NN (kırmızı) ve GA (mavi) yöntemleriyle 16 hedef için elde edilen yol planları

Şekil 10'da YDS aktifleştirildiğinde elde edilen düğümler arası gezinti planları verilmiştir. Dağılım (I)'de herhangi bir değişiklik gözlenmemiştir. Dağılım (II)'de R2'ye 7 hedef atanmışken, R1'e 9 hedef atanmıştır. Bu durumun temel nedeni yığılımın R1 robotuna doğru olmasıdır. Yük dengeleme algoritması robota atanan hedeflerin gezinti maliyetleri arasındaki farkın eşik değerini geçmesi sebebiyle R2'ye uzak olan hedefi R1'e atamıştır. Yeni atamalar üzerinde NN ve GA yeniden çalıştırılmış ve yeni yol planları elde edilmiştir. Yol maliyetleri arası farkın eşik değeri altına inmesiyle algoritma sonlandırılmıştır.



Şekil 10. '16' hedef için elde edilen yol planları (YDS açık)

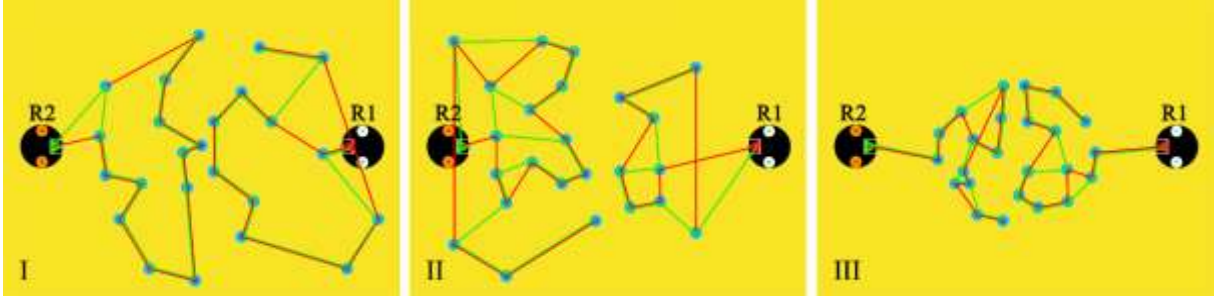
Deney ortamları 24 hedef içeren konfigürasyon için Şekil 11'de gösterildiği gibidir. Şekilde I numaralı kısımda rastgele dağılım, II numaralı kısımda sağa yığılı dağılım ve III numaralı kısımda ise toplu dağılım görülmektedir.



Şekil 11. Farklı konumlardaki 24 hedefin farklı dağılım konfigürasyonları

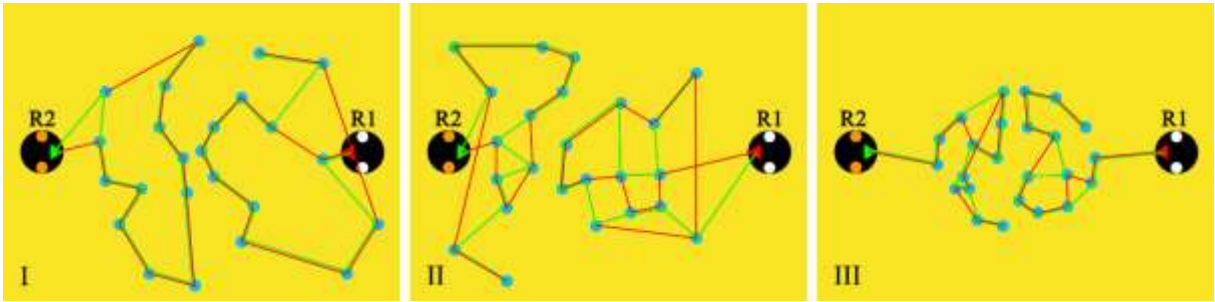
En yakın komşu ve genetik algoritma yöntemleri ile 24 hedef için elde edilen yol planları Şekil 12'te verilmiştir. Kırmızı ile gösterilen yol planı en yakın komşu yöntemi ile elde edilirken, mavi ile gösterilen yol planı genetik algoritma yöntemi ile elde edilmiştir. Yine benzer şekilde elde edilen yolların benzerlik oranı yüksek olmakla beraber farklılık gösterdiği bölümler de bulunmaktadır. GA iterasyon sayısı 24 hedef için 60 olarak ayarlanmıştır. Hedef sayısı arttıkça ihtiyaç duyulan asgari iterasyon sayısı da artmaktadır. İterasyon sayısının fazla artırılması GA yönteminde daha iyi sonuçlara ulaşılmasını sağlayabilir ancak algoritmik performans düşecektir. Dağılımın homojen

karakteristik gösterdiği (I) ve (III) numaralı deneylerde yine benzer ya da yakın sayıda hedeflerin R1 ve R2 robotlarına atandığı görülmektedir. Diğer taraftan sol tarafa yığılım olan (II) numaralı deneyde ise R1 robotuna 8 hedef atanmışken, R2 robotuna bu değer iki katı olan 16 hedef atanmıştır. Bu durum robotlar arasında dengesiz bir yük dağılımı ortaya çıkmasına neden olmaktadır.



Şekil 12. NN (kırmızı) ve GA (mavi) yöntemleriyle 24 hedef için elde edilen yol planları

Şekil 13'te YDS algoritmasının çalıştırılmasıyla (I) numaralı dağılımda R1 robotuna atanan sadece bir hedefin atanması R2 robotuna yeniden yapılmıştır. (II) numaralı deneyde ise hedef sayısına ve maliyete göre dağılım yeniden gerçekleştirilmiştir. Robotlara daha dengeli bir hedef atanması gerçekleştirilmiştir. (III) numaralı deneyde ise hedeflerin hâlihazırda dengeli sayıda atanması ve yol maliyetleri arasındaki farkı eşik değeri altında olması nedeniyle herhangi bir değişiklik uygulanmamıştır.



Şekil 13. '24' hedef için elde edilen yol planları (YDS açık)

Yapılan deneylerde yük dengeleme ile hedef sayısı ve hedeflerin robotlara olan yakınlık durumları göz önüne alınarak hedef görevlerin robotlara paylaşılması elde edilen yolların çakışmamasını da sağlamaktadır. Bu durum ise robotların birbirini bekleme, rahatsız etme gibi olumsuz durumları en aza indirmektedir.

Tablo 3'de R1 robotu,

Tablo 4'te ise R2 robotu için NN ve GA yöntemleriyle YDS kapalı iken elde edilen yol maliyetleri tüm hedef sayıları ve dağılım konfigürasyonlarında verilmiştir. Tablo verileri robotların bireysel iş yüklerini görmek açısından temel veriler sağlamaktadır. GA yöntemi genellikle birkaç konfigürasyon haricinde NN yönteminden daha az maliyetle yolları oluşturmuştur. Diğer taraftan NN ve GA yönteminin aynı maliyetle oluşturduğu yol planları da bulunmaktadır. Hedef sayısı arttıkça genellikle yol maliyetleri her iki yöntemde de artmaktadır. Hedeflerin dağılımlarına göre ise rastgele dağılımda maliyetin daha yüksek olduğu toplu durumlarda ise maliyetin daha düşük olduğu gözlemlenmiştir.

Tablo 3. R1 için deneylerde elde edilen yol maliyetleri (px) – YDS kapalı

Deney Adı	Dağılımlar		
	R	Y	T

	NN	GA	NN	GA	NN	GA
8 Hedef	624	593	734	734	401	395
16 Hedef	998	1027	1129	1092	635	602
24 Hedef	1428	1342	1098	985	837	842

Tablo 4. R2 için deneylerde elde edilen yol maliyetleri (px) – YDS kapalı

Deney Adı	Dağılımlar					
	R		Y		T	
	NN	GA	NN	GA	NN	GA
8 Hedef	565	588	305	305	362	349
16 Hedef	961	986	622	622	701	621
24 Hedef	1349	1318	1820	1746	878	866

Tablo 5’de R1 robotu, Tablo 6’da ise R2 robotu için NN ve GA yöntemleriyle YDS açık iken elde edilen yol maliyetleri tüm hedef sayı ve dağılım düzenlerinde verilmiştir. Benzer şekilde yük dengeleme aktif durumdayken de GA yöntemi NN yöntemine göre daha iyi sonuçlar vermiştir. Yol mesafeleri arasındaki fark, YDS ile daha da azaltılmıştır. Kalın metin ile verilen sonuçlar yük dengeleme kapalı durumda iken elde edilen sonuçlara göre daha iyi sonuçlar elde edildiği anlamına gelmektedir. YDS’ nin birkaç durum haricinde görev paylaşımı için genellikle daha iyi sonuçlar verdiği söylenebilir.

Tablo 5. R1 için deneylerde elde edilen yol maliyetleri (px) – YDS açık

Deney Adı	Dağılımlar					
	R		Y		T	
	NN	GA	NN	GA	NN	GA
8 Hedef	624	593	613	613	361	352
16 Hedef	998	1027	882	823	562	529
24 Hedef	1436	1302	1495	1276	843	856

Tablo 6. R2 için deneylerde elde edilen yol maliyetleri (px) – YDS açık

Deney Adı	Dağılımlar					
	R		Y		T	
	NN	GA	NN	GA	NN	GA
8 Hedef	565	565	546	546	352	344
16 Hedef	961	986	1041	982	672	588
24 Hedef	1268	1218	1369	1209	798	781

Tablo 7’de her bir konfigürasyonda oluşan toplam iş yükü (maliyet) gösterilmektedir. Toplam iş yükleri incelendiğinde GA yönteminin çoğu durumda NN yönteminden daha iyi sonuçlar verdiği anlaşılmaktadır. Toplam maliyetlerde NN yöntemi GA ile eşitlik gösterdiği durumlar da mevcuttur. GA yöntemi NN yöntemi ile karşılaştırıldığında yol maliyetlerinde %0 ile %13,24 arasında değişen oranlarda iyileştirmeler sağlamıştır. Diğer taraftan yük dengelemenin toplam yol maliyeti üzerindeki etkisi incelendiğinde, dağılımın bir tarafa yığılı (Y) olduğu 8 ve 16 hedef deneyleri hariç diğer tüm durumlarda iyileştirmeler sağladığı gözlemlenmiştir. Toplam yol maliyetinde iyileştirme sağlanamayan bu iki durumda, arka planda robotlara benzer maliyetlerde iş yükü verilmiş ve kayda değer iyileştirmeler sağlanmıştır. Bu sonuçlar hedef sayısının arttıkça yük dengelemenin daha iyi sonuçlar verebileceğini ortaya koymaktadır. Diğer taraftan robotlara benzer oranlarda iş yüklerinin verilmesi ile zaman ve enerji hususlarında daha verimli bir çalışma altyapısı sağlanmıştır. Koyu renkli değerler Genetik algoritmanın her bir hedef sayı ve dağılımda YDS açık veya kapalıyken üstün olduğu durumları belirtmektedir.

Tablo 7. Robotların her bir konfigürasyondaki toplam iş yükleri (toplam maliyet)

Deney Adı	Dağılımlar					
-----------	------------	--	--	--	--	--

	R		Y		T	
	NN	GA	NN	GA	NN	GA
8 Hedef YDK	1189	1181	1039	1039	732	726
8 Hedef YDA	1189	1158	1159	1159	713	696
16 Hedef YDK	1959	2013	1751	1714	1336	1223
16 Hedef YDA	1959	2013	1923	1805	1234	1117
24 Hedef YDK	2777	2660	2918	2731	1715	1708
24 Hedef YDA	2704	2520	2864	2485	1638	1637

Tablo 8’de yöntemler tarafından oluşturulan yol planı elde etme süre ortalamaları verilmektedir. NN yöntemi tüm konfigürasyonlarda yüksek hızda çalışmıştır. Hedef sayısının az olduğu durumlarda performans farkı 5-6 kat arasında gerçekleşirken, hedef sayısının artmasıyla bu fark 8 kata kadar çıkmaktadır. NN yönteminin daha hızlı olmasının temel nedeni; yöntemin sadece bağlılık matrisi ve mesafe matrisi verilerini kullanarak yol planını oluşturmasındandır. GA yöntemi ise iterasyon sayısına bağlı olarak her bir adımda ulaştığı çözümün en iyi çözüm olup olmadığını kontrol ederek ilerlemektedir. GA yöntemi bu kontrol işlemi için uygunluk fonksiyonu ile çözüm maliyetleri mevcut çözüm ve yeni elde edilen çözüm arasında karşılaştırma yapar. Bu durum ise yöntemin daha çok hedef içeren durumlarda oldukça yavaşlamasına, hatta iterasyon sayısının yetersiz kalması durumunda, elverişsiz çözümlerin ortaya çıkmasına neden olacaktır. YD sistemi ek olarak sisteme sadece ortalama %2,94 performans kaybı yaşatmıştır.

Tablo 8. Ortalama yol planı elde etme süreleri

Deney Adı	Dağılımlar					
	R		Y		T	
	NN	GA	NN	GA	NN	GA
8 Hedef	0,332s	1,511s	0,314s	1,554s	0,288s	1,442s
16 Hedef	0,345s	1,963s	0,330s	1,955s	0,308s	1,828s
24 Hedef	0,371s	2,588s	0,349s	2,706s	0,327s	2,579s

Görevlerin dağıtılmasında hedef sayısının yanında bu hedefler arası oluşan toplam mesafe ve her bir hedefin robotlara olan mesafe değerleri de yük dengeleme için ölçüt olarak alınmaktadır. Dolayısıyla hem hedef sayısı hem de toplam maliyet bakımından optimum nokta yakalanmaya çalışılmaktadır. YDS’ nin deneylerin çoğunda iyileştirmeler sağladığı ve robotlara dengeli bir görev paylaşımı yaptığı görülmektedir. Yük dengeleme algoritmasında yapılacak iyileştirmeler, farklı GSP yaklaşımları ve sezgisel yaklaşımlar ile toplam performansta daha fazla iyileştirmeler yapılabileceği öngörülmektedir.

6. Sonuç ve Öneriler

Bu çalışmada rastgele, bir tarafa yığılı ve toplu dağılım gösteren farklı sayılardaki hedefler üzerinde robotlara görev paylaşımı, dengeli yük dağıtımı, yol planı çıkarılması ve çoklu GSP problemine yönelik yöntemler irdelenmiş ve gerekli tasarımlar gerçekleştirilmiştir. Bu çalışmanın ana odak noktası olan YDS ile hedeflerin hem sayı hem de yol maliyeti bakımından verimli bir şekilde robotlara ataması yapılmıştır. Elde edilen sonuçlar YDS’nin çok farklı senaryolarda varsayılan yol planı sonuçlarını iyileştirdiği gözlemlenmiştir. YDS çoklu hedefler ve birden fazla robot üzerinde görev paylaşımı sağlaması açısından verimli sonuçlar üretmiştir. Deneylerden elde edilen diğer gözlemlere göre maliyet açısından GA daha başarılı performans sergilemiştir. Hız konusunda ise NN yöntemi GA yöntemine göre çok daha iyi bir performans göstermiştir. Bu durumun temel nedeni GA yönteminin iterasyon sayısına bağlı olarak elde ettiği çözümleri her adımda daha da iyileştirmeye çalışmasından kaynaklanmaktadır. Her iki yöntem de kullanım alanlarına ve ihtiyaçlara göre avantaj ve dezavantajlar gösterecektir. Toplamda 9 farklı deney 3 farklı hedef sayısında ve 3 farklı dağılımda gerçekleştirilmiştir.

Yürütülen deneylerin her biri üç kere aynı konfigürasyonda tekrar edilmiştir. Bu tekrarlar ışık yoğunluğunun fazla değişken olduğu durumlarda renk maskelerinin hatalı tespit yaptığı gözlemlenmiştir. Bu duruma çözüm olarak ise uyarlanabilir maske tasarımı tasarlanması veya derin öğrenme gibi makine öğrenmesi içeren yöntemlerin kullanılması önerilmektedir. Çalışma kapsamında temel amaçlar olan hedeflerin birden fazla robota dağıtılarak yol planlarının oluşturulması görevleri

başarı ile yerine getirilmiştir. Yürütülen deneylerde ışık etkisi ve robotla bağlantı kesilmesi olumsuz durumları göz önüne alındığında gerçekleştirilen toplam 27 deneyden (tekrarlar dahil) 24'ü başarı ile tamamlanmıştır. Bu olumsuz durumlar hesaba katıldığında başarı %85,18 olmuştur. Robotla bağlantı kesilmesi iki kez yaşanmıştır. Bu kesintiler göz önüne alınmadığında ise başarı %96,29 civarında olmuştur. Diğer taraftan önerilen YDS deneylerin çoğunda başarılı sonuçlar üretmiş ve yol planlama yöntemleri arasında ise en iyi sonuçlar GA yöntemi ile elde edilirken en hızlı sonuçlar ise NN yöntemi ile elde edilmiştir. Yük dengeleme sistemi, deneysel analizler ve çalışma ortamı parametreleri detaylı olarak incelenmiştir. Gelecek çalışmada, yük dengeleme yönteminin daha da iyileştirilerek, robotların sayısının artırılması ve özgün bir yol planlama yöntemi geliştirilmesi planlanmaktadır.

Teşekkür

Bu çalışma TÜBİTAK116E568 numaralı proje kapsamında ortaya çıkarılmıştır. Çalışma kapsamında İnönü Üniversitesi Robotik ve Otonom Sistemler Ekibi (ROSE) tarafından verilen teknik ve çalışma ortamı destekleri için teşekkürü bir borç biliriz.

Kaynaklar

- [1] Siciliano B., Khatib O. 2008. Handbook of Robotics. 1st ed. Springer-Verlag, Berlin.
- [2] Fukuda T., Nakagawa S., Kawauchi Y., Buss M. 1989. Structure decision method for self-organizing robots based on cell structures. CEBOT, International Conference on Robotics and Automation Proceedings, C.2: 695-700.
- [3] Premvuti S., Yuta S. 1990. Consideration on the cooperation of multiple autonomous mobile robots. IEEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications, C.1: 59-63.
- [4] Asama H., Matsumoto A., Ishida Y. 1989. Design of an Autonomous and Distributed Robot System: Actress. IEEE/RSJ International Workshop on Intelligent Robots and Systems '89. The Autonomous Mobile Robots and Its Applications. IROS '89. Proceedings, 283-90.
- [5] Asama H., Habib M. K., Endo I., Ozaki K., Matsumoto A., Ishida Y. 1991. Functional distribution among multiple mobile robots in an autonomous and decentralized robot system. IEEE International Conference on Robotics and Automation Proceedings, C.3: 1921-26.
- [6] Asama H., Ozaki K., Ishida Y., Yokota K., Matsumoto A., Kaetsu H., Endo I. 1994. Collaborative team organization using communication in a decentralized robotic system. Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems '94., IROS '94, C.2. (2): 816-23.
- [7] Kanayama Y., Kimura Y., Miyazaki F., Noguchi T. 1991. A stable tracking control method for a non-holonomic mobile robot. Proceedings IROS '91: IEEE/RSJ International Workshop on Intelligent Robots and Systems '91, Osaka, Japan, 3: 1236-1241.
- [8] Li C.S., Mamat R., Braunl T. 2009. Market-based approach for multi-team robot cooperation. 2009 4th International Conference on Autonomous Robots and Agents, 62-67.
- [9] Chaomin L., Yang S. X., Yuan X. 2002. Real-time area-covering operations with obstacle avoidance for cleaning robots. IEEE/RSJ International Conference on Intelligent Robots and Systems, C.3. (3): 2359-64.
- [10] Ok K., Ansari S., Gallagher B., Sica W., Dellaert F., Stilman M. 2013. Path planning with uncertainty: Voronoi Uncertainty Fields. IEEE International Conference on Robotics and Automation (ICRA), 4596-4601.
- [11] Çayırpunar Ö. 2009. Çoklu Robot Sistemlerinde Robotlar Arası Haberleşme ve İşbirliği Kullanılarak Arama Verimliliğinin Artırılması. ETÜ, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Erzurum.
- [12] Tiemin H., Yang S.X. 2002. Real-time torque control of nonholonomic mobile robots with obstacle avoidance. Proceedings of the IEEE International Symposium on Intelligent Control, 81-86.
- [13] Xiaochuan W., Yang S.X. 2003. A neuro-fuzzy approach to obstacle avoidance of a nonholonomic mobile robot. Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003), C1. (1):29-34.

- [14] Burgard W., Stachniss C., Grisetti G., Steder B., Kümmerle R., Dornhege C., Ruhnke M., Kleiner A., Tardós J.D. 2009. A Comparison of SLAM Algorithms Based on a Graph of Relations. In Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), St. Louis, MO, USA.
- [15] Rubenstein M., Ahler C., Nagpal R. 2012. Kilobot: A low cost scalable robot system for collective behaviors. IEEE International Conference on Robotics and Automation, Saint Paul, MN, 3293-3298.
- [16] Anonim 2018. Robotics in Manufacturing and the Life Sciences - Fraunhofer IFF". 2018. Fraunhofer Institute for Factory Operation and Automation IFF. 2018. <https://www.iff.fraunhofer.de/en/business-units/robotic-systems/research/robotics-manufacturing-life-sciences.html>.
- [17] Dönmez E., Kocamaz A.F., Dirik M. 2017. Visual based path planning with adaptive artificial potential field. 25th Signal Processing and Communications Applications Conference (SIU), Antalya, 1-4.
- [18] Dirik M., Kocamaz, A.F. 2015. Tepe Kamera sistemi ile Statik Ortamlarda Yapay Potansiyel Alan Algoritması ile Gezgin Robot Rota Planlaması. 23th Signal Processing and Communications Applications Conference (SIU), 124-127.
- [19] Dirik M., Kocamaz A.F., Dönmez E. 2016. Vision-based decision tree controller design method sensorless application by using angle knowledge. 24th Signal Processing and Communication Application Conference (SIU), 1849-52.
- [20] Kocamaz A.F., Dirik M., Dönmez E. 2016. Head Camera-Based Nearest Neighborhood Relations Algorithm Optimization and the Application of Collecting the Ping-Pong Ball. International Conference on Natural Science and Engineering (ICNASE'16), 67-72.
- [21] Dönmez E., Kocamaz A.F., Dirik M. 2017. A Vision-Based Real-Time Mobile Robot Controller Design Based on Gaussian Function for Indoor Environment. Arabian Journal for Science and Engineering, 1-16.
- [22] Dönmez E., Kocamaz A.F. 2018. Multi Target Task Distribution and Path Planning for Multi-Agents. 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya, Turkey, 1-7.