

Baskılı Devre Kartlarındaki Delik Alanlarının Görüntü İşleme Teknikleri Kullanılarak Tespit Edilmesi ve Potansiyel Delim Güzergâhının Genetik Algoritmalar ile Eniyilmesi

Mehmet KARAKOÇ*¹

¹Antalya AKEV Üniversitesi, Mühendislik ve Mimarlık Fakültesi, Yazılım Mühendisliği Bölümü, Serik, Antalya

(Alınış / Received: 12.12.2016, Kabul / Accepted: 28.06.2017, Online Yayınlanma / Published Online: 14.08.2017)

Anahtar Kelimeler

Baskılı devre kartı,
Kontrol sistemi,
Görüntü işleme,
Genetik algoritmalar,
Yerel arama,
Gezgin satıcı problemi

Özet: Bir elektronik devrenin fiziksel gerçekleştirilmesi, bir elektronik kartın *Baskılı Devresini* (PCB) delme işlemini gerektirir. Delinecek deliklerin konumlarının ve konum sırasının ise bu işlemi yerine getirecek *Kontrol Sistemi* tarafından bilinmesi gerekir. Bu çalışmada, bu işlemin düşük maliyetle ve hızlı bir biçimde gerçekleştirilebilmesi için gerekli bilgileri bir kontrol sistemine sağlayacak yöntemler ele alınmaktadır. Bu bilgiler, PCB üzerine yerleştirilecek elektronik devre elemanlarının konumları ve bu konumları içeren en kısa güzergâhtır. Konumlar, bilgisayar ortamındaki iki-boyutlu PCB görüntüsüne *Görüntü İşleme* teknikleri uygulanarak tespit edilir. Kontrol sisteminin takip edeceği en kısa güzergâh ise *Genetik Algoritmalar* ve *Yerel Arama* tabanlı melez bir algoritma ile *Gezgin Satıcı Problemi* çözülerek belirlenir. Önerilen bütünlük yöntemi bilgisayar benzetimleri ile desteklenmiş ve tüm delik konumlarını içeren çok daha kısa güzergâhlar elde edilmiştir.

Detection of the Hole Areas on Printed Circuit Boards using Image Processing Techniques and Optimization of the Potential Drilling Path with Genetic Algorithms

Keywords

Printed circuit board,
Control system,
Image processing,
Genetic algorithms,
Local search,
Traveling salesman problem

Abstract: The physical implementation of an electronic circuit requires drilling the *Printed Circuit Board* (PCB) of an electronic card. The *Control System* to fulfill this task needs to know the locations of the holes to be drilled with the sequence. In this study, the methods to provide the required information to a control system are considered for rapidly accomplishing this task at low cost. This information consists of the locations of the electronic circuit components to be placed on the PCB and the shortest path with these locations. The locations are detected by applying *Image Processing* techniques to the 2D PCB image on the computer environment. The shortest path to be followed by the control system is determined by solving the *Traveling Salesman Problem* with a hybrid algorithm based on *Genetic Algorithms* and *Local Search*. This integrated method proposed was supported with computer simulations and the much shorter paths were obtained with all holes locations.

1. Giriş

Tüm elektronik cihazlarda bulunan Baskılı Devre Kartı (PCB - *Printed Circuit Board*), üzerine montajı yapılacak elektronik devre elemanlarının birbirleri arasında olan elektriksel iletişimini sağlar. Kartın üzerindeki yollar çeşitli safhalardan geçirilerek elde edilir ve elektronik elemanların bacakları bu yüzeye lehimlenir. Yol aralıkları ve PCB üzerine yerleştirilecek bu bileşenler, farklı şekil ve büyüklüklerde olabilirler. Bu bileşenlerin yerleştirileceği noktalarda delim işlemlerini gerçekleştirecek bir kontrol sisteminin, PCB

üzerindeki belirli delik alanlarına yönlendirilebilmesi gerekir. Delik konumları, genellikle dosya gibi ortamlarda tutulur ve hazır programlar aracılığıyla sağlanır. Örneğin Bilgisayar Destekli Tasarım (CAD - *Computer Aided Design*) programlarıyla oluşturulan *Excellon* dosyaları, baskı devre endüstrisinde standart olarak kullanılır. Bu dosyalarda, PCB üzerindeki tüm delik konumlarının belirli bir ofset noktasına göre belirlenmiş uzaklıkları, delik çapı ve farklı pek çok mekanik parametre bulunur.

CNC (*Computer Numerical Control*) baskı devre delme/delgi makinesi projesi olan *Maximus* [1],

Excellon biçimindeki dosyaları okuyarak delik koordinatlarını belirleyen ve bu koordinatlara göre PCB'yi otomatik olarak delen sistemin mekanik ve elektronik tasarımıdır. İlgili çalışmada bilgisayar yazılımının temel görevi, tasarlanan Grafiksel Kullanıcı Arabirimi/Arayüzü (GUI - *Graphical User Interface*) aracılığıyla makine ile haberleşmeyi sağlamaktır. Delinecek delik konumlarının tespit edilmesi ise söz konusu değildir.

1.1. Literatür taraması

Orhan [2], motorlara ilişkin tüm kontrol işaretlerini yazılım tarafından oluşturarak, çok sayıda giriş ve çıkışa olanak veren, motor sürücüleri ve referans noktalarına ait giriş bilgileri gibi diğer elektronik devreler arasındaki bilgi alışverişini sağlayan bir arabirim geliştirmiştir. Ayrıca kullanıcıya, hazır bir kart dosyasını çağırarak kart delme gibi özellikler de sunmuştur. Mehmetcik [3] ise cisim tanıma yöntemlerinden yararlanarak, düşük maliyetli ve kolay uygulanabilir bir yöntem geliştirmiştir. Uygulamasında, cisim taramak için bir çizgi lazer modülü, cismin görüntüsünü alan bir kamera ve cisim belirli aralıklarla çizgi lazer ışığını kesecek şekilde kaydıran bir adım-motor sistemi kullanmıştır.

Aslantaş'ın [4], araştırmacıların daha fazla ağaç halkası üzerinde inceleme yapabilmeye (örneğin halkalar arasındaki uzaklıklar) olanağı bulabilecekleri sistemi, bir mikroskop ve bu mikroskoba iliştirilen ağaç halkasından örnekler almak için gerekli bir kameradan oluşmaktadır. Görüntü alınacak örnek, mikroskobun altındaki masaya yerleştirilmiştir. Kamera, mikroskop odağı ve masanın pozisyonu bir iş-istasyonu ile kontrol edilirken, masanın hareketini sağlamak için iki adım-motor yerleştirilmiştir. Kalaycı vd. [5], sayısal optik mikroskoplarda elektronik devre kontrollü (çok noktadan kontrol yöntemi temelli) otomatik odaklama işlemi yapabilen sayısal-görüntüleme sistemlerinde, mekanik sisteme ek olarak Görüntü İşleme (Gİ) yazılımı da geliştirmişlerdir. Heterojen yüksekliklere sahip nesnelere incelenmesinde, mikroskobun otomatik olarak odaklanması sağlanır. Nesnelere şekillerine göre görüntü üzerinde belirlenen noktalarla en uygun odaklama noktası belirlenir ve motorlu düzenele mikroskobun z-eksenindeki konumu en uygun noktaya taşınır.

Altinkurt ve Kahrıman [6], hedef nesnenin kamera ile alınan görüntünün merkezine çekilmesini sağlayarak, gerçek-zamanlı bir nesne takibi uygulaması geliştirmişlerdir. Yakalanan görüntünün *MATLAB* ile uyarlanabilir bir biçimde işlenmesiyle, hedef nesnenin merkezini tespit etmişler ve bu merkez görüntünün merkezine olan uzaklığını bularak, seri-porta bağlı mikro-denetleyicili devre aracılığıyla, üzerinde kamera bulunan adım-motorun hareketini sağlamışlardır. Peker ve Zengin [7] ise bilgisayara takılacak bir Web kamerası aracılığıyla ortam güvenliğinin düşük maliyetle sağlanabilmesi

için bir program geliştirmişlerdir. Kamera ile alınan görüntüler Gİ teknikleri ile işlenir ve güvenliği sağlanmak istenen ortama yapılan izinsiz girişler algılanır. Bilgisayara bağlı adım-motor düzeneğinin üzerinde bulunan kamera, nesnenin hareketli piksellerinin ağırlık merkezine göre dönerek, hareket eden nesnelere sürekli olarak takip edecek şekilde yönlendirilir. Ten rengi alanlar üzerindeki yüz tespit edilir ve şüpheli kişinin yüz görüntüsünün kime ait olduğunu belirlemek için yüz tanıma işlemi gerçekleştirilir.

Çayıroğlu ve Şimşir [8], radyo frekansı sinyalleri ile uzaktan kumanda edilen bir robot araba üzerine kontrolü ve görüntü alımı uzaktan olan bir kamera bağlayarak, kontrol işlemini gerçekleştirmişlerdir. Uzaktan gönderilen sinyallerle, kamera düşey ve yatay ekseninde adım-motor kontrolü ile çevrilmiş ve alınan görüntüler, gerçek-zamanlı olarak ve doğrudan görsel temas kurulmadan bilgisayar ekranından takip edilmiştir. Geliştirilen yazılım ve bilgisayarın paralel-portuna bağlı bir devre aracılığıyla, robot arabanın kamerasının hareketi için kullanılan adım-motorların ve arabanın hareketi için kullanılan DC-motorun kontrolü gerçekleştirilmiştir. Kameranın sağa ve sola dönme hareketi bağlı bulunduğu adım-motorun kontrolüyle, adım-motorun kontrolü ise PIC kullanılarak sağlanmıştır. Böylece, güvenlik nedeniyle insanların giremediği bir ortama robot arabanın gönderilmesi ve uzaktan gerekli görüntüler alınıp bilgisayara gönderildikten sonra, robotun bilgisayar üzerinden Gİ ile otomatik olarak yönlendirilmesi mümkün olmuştur.

Yılmaz [9], girilen değiştirilebilir koordinatları motorların buldukları koordinatlarla karşılaştırmış ve adım-motorun bir noktadan başka bir noktaya hareketini sağlamıştır. Gökmen [10], adım-motor kontrolünde, hız ve konum kontrolünü gerçekleştirmiştir. Uyar vd. [11] ise kamera ile ebatları veya çeşitli özellikleri sayesinde tanımlanan nesnelere, sınıflandırmışlar ve daha önceden tanımlı yörüngeyi takip ederek belirlenmiş konumlarda bölümlenmişlerdir. Uygun [12], adım-motorun endüstri ve eğitim uygulamalarında kontrolü için bir yazılım geliştirmiştir. Yıldız [13], CAD dosyalarındaki uygulamada kullanılacak olan verileri okuyup veri tabanına aktararak, kontrol sistemine gerekli parametreleri sağlamıştır. Usta vd. [14] ise askerî alanlar için bilgisayar destekli silah kontrolü sistemlerinde, kamera ile alınan görüntülerdeki hareketli nesnelere tespit edilmesi ve silahın hareketi için adım-motorlar kullanmışlardır.

Katagiri vd. [15], PCB denetimlerini kolaylaştırmak amacıyla her PCB tabakası üzerindeki bir dizi kablolama (kablo bağlantısı) örüntüsünü (*wiring patterns*) sınamada, en kısa uzunluklu güzergâhı bulacak bir eniyileme algoritması ele almışlardır. Problem, öncelik-kısıtlı (*precedence-constrained*) Gezgin Satıcı Problemi (TSP - *Traveling Salesman Problem*) veya 0-1 tamsayı programlama problemi

olarak modellenmiştir. Kullanılan verimli bir sezgisel algoritmanın haricinde, 2-opt ve or-opt yerel aramalarından da yararlanılmıştır. Katagiri vd. [16] ise tüm PCB kablolama örüntülerinin kontrolü için toplam güzergâh uzunluğunu en aza indirecek bir yol eniyileme problemini ele almışlardır. Problemin genelleştirilmiş dağıtım-toplamalı (*pickup and delivery*) TSP olarak formüle edildiği gösterilmiş ve büyük-ölçekli gerçek-dünya problemlerini pratik bir zamanda çözebilmek için verimli bir sezgisel algoritma önerilmiştir.

PCB delik delme işleminde, Lim vd. [17], en kısa delim yoluna ilişkin sıralamanın eniyilemesi (*sequence optimization*) için *Cuckoo* arama algoritması kullanmışlardır. İsmail vd. [18] ise en uygun güzergâhı aramak için seyahat süresini azaltacak ateşböceği algoritması (*firefly algorithm*) uygulamışlardır. Narooei vd. [19], çok sayıda deliği içeren rota uzunluğunu en aza indirmede, karınca kolonisi eniyilemesi (*ant colony optimization*) algoritması kullanmışlardır. Bir CNC makinesi kullanarak, farklı sayılarda delikler içeren delim işlemleri için en kısa güzergâhları belirlemişlerdir. Dalavi vd. [20] ise delik-açma işlemlerinin en uygun sırasının belirlenmesi için *shuffled frog leaping* algoritması kullanmışlardır. Srivastava [21], PCB delimine ilişkin rotalama sürecini eniyileme için yeni geliştirilmiş sezgisel tekniklerden birini, delik konumları için mevcut TSP dosyası kullanmıştır. Çözüm, bir nehri oluşturan su damlacıklarının davranışı ve başlangıç-hedef noktaları arasındaki seyahatine ilişkin güzergâhının belirlenmesinin benzetimini içeren zeki su damlacıkları (*intelligent water drops*) algoritması (daha az yineleme söz konusu) temellidir. Srivastava'ya [21] göre, PCB delik rotasını eniyileme süreci, PCB montaj hattında (*assembly line*) hâlen bir darboğaz teşkil etmektedir.

Abdullah vd.ne [22] göre, bir CAD veya bilgisayar destekli üretim sisteminde oluşturulan güzergâh *en uygun olan* olmayabilir ve daha büyük uzunluklara bağlı olarak işlem zamanı (*machining time*) artabilir. En kısa delim güzergâhını TSP olarak uyarlamışlar, delik delme işlemine ilişkin daha kısa uzaklıklar için karınca kolonisi eniyilemesinden ve Genetik Algoritmalar (GA) tabanlı yol eniyilemeden yararlanmışlardır. Alkaya ve Duman [23] ise son yıllarda PCB montajında (*assembling*) popüler hâle gelen *chip shooter component placement machines* eniyilemesini çalışmışlardır. Montaj süresini en aza indirme problemi ve sıra-bağımlı (*sequence-dependent*) TSP ve karesel atama problemi (*quadratic assignment problem*) alt-problemlerine çözüm için yinelemeli bir yöntem önermişlerdir.

Shinde ve Morade [24], bir PCB hata tespiti ve sınıflandırma sistemi önermişlerdir. Hataların altında yatan nedenlerin çözümlenmesini ve belirlenmesini amaçlamışlar, Gİ teknolojisi temelli bir otomatik hata tespiti sistemi tasarlamışlardır. Şablon PCB ve denetlenecek PCB görüntüleri üzerinde görüntü

çıkarma (*image subtraction*) algoritması uygulanarak, PCB üzerindeki hatalar bulunmuş ve hatalara ilişkin ayrıntıları elde edebilmek için çeşitli yapısal özellikler çıkarılmıştır.

Vinita ve Kaushik [25], PCB için eksik hat, kısa devre, eksik delik ve açık/kırık alan gibi kart üretim hatalarını otomatik olarak tespit edecek bir bilgisayarlı görü (*computer vision*) sistemi sunmuşlardır. Kritik ve potansiyel hataları dikkate almışlar, sorunlu ana bölgeleri öne çıkarabilen basit bir çıkarma algoritması kullanarak, standart bir PCB görüntüsünü bir PCB görüntüsü ile karşılaştırmışlardır. Ayrıca, bu yöntemi gerçek bir PCB'de geliştirmek için problemler alanları büyütme ve temel PCB görüntüsünden daha küçük biri dizi PCB bölümündeki hataları bulmayı önermişlerdir. Ancău [26], PCB üretimini eniyilemeyi amaçlamış, delikler arasındaki delim güzergâhının uzunluğunu en aza indirmek ve işlem zamanını azaltmak için TSP'yi çözecek orijinal bir melez sezgisel algoritma önermiştir. Aoyama vd. [27] ise PCB üzerindeki delim işlemleri için söz konusu olan hareket süresine odaklanmışlardır. PCB üzerinde açılacak deliklerin sırasını dikkate almışlar, hareket rotasını belirlemek için TSP uygulamışlardır.

1.2. Çalışmanın katkısı

Bu çalışmada, bir elektronik karta ilişkin PCB görüntüsündeki farklı şekil ve büyüklüklerdeki delik alanlarını Gİ teknikleri ile tespit edecek ve tüm delik konumlarını içeren en kısa güzergâhı geliştirilen GA ve yerel arama tabanlı melez bir algoritma ile belirleyecek bütünlük bir yöntem önerilmektedir. Ayrıca, belirlenen güzergâhı takip edeceği varsayılan bir kontrol sistemi için bu güzergâha ilişkin potansiyel hareket bilgisi de üretilmektedir.

Farooque vd. [28], bir PCB delme makinesinin tasarımı ve gerçekleştirilmesinde, delik koordinatlarını el ile girmek gerekmez, devre görüntüsünden delikleri otomatik olarak tespit etmişlerdir. Deliklerin tespit edilebilmesi için dairesel *Hough* dönüşümü (*circular Hough transform*) kullanmışlardır. *Hough* dönüşümü, sayısal görüntülerdeki düzgün doğruların ve doğrusal olmayan eğri, çember ve elips gibi düzensiz şekillerin tespit edilmesini kolaylaştıran ve görüntü analizi için sıkça kullanılan bir yöntemdir. Bu dönüşümde, görüntüden yakalanacak her şekil için nesnenin özelliklerine bağlı olarak ayrı parametreler ve denklemler geliştirilmiştir. Çalışma kapsamında önerilen yöntem ise şekil türünden bağımsız olarak ve piksel komşuluklarına bakarak şekil bulmaya yönelik bir algoritma kullanır ve her şekil için ayrı parametre veya denklem tanımlamaları gerektirmez. Farooque vd.nin [28] yol planlama (*path planning*) için kullandıkları satır satır tarama (*row by row scanning*) yöntemi ise bir koordinat değeri sabit tutulup diğeri değiştirilerek *x-y* koordinatları boyunca düzlemin taranması şeklinde uygulanır. Bu çalışmada ise bu problem TSP olarak ele alınmıştır.

Mevcut çalışmalarda, Gİ tekniklerine dayalı olarak konum tespiti ve tespit edilen konumlara ilişkin güzergâhın eniyilemesi söz konusu değildir. Çalışma kapsamında, açık kaynak kod, CAD programı veya kütüphane gibi herhangi bir yardımcı araç kullanılmamıştır. Bu çalışmanın temel katkısı, problemin hem Gİ hem de eniyileme kısımlarının ayrı ayrı ele alınıp birleştirilmesiyle, bütünleşik bir yöntem önerilmiş olmasıdır. Örneğin, (i) mevcut bir kart görüntüsü veya benzer bir görüntüden, sadece ilgili alanları tespit etmek istenebilir; (ii) söz konusu alanlara ilişkin konumların bilindiği durumlarda, sadece en kısa güzergâhın belirlenmesi yeterli olabilir veya (iii) mevcut görüntüden doğrudan en kısa güzergâhın elde edilmesi istenebilir. Bu çözüm bu şekliyle, farklı alanlardaki problemlere cevap verebilme ve çözümlerine uyarlanabilme noktalarında oldukça esneklerdir.

Çalışmanın geri kalan kısmı şu şekilde düzenlenmiştir. 2. Bölümde, PCB görüntülerindeki delik alanlarının tespit edilebilmesi için önerilen çözüm yöntemi ve potansiyel delim güzergâhının eniyilemesi amacıyla uygulanan çözüm tüm ayrıntılarıyla anlatılmaktadır. 3. Bölümde, gerçekleştirilen başarımların testlerine ilişkin ayrıntılar ve elde edilen test sonuçları verilmekte ve önerilen yöntemin başarımlarını tartışılmaktadır. 4. Bölümde ise çalışmaya ilişkin sonuçlar sunulmaktadır.

2. Materyal ve Metot

Tasarlanan sistemin işleyişine ilişkin akış Tablo 1'de verilmiştir.

Tablo 1. Önerilen yöntemle ilişkin iş-parçacığı listesi.

İş-Parçacığı	Açıklama
1. PCB görüntüsünün alınması	Orijinal boyutlar
2. Görüntüye ön işleme uygulanması	Gİ teknikleri
3. Delik alanlarının tespit edilmesi	Önerilen çözüm
4. İlk tespit güzergâhının çizilmesi	Görselleştirme
5. En kısa güzergâhın belirlenmesi	GA + yerel arama
6. Yeni güzergâhın çizilmesi	Görselleştirme
7. Hareket bilgisinin üretilmesi	Kontrol sistemi
8. Sonuçların görüntülenmesi	Sonuçlar sunulur.

Sistemi oluşturan temel iş-parçacıklarına ilişkin tüm ayrıntılar ise alt bölümlerde verilmiştir.

2.1. Delik alanlarının tespit edilmesi

Bu bölümde, PCB görüntülerine uygulanan ön işleme ve delik alanlarının tespit edilebilmesi için önerilen çözüm yöntemi anlatılmaktadır.

2.1.1. Ön işleme için uygulanan görüntü işleme teknikleri

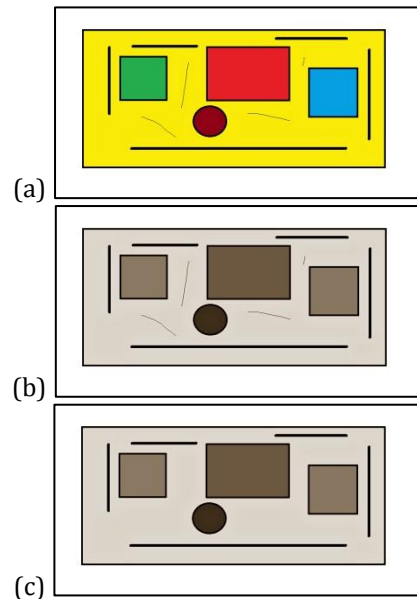
Delik alanlarının tespit etmeden önce, üzerinde çalışılacak PCB görüntüsüne Şekil 1'de akışı verilen bir dizi işlem uygulanır.

Algoritma: applyPreprocessingToImage(Image loadedPCBImage)	
1. grayImage	= convert2Gray(loadedPCBImage)
2. denoisedImage	= removeNoiseWithMeanFilter(grayImage)
3. imgWithEdges	= detectEdgesWithSobel(denoisedImage)
4. histogram	= createHistogram(imgWithEdges)
5. threshold	= calculateThresholdValue(histogram)
6. binaryImage	= convert2Binary(imgWithEdges, threshold)
7. finalPCBImage	= cropPCBPart(binaryImage)
8. return	finalPCBImage

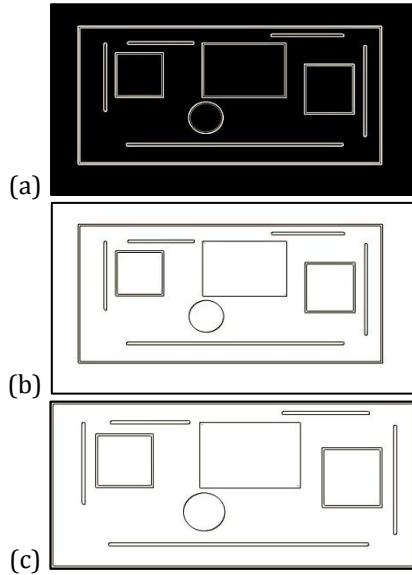
Şekil 1. Tablo 1'deki 2. adım için iş-akışı

İlk olarak, görüntü (Şekil 2(a)), piksel değerleri belirli sayılarla çarpılarak griye çevrilir (Şekil 2(b)). Görüntü üzerindeki gürültülerin, gereksiz ayrıntıların ve belirsizliklerin giderilmesi için gri görüntünün renk değerleri üzerine, temel olarak komşuluk prensibini ele alan ortalama filtreleme (*mean filter*) uygulanır (Şekil 2(c)). Bu işlem, filtreye ilişkin "3×3"lük çerçeveye karşılık gelen görüntüdeki her alt görüntü için uygulanır. Her alt görüntüdeki piksel değerlerinin ortalaması, bu çerçevedeki merkez noktanın yeni piksel değeri olur.

Daha sonra, *Sobel* kenar-belirleme algoritmasıyla, "3×3"lük kenar matrisleri kullanılarak, görüntü üzerindeki yatay, dikey ve köşegen şeklindeki kenarlar, kenarlar beyaz ve diğer kısımlar siyah olacak şekilde belirlenir (Şekil 3(a)). Görüntünün bu hâlinin piksel değerlerinin dağılımını içeren histogram oluşturulur ve en uygun eşik değeri hesaplanır. Değeri eşik değerinden küçük olan piksel değerleri beyaz, diğer piksel değerleri ise siyah yapılarak ikili görüntü (*binary image*) elde edilir (Şekil 3(b)). Son olarak, siyah-beyaz bu görüntünün en dar alanda siyah piksellerle çevrili kısmı kırılır (Şekil 3(c)). Ön işleme sonunda işlenmeye hazır bu görüntü, şekillere ait hatlar siyah ve diğer kısımlar (şekillerin iç kısımları ve delikler dışındaki alanlar) beyaz olacak şekilde elde edilir. Böylece, görüntüdeki şekillere ait hatlara bakarak delik alanlarını tespit edebilmek mümkün olur.



Şekil 2. Örnek görüntü (a), griye çevrilmiş (b) ve ortalama filtre ile gürültüleri giderilmiş (c) görüntüler



Şekil 3. Kenarları belirlenmiş (a) ve siyah-beyaz hâle getirilmiş (b) görüntüler ve işlenmeye hazır görüntü (c)

2.1.2. Önerilen çözüm yöntemi

Dikdörtgen ve çember gibi düzenli ve/veya girintili içeren düzensiz şekillerden oluşabilen tüm delik alanlarını tespit edebilmek için siyah-beyaz görüntü taranır ve görüntüdeki kenarlar arasındaki yollarda ilerleyerek siyah piksellerle sınırlandırılmış şekiller aranır. Şekilden bağımsız olarak geliştirilen bu yöntemle ilişkin iş-akışı Şekil 4'te verilmiştir.

Algoritma: detectHoleLocationsOnPCBImage(Image finalPCBImage)	
1.	holeList = detectHoles(finalPCBImage)
2.	shapelImage = differentiateHoles(finalPCBImage, holeList)
3.	holeImage = extractCenters(shapelImage)
4.	holeImage = drawDrillingOrderNumbers(holeImage)
5.	holeImage = drawDrillingPath(holeImage)
6.	İlk tespit güzergâhını holeImage üzerinde görüntüle

Şekil 4. Tablo 1'deki 3. ve 4. adımlar için iş-akışı

Delik alanlarının tespit edilebilmesi için önerilen algoritma iki adımdan oluşmaktadır: şekli ve/veya şeklin bazı parçalarını, (1) soldan sağa yatay boyunca tarama ve (2) yukarıdan aşağıya düşey boyunca tarama. Geliştirilen, şekli yatay boyunca tarayan algoritma (Şekil 4'te 1. adım) Şekil 5'te verilmiştir.

Şekil 5'te goto durumlarında, taramalara kalınan noktadan devam edilir. Delik alanlarına ilişkin x_c ve y_c değişkenleri ise Tablo 2'de verilmiştir. Her delik alanı için yatay boyunca düşeydeki dış-sınır noktaları (siyah (x_i, y_b) ve (x_i, y_e) pikselleri) arasındaki uzunluk değerleri karşılaştırılır. Bu değerlerin tümü aynı ise şekil bir dikdörtgen olabilir. Çok küçük veya büyük şekillerin delik alanı olarak değerlendirilmemesi için art arda gelen uzunluklar arasındaki farklar kontrol edilir. Değerler azalıyor ve daha sonra artıyorsa, şekil iç-bükey bir çokgen olabilir. Ayrıca, uzunluklar bir baştan bir sondan karşılaştırılarak, bu değerlerdeki değişime bakılır. Aynı ise şeklin merkezinden geçen düşeye göre simetrik özelliğini sağladığı kabul edilir.

Algoritma: detectHoles(Image finalPCBImage)	
1.	holeList = []
2.	" x_b, y_b " = finalPCBImage üzerinde sağında beyaz piksel (x_{b+1}, y_b) olan siyah piksel
3.	while " (x_{b+1}, y_b) için düşeyde yukarı-aşağı yönlerde beyaz pikseller boyunca ilerle" do
4.	if "şekil sol taraftan siyah piksellerle sınırlı/kapalı değil" then
5.	goto adım-2
6.	end if
7.	end while
8.	while "sağa doğru ilerlenirken bulunacak ilk siyah piksele (x_e, y_b) kadar yatayda ilerle" do
	// yatay boyunca her piksel $(x_i: b < i < e)$ için şekil
	// üst ve alt kısımlardan kapalı olmalı
9.	if "düşeyde yukarı-aşağı yönlerde beyaz pikseller boyunca ilerleyerek siyah piksellere $((x_i, y_b)$ ve $(x_i, y_e))$ ulaşamadı" then
10.	goto adım-2
11.	end if
12.	if " (x_i, y_b) ve (x_i, y_e) piksellerinin sol ve sağ komşuları için kapalılık yok" then
13.	goto adım-2
14.	end if
15.	end while
16.	while " (x_{e-1}, y_b) için düşeyde yukarı-aşağı yönlerde beyaz pikseller boyunca ilerle" do
17.	if "şekil sağ taraftan siyah piksellerle sınırlı/kapalı değil" then
18.	goto adım-2
19.	end if
20.	end while
21.	hole = new PointF(x_c, y_c)
22.	holeList.Add(hole)
23.	if "görüntüde taranacak alan mevcut" then
24.	goto adım-2
25.	end if
26.	return holeList

Şekil 5. Delik alanlarını tespit eden algoritma

Şekil 5'teki algoritmanın en kritik kısmı "8-15 adımları" olup, bu kısım görüntüde taranacak alan mevcut olduğu sürece yinelenir ve her şekil için yatay ve düşey eksenler boyunca ilerlenir. Olası şekil sayısı n ve her şekil için genişlik ve yükseklik değerleri sırasıyla w ve h olmak üzere, algoritmanın zaman karmaşıklığı $O(n*w*h)$ olarak ifade edilebilir.

Tablo 2. Delik alanı değişkenleri.

Değişken	Değer
Merkez noktanın x -koordinatı	$x_c = (x_b + x_e) \div 2$
Merkez noktanın y -koordinatı	$y_c = (y_b + y_e) \div 2$

Tablo 2'de y_b ve y_e sırasıyla, merkez noktanın düşeyinde yukarı-aşağı yönlerde ilerleyerek ulaşılan siyah piksellerin y -koordinatı değerleridir. Görüntüdeki her delik alanı tespit edildikten sonra, karşılık gelen şekil için merkez piksel koyu-mavi ve şeklin iç kısmını pembe yapılarak, bu şekil görüntünün diğer kısımlarından farklılaştırılır. Tüm delik alanları tespit edildikten sonra, görüntü üzerindeki merkez pikseller dışındaki tüm alan beyaz yapılarak, tüm delik konumları belirginleştirilir. Daha sonra, tespit edilme sırasına göre, delik konumlarının her birinin sağ altına belirli biçim ve renkte sıra numaraları çizdirilir. Son olarak, birbirini takip eden delik

konumları arasındaki hareketi belirlemek için bu konumlar arasına mor çizgiler çizdirilerek potansiyel delim güzergâhı görüntülenir.

2.2. Potansiyel delim güzergâhının eniyilemesi

Elde edilen konum listesi (Şekil 5'te *holeList*) kullanılarak, TSP'de olduğu gibi, kartın sol üst köşesinde (sıfır) başlayıp sonlanan ve her delik konumunu (toplam delik konumu sayısı: n) tam olarak bir kez ziyaret eden en kısa güzergâh belirlenir. TSP, toplam konum sayısı " $n + 1$ " olacak şekilde GA ile çözülür. Nüfustaki her kromozom (aday çözüm), noktaların ziyaret edilme sıralarına karşılık gelen numaraları içeren bir tur olarak kodlanır. Başlangıç nüfus, her konum için bu konuma en yakın uzaklıktaki c konum belirlenerek, birbirine daha yakın nokta çiftleri p olasılıkla art arda gelebilecek şekilde üretilir. Böylece, daha iyi kromozomlar oluşturulabilir. GA uygunluk değeri çözüme ilişkin turun uzunluğu olup, en kısa tur uzunluğuna sahip kromozom ise en iyi çözümdür.

2.2.1. En kısa güzergâhın belirlenmesi

NP-zor bir problem olan TSP'nin çözümü için geliştirilen melez algoritma Şekil 6'da verilmiştir.

Algoritma: `determineTheShortestDrillingPath(double[,] distanceMatrix)`

1. Başlangıç nüfusu üret
2. Ebeveynlerin uygunluk değerlerini hesapla
3. Ebeveynleri sırala
4. `drillingPath = en az uygunluk değerine sahip çözüm`
5. En kısa güzergâhı (*drillingPath*) görüntüle
6. **while** "nesil sayısına ulaşılmadı" **do**
7. Seçkinlik uygula
8. **repeat**
9. - iki ebeveyn seç
10. - ebeveynler arasında çaprazlama işlecini uygula
11. - çocuklara *yemel arama* uygula
12. **until** "yeni nüfus için nüfus büyüklüğüne ulaşıldı"
13. Çocukların uygunluk değerlerini hesapla
14. Kromozomları (seçkin ebeveynler ve çocuklar) sırala
15. `drillingPath = en az uygunluk değerine sahip çözüm`
16. En kısa güzergâhı (*drillingPath*) görüntüle
17. **end while**
18. **return** `drillingPath`

Şekil 6. Tablo 1'deki 5. ve 6. adımlar için sözde kod

Şekil 6'da *distanceMatrix*, tüm nokta-çiftleri (delik konumları) arasındaki Öklid karesel uzaklıklarını (*Euclidean squared distances*) içeren uzaklık matrisidir. Bu matris, kromozomların uygunluk değerlerini hesaplamak için kullanılır. Yeni nüfus iki aşamada oluşturulur: (1) nüfustaki en iyi e ebeveyn doğrudan bir sonraki nesle aktarılır (7. adım) ve (2) nüfus büyüklüğü (n) değişmeyecek şekilde, genetik işlemlerle " $n-e$ " çocuk üretilir (8-12 adımları). 11. adımda, 2-opt yerel arama algoritması uygulanır. Bu algoritmada, yerel en iyi noktaya ulaşılmadığı sürece, kromozom iki noktadan kırılır ve bu aralık ters çevrilerek yeni bir kromozom elde edilir. Böylece,

mevcut çözümler iyileştirilerek, bütünsel en iyi çözüme daha hızlı bir biçimde yakınsanabilir.

Şekil 6'daki algoritmanın en kritik kısmı olan yerel arama, karmaşıklığını azaltabilmek için sadece kromozomu iyileştirecek durumlarda uygulanır. Böylece, konum sayısı n olmak üzere, yerel aramanın $O(n^2)$ zaman karmaşıklığı $O(n \cdot \log n)$ olur. Algoritmadaki yerel arama içeren "8-12 adımları" " $(n-e)/2$ " kez uygulanır ve bu nedenle, maksimum nesil sayısı m olmak üzere, algoritmanın zaman karmaşıklığı $O(m \cdot n^2 \cdot \log n)$ olarak ifade edilebilir.

2.2.2. Delik konumları arasındaki hareket bilgisinin üretilmesi

Varsayılan kontrol sistemi tarafından takip edilecek güzergâha ilişkin potansiyel hareket bilgisini üreten algoritma (Tablo 1'deki 7. adım) için belirlenen güzergâh boyunca, başlangıç konumuna geri dönülene kadar, kontrol sisteminin bulunduğu nokta ile bir sonraki hedef nokta arasındaki uzaklıklar hesaplanır ve tüm uzaklık bilgileri görüntülenir. Her uzaklık kontrol sisteminin *birim ilerleme miktarı* L ile çarpılarak, gerçek uzaklık bilgisi elde edilir. Kontrol sisteminin birbirini takip eden iki konum arasındaki hareketini belirleyen algoritma için hesaplamalar *Chebyshev uzaklığı* kullanılarak yapılmıştır.

Konum sırasının eniyilemesi, TSP'ye benzer ve sıralama türü bir problemidir. Literatürde *delim yolunu eniyileme* problemi olarak çalışılan benzer uygulamalarda, tur uzunluğunu en aza indirmek ve işlem zamanını azaltmak amaçlanır.

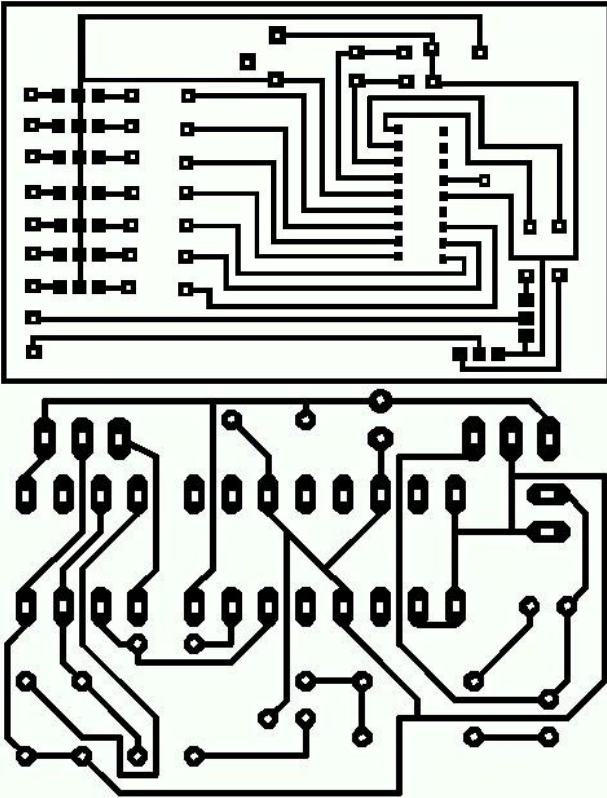
3. Bulgular

Bu bölümde, önerilen yöntemin başarımını gösterebilmek için yapılan testlere ilişkin ayrıntılar verilmiştir.

3.1. Veri kümesi ve deneysel kurulum

Geliştirilen melez algoritma iki farklı şekilde test edilmiştir: (1) algoritmanın istatistiksel başarımını gösterebilmek için *TSPLIB* kütüphanesindeki (Discrete and Combinatorial Optimization [29]) on TSP örneğine ilişkin şehirlerin ve (2) mevcut iki PCB görüntüsü (Şekil 7) için belirlenen delik alanlarının konumları kullanılarak.

Şekil 7'de verilen görüntüler sistemin kontrol edildiği arayüzdeki görüntü kutucuğuna yüklenir ve önerilen bütünleşik yöntem, Tablo 1'de verildiği gibi, şu adımlarla bu görüntülere uygulanır: (i) PCB görüntüsünün alınması, (ii) görüntüye önışleme uygulanması, (iii) delik alanlarının tespit edilmesi, (iv) ilk tespit güzergâhının çizilmesi, (v) en kısa güzergâhın belirlenmesi ve (vi) yeni güzergâhın çizilmesi.



Şekil 7. Örnek test görüntüleri (bu görüntüler için ön işleme adımları ile elde edilen ara görüntüler verilmemiştir)

Görüntüye ön işleme uygulanması: Örnek görüntü, ilk olarak griye çevrilir ve ortalama filtre ile gürültüleri giderilir; daha sonra, kenarları belirlenip siyah-beyaz hâle getirilir ve işlenmeye hazırlanır.

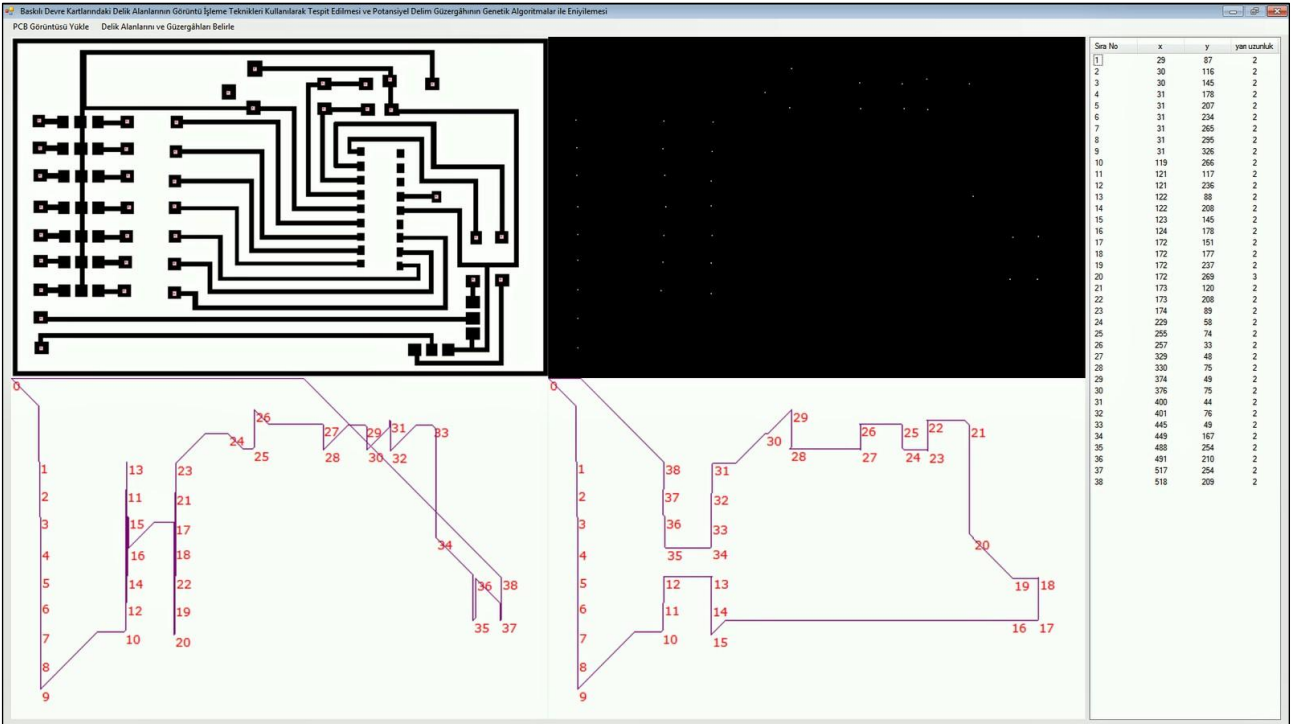
Delik alanlarının tespit edilmesi: Görüntülerdeki ilgili olunan delik alanları, önerilen yöntem ile tespit edilir ve ilk tespit güzergâhı belirlenir. Bu işlem, Şekil 7'deki görüntüler üzerinde siyah pikseller ile sınırlanmış ve iç kısmı beyaz olan ve gözle görülüp ayırt edilebilen alanların tespit edilmesi şeklinde uygulanır. Yöntem şekil kapalılığını esas alır ve uygulamada piksel komşuluklarına bakılır.

En kısa güzergâhın belirlenmesi: Konumları bilinen bir dizi nokta için nokta çiftleri arasındaki uzaklıklar hesaplanır ve literatürde oldukça yoğun bir biçimde çalışılan TSP çözülerek yeni en kısa güzergâh belirlenir.

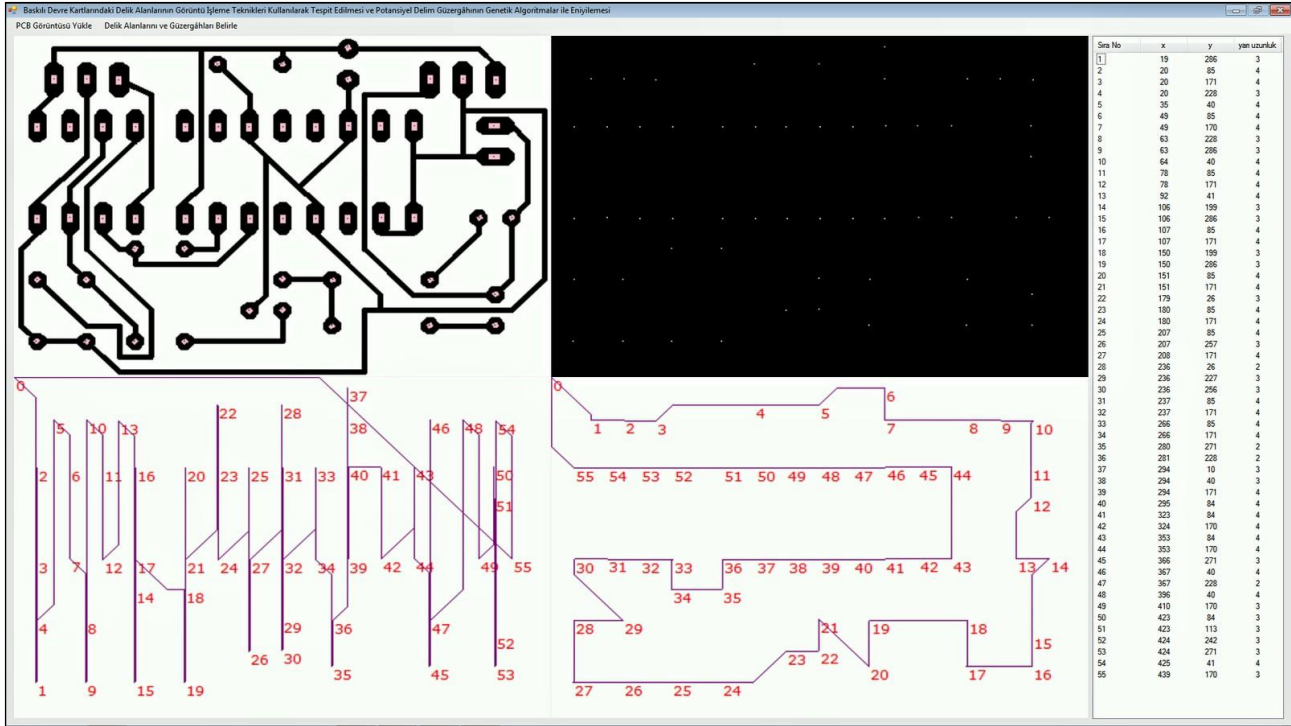
Her test için her ara işlem sonunda, mevcut görsel aşamaya ilişkin görüntüler kaydedilmiştir. Elde edilen sonuçlar, geliştirilen programa ilişkin GUI ile birlikte Şekil 8'de ve Şekil 9'da verilmiştir. Şekillerde görüldüğü gibi, her iki gerçek PCB görüntüsü için de ilk olarak, delik alanları tespit edilerek merkez noktalar bulunmakta ve yarıçap bilgileri hesaplanmakta ve daha sonra, en kısa güzergâhlar belirlenmektedir. Çalışma kapsamında önerilen bu yöntem aracılığıyla, delik alanlarının hızlı bir biçimde bulunabildiği ve tur uzunluklarında belirgin ölçüde iyileşmeler sağlandığı görülmektedir.

Algoritma değişkenleri ve kullanılan değerleri Tablo 3'te verilmiştir.

Başlangıç nüfusu üretmede (Bkz. Bölüm 2.2), c değeri 5, p olasılığı ise %90 olarak tercih edilmiştir.



Şekil 8. Önerilen bütünleşik yöntemin, Şekil 7'de verilen üstteki görüntüye uygulanması: üstte, delik alanları (a) ve merkezleri (b - görüntünün tersi verilmiştir); altta, ilk tespit güzergâhı (c) ve en kısa güzergâh (d) [tur uzunlukları 3050,65 ve 2042,79 olup, tespit edilen 38 delik alanı için tur uzunluğu %67'ye düşmüştür]



Şekil 9. Önerilen bütünleşik yöntemin, Şekil 7’de verilen alttaki görüntüye uygulanması: üstte, delik alanları (a) ve merkezleri (b - görüntünün tersi verilmiştir); altta, ilk tespit güzergâhı (c) ve en kısa güzergâh (d) [tur uzunlukları 6996,44 ve 2391,07 olup, tespit edilen 55 delik alanı için tur uzunluğu %34’e düşmüştür]

Çalışma kapsamında geliştirilen program (tasarlanan GUI (Şekil 8 ve Şekil 9) ve geliştirilen algoritmalar) için *Microsoft Visual Studio 2012* platformunda C# programlama dili tercih edilmiştir. Yapılan testler için 3.20 GHz işlemci ve 8GB RAM ile Intel(R) Core(TM) i5-3470 CPU içeren Windows 7 Ultimate işletim sistemi yüklü bir masaüstü bilgisayar kullanılmıştır.

Tablo 3. Genetik algoritma değişkenleri ve kullanılan değerler.

Değişken	Değer
Nüfus büyüklüğü ve maksimum nesil sayısı	100 - 1000
Seçim yöntemi	Turnuva seçimi
Çaprazlama işlemci ve olasılığı	Tek noktalı permütasyon çaprazlama - %80
Mutasyon işlemci ve olasılığı	2-opt algoritması - %5
Seçkin ebeveyn sayısı (e)	2

3.2. Gerçekleştirilen başarımlar ve sonuçlar

İlk olarak, TSP örneklerine ilişkin şehirlerin konumları kullanılmış, geliştirilen program 100’er kez çalıştırılarak tur uzunlukları (Tablo 4) ve güzergâhlar belirlenmiştir.

Tablo 4’te, her örnek adındaki sayısal kısım ilgili örneğe ilişkin şehir sayısı olup; “en iyi”, ort, ss ve süre sırasıyla, bilinen en iyi değer, geliştirilen yöntem ile elde edilen ortalama uzaklık ve standart sapma değerleri ve GA işletimine ilişkin milisaniye (ms) cinsinden ortalama çalışma süresidir. Algoritmanın işletimi, en iyi değer bulunduğunda iyileşme ile otomatik olarak sonlandırılır.

Tablo 4. TSP çözümüne ilişkin sonuçlar.

Örnek	En İyi (*)	Ort	SS	Süre (ms)
eil51	426	426,21	0,41	1362,64
berlin52	7542	7542	0	165,55
st70	675	675	0	583,07
eil76	538	538,74	1,05	4899,31
pr76	108159	108196,9	58,4	5226,63
kroA100	21282	21282	0	3926,75
kroB100	22141	22169,95	30,14	13487,7
kroC100	20749	20749	0	7236,75
kroD100	21294	21320,51	36,12	16030,53
kroE100	22068	22080,25	22,69	12744,93

* (Discrete and Combinatorial Optimization [30])

İlk deneyde, Tablo 4’te görüldüğü gibi, literatürden bilinen en kısa tur uzunlukları ve en kısa güzergâhlar veya en iyiye yakın değerler ve ilgili güzergâhlar bulunmuştur. Standart sapma değerleri, “berlin52”, “st70”, “kroA100” ve “kroC100” örnekleri için sıfır iken; diğer örneklerde ise oldukça düşüktür.

Önerilen yöntemin gerçek PCB görüntülerine uygulanması ile elde edilen benzetim çıktıları ise tur uzunluklarındaki iyileşmeler ile birlikte Şekil 8’de ve Şekil 9’da verilmiştir. İkinci deneyde, tüm delik alanları tespit edilmiş ve delim güzergâhlarını eniyileme ile tur uzunlukları belirgin ölçüde azalmıştır. Şekil 8’de ve Şekil 9’da, sırasıyla, tespit edilen 38 ve 55 delik alanı için tur uzunlukları %67’ye ve %34’e düşmüştür. Bu bulgu, gerçek PCB görüntüleri için hem tüm delik alanlarının tespit edilebildiğini hem de en kısa güzergâhların belirlenebildiğini göstermiştir. Önerilen yöntem, gerçek-dünya problemlerinde başarılı bir biçimde uygulanabilme noktasında umut vericidir.

4. Tartışma ve Sonuç

Bu çalışmada, bir elektronik karta ilişkin PCB görüntüsündeki delik alanlarının Gİ teknikleri kullanılarak tespit edilmesi ve tüm delik konumlarını içeren en kısa güzergâhın GA ve yerel arama tabanlı melez bir algoritma ile belirlenmesi amaçlandı. Bilgisayar ortamına aktarılan PCB görüntüleri üzerinde istenen ayrıntılar ortaya çıkarılarak, elektronik devre elemanlarının yerleştirileceği delik alanları tespit edildi ve potansiyel delim güzergâhı belirlendi. Ayrıca, PCB üzerinde delim işlemleri gerçekleştirecek bir kontrol sistemi (varsayılan bir adım-motor veya CNC makinesi) için bu güzergâh boyunca takip edilecek potansiyel hareket bilgisi üretildi ve girdi olarak düzenlendi.

Çalışma kapsamında, donanım ile ilgili bir gerçekleştirme, bir adım-motorun sürülmesi, elektronik ve mekanik aksam tasarımı ve kontrolü veya bir delim işlemi söz konusu değildir. Önerilen bütünlük yönteminden, benzer alanların tespit edilmesi ve bu alanlara ilişkin konumları içeren güzergâhın eniyilemesi amaçlanan benzeri çalışma ve uygulamalarda yararlanılabilir. Ayrıca, geliştirilen program, delik konumlarının bilinmediği, bu bilginin mevcut olduğu dosya vb. ortamların temin edilemediği veya mevcut delik konumlarının belirlenerek doğrulanması gerektiği durumlarda kullanılabilir. Gerekli bilgilerin geliştirilen program aracılığıyla otomatik olarak sağlanması ve delim güzergâhının uzunluğunun en aza indirilmesi, işlem zamanında belirgin ölçüde azalma sağlayabilecek ve üretim sistemlerinin verimliliğini doğrudan artıracaktır.

Gelecek çalışma olarak, bu çalışmadaki problemin birden çok güzergâh içerecek şekilde genişletilmesi ve geliştirilen melez algoritmanın bu yönde yapılandırılması planlanmaktadır.

Kaynakça

- [1] Yıldırım, A. 2003. CNC baskı devre delme makinesi-maximus projesi. Hacettepe Üniversitesi, Mühendislik Fakültesi, Bitirme Projesi, Ankara.
- [2] Orhan, A. 1994. Bilgisayar kontrollü, programlanabilir baskılı devre matkap tezgâhı. Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 102s, İstanbul.
- [3] Mehmetcik, S. 2010. Optoelektronik yöntemlerle cisim tanıma. Marmara Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 69s, İstanbul.
- [4] Aslantaş, A. 2006. Ağaç kesitindeki yıllık halkaların görüntü işleme yöntemi ile incelenmesi. Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 60s, Isparta.

- [5] Kalaycı, A. G., Altuğ, H., Özkorucuklu, S. 2013. Çok Noktadan Otomatik Odaklama Kontrollü Sayısal Mikroskop. Teknik Bilimler Dergisi, 3(5), 13-18.
- [6] Altınkurt, Ö., Kahrıman, M. 2011. Gerçek Zamanlı Olarak, Anfis ile Renk Tabanlı Nesne Tespit ve Motorlu Sistem ile Takip Edilmesi. SDU Teknik Bilimler Dergisi, 1(1), 1-5.
- [7] Peker, M., Zengin, A. 2011. Gerçek Zamanlı Harekete Duyarlı Bir Görüntü Tanıma Sistemi. 6th International Advanced Technologies Symposium (IATS'11), 16-18 May, Elazığ, 92-97.
- [8] Çayıroğlu, İ., Şimşir, M. 2008. Pıç ve Step Motorla Sürülen Bir Mobil Robotun Uzaktan Kamera Sistemi ile Kontrolü. Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 24(1-2), 1-16.
- [9] Yılmaz, S. 2006. Bir robot kolu mekanizmasında adım motorları vasıtasıyla, verilen koordinatlara hareketin gerçekleştirilmesi. Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 142s, Sakarya.
- [10] Gökmen, Ü. 2008. Bir eğitim simülatoründeki çok sayıda adım motorunun pc ile hız ve konum kontrolünün gerçekleştirilmesi. Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 65s, İstanbul.
- [11] Uyar, E., Gören, A., Çetin, L. 2003. Birleşik Görsel Robot Sistemi ile Nesne Tasnifi. III. Ulusal Hidrolik Pnömatik Kongresi ve Sergisi, 4-7 Aralık, İzmir, 403-408.
- [12] Uygun, D. 2009. Hibrit Adım Motorlarının Eğitim ve Endüstriyel Uygulama Amaçlı Visual Basic Tabanlı Kontrol Yazılımı ve Uygulaması. 5th International Advanced Technologies Symposium (IATS'09), 13-15 May, Karabük.
- [13] Yıldız, M. 2012. BDT yazılımlarına ait verilerin, profil delme makinesi kontrol programında kullanılmak üzere dönüştürülmesi ve işlem süresinin optimizasyonu. Karabük Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 120s, Karabük.
- [14] Usta, M. A., Akyazı, Ö., Akpınar, A. S. 2011. Bilgisayar Üzerinden Bir Silah Kontrol Sistemi Prototipinin Tasarımı ve Gerçekleştirilmesi. 6th International Advanced Technologies Symposium (IATS'11), 16-18 May, Elazığ, 240-244.
- [15] Katagiri, H., Guo, Q., Wu, H., Hamori, H., Kato, K. 2015. Path Optimization for Electrically Inspecting Printed Circuit Boards with Alignment Marks. Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS), 18-20 March, Hong Kong, 979-984.
- [16] Katagiri, H., Qingqiang, G., Bin, W., Muranaka, T., Hamoric, H., Kato, K. 2015. Path Optimization for Electrical PCB Inspections with Alignment

- Operations using Multiple Cameras. *Procedia Computer Science*, 60, 1051-1060.
- [17] Lim, W. C. E., Kanagaraj, G., Ponnambalam, S. G. 2012. Cuckoo Search Algorithm for Optimization of Sequence in PCB Holes Drilling Process. ss 207-216. Sathiyamoorthy, S., Caroline, B. E., Jayanthi, J. G. ed. 2012. *Emerging Trends in Science, Engineering and Technology*. Springer, India, 796s.
- [18] Ismail, M. M., Othman, M. A., Sulaiman, H. A., Misran, M. H., Ramlee, R. H., Abidin, A. F. Z., Nordin, N. A., Zakaria, M. I., Ayob, M. N., Yakop, F. 2012. Firefly Algorithm for Path Optimization in PCB Holes Drilling Process. *International Conference in Green and Ubiquitous Technology*, 7-8 July, Jakarta - Indonesia, 110-113.
- [19] Narooei, K. D., Ramli, R., Rahman, M. N. A., Ibrahım, F., Qudeiri, J. A. 2014. Tool Routing Path Optimization for Multi-Hole Drilling Based on Ant Colony Optimization. *World Applied Sciences Journal*, 32(9), 1894-1898.
- [20] Dalavi, A. M., Pawar, P. J., Singh, T. P. 2016. Tool Path Planning of Hole-Making Operations in Ejector Plate of Injection Mould using Modified Shuffled Frog Leaping Algorithm. *Journal of Computational Design and Engineering*, 3(3), 266-273.
- [21] Srivastava, P. R. 2015. A Cooperative Approach to Optimize the Printed Circuit Boards Drill Routing Process using Intelligent Water Drops. *Computers and Electrical Engineering*, 43, 270-277.
- [22] Abdullah, H., Ramli, R., Wahab, D. A., Qudeiri, J. A. 2015. Simulation Approach of Cutting Tool Movement using Artificial Intelligence Method. *Journal of Engineering Science and Technology*, 10(Special Issue on 4th International Technical Conference (ITC) 2014), 35-44.
- [23] Alkaya, A. F., Duman, E. 2015. Combining and Solving Sequence Dependent Traveling Salesman and Quadratic Assignment Problems in PCB Assembly. *Discrete Applied Mathematics*, 192, 2-16. 11th Cologne/Twente Workshop on Graphs and Combinatorial Optimization (CTW 2012).
- [24] Shinde, N. K., Morade, S. S. 2015. PCB Inspection System using Image Processing. *International Journal of Science, Engineering and Technology Research (IJSETR)*, 4(4), 1009-1012.
- [25] Vinita, Kaushik, S. 2016. PCB Fault Detection by Image Subtraction Method. *International Journal of Scientific Engineering and Applied Science (IJSEAS)*, 2(1), 147-152.
- [26] Ancău, M. 2008. The Optimization of Printed Circuit Board Manufacturing by Improving the Drilling Process Productivity. *Computers & Industrial Engineering*, 55(2), 279-294.
- [27] Aoyama, E., Hirogaki, T., Katayama, T., Hashimoto, N. 2004. Optimizing Drilling Conditions in Printed Circuit Board by Considering Hole Quality: Optimization from Viewpoint of Drill-Movement Time. *Journal of Materials Processing Technology*, 155-156, 1544-1550. *Proceedings of the International Conference on Advances in Materials and Processing Technologies: Part 2*.
- [28] Farooque, S. N., Faizan, A. M., Shaikh, J., Pal, P. 2015. Automated PCB Drilling Machine with Efficient Path Planning. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(4), 108-110.
- [29] *Discrete and Combinatorial Optimization*. 2008. Index of /software/TSPLIB95/tsp. <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp/> (Erişim tarihi: 13.04.2017).
- [30] *Discrete and Combinatorial Optimization*. 2007. Optimal solutions for symmetric TSPs. <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/STSP.html> (Erişim tarihi: 13.04.2017).