

**Karma Evrimsel Algoritmalar Yardımı ile Temsili Olarak
Üretilmiş Bir Alan Üzerinde Sensörlerin Konumlandırılması
ve Parametrelerinin Tayini**

Proje No: 106E159

Prof . Dr. Haluk Rahmi Topcuoğlu
Yard. Doç.Dr. Murat Ermiş
Mesut Sifyan

MAYIS 2010
İSTANBUL

Önsöz

Bu projemiz, TÜBİTAK'ın desteği ile 1 Şubat 2007 tarihinde başlayan “Karma Evrimsel Algoritmalar Yardımı ile Temsili Olarak Üretilmiş Bir Alan Üzerinde Sensörlerin Konumlandırılması ve Parametrelerinin Tayini” başlıklı ve “106E159” numaralı bilimsel araştırma projesi olup, 30 Kasım 2009 tarihinde sonuçlandırılmıştır. Temsili olarak üretilen 3-boyutlu bir arazide, sensörlerin konumlandırılması ve sensör parametrelerinin tayinine yönelik olarak, çok ölçütlü fayda teorisi tabanlı yeni bir model, projemiz kapsamında tasarlanarak gerçekleştirilmesi tamamlanmıştır. Sensörlerin yerleştirilmesi problemi için, çok ölçütlü fayda teorisi kullanılarak tasarlanan model göz önüne alınarak, karma evrimsel algoritma tabanlı özgün bir çözüm geliştirilmiştir. Yapılan deneysel çalışmalar, çözümümüzün çeşitli askeri keşif uygulamalarında farklı ölçütler için etkin bir şekilde kullanılabileceğini göstermektedir. Sensör konumlandırma problemine ek olarak, çalışmamızda geliştirilen çok fonksiyonlu üç boyutlu arazi ve sensör simülatörü revize edilerek, kablosuz çoklu ortam sensör ağlarına yönelik, bağlı-kapsama problemi için de evrimsel algoritma tabanlı özgün bir çözüm sunulmuştur. Kablosuz çoklu ortam sensör ağlarına ilişkin olarak yapılan çalışmanın gereç, yöntem ve bulgularına ait detaylar projemizin kesin raporuna eklenmemiş olup, ekte verilen ve değerlendirme aşamasında olan makalelerde yer almaktadır.

İçindekiler

1	Giriş.....	1
2	Genel Bilgiler	4
2.1	<i>Proje Kapsamında Geliştirilen Sistemin Yazılım Mimarisi</i>	<i>4</i>
2.2	<i>Sensör Sistemlerine Yönelik Geliştirilen Algılama Yöntemleri.....</i>	<i>8</i>
2.2.1	Sensör Görünürlük Değerinin Hesaplanma Aşamaları.....	8
2.3	<i>Görüş Konisi Oluşturma Algoritması</i>	<i>8</i>
2.3.1	Yatay FOV Testi	9
2.3.2	Dikey FOV Testi	12
2.4	<i>Görüş Çizgisi (Line-of-Sight [LOS]) Algoritmaları.....</i>	<i>13</i>
2.4.1	Janus LOS Algoritması	14
2.4.2	Bresenham LOS Algoritması.....	15
3	Gereç ve Yöntem	16
3.1	<i>Çok - Ölçütlü Fayda Teorisi.....</i>	<i>16</i>
3.1.1	Görünürlük - Faydasının Hesaplanması	17
3.1.2	Gizlenme - Faydasının Hesaplanması.....	19
3.1.3	Maliyet - Faydasının Hesaplanması	20
3.2	<i>Karma Evrimsel Algoritma Tabanlı Çözüm Yöntemi</i>	<i>20</i>
3.2.1	Kromozom Yapısı	21
3.2.2	Başlangıç Popülasyonunun Oluşturulması.....	21
3.2.3	Sezgisel Olarak Sensör Konumlarının Belirlenmesi.....	22
3.2.4	Seçim Yöntemi	22
3.2.5	Varyasyon Operatörleri.....	22
3.2.6	Çaprazlama İşlemi.....	22
3.2.7	Yerel İyileştirme Prosedürü.....	24
3.2.8	Mutasyon Operatörleri	26
3.3	<i>Hareketli Sensörler için Genel Güzergah Planlama Adımı</i>	<i>27</i>
3.3.1	Olasılıksal Yol Haritasının Oluşturulması.....	27
3.3.2	Kontrol Noktası Seçimi.....	30
3.3.3	Kontrol Noktalarının Güzergahlara Atanması.....	30
3.4	<i>Hareketli Sensörler için Yerel Güzergah Planlama Adımı.....</i>	<i>31</i>
3.4.1	Kromozom Yapısı	31
3.4.2	Uygunluk Fonksiyonu.....	32
3.4.3	Başlangıç Popülasyonunun Oluşturulması	35
3.4.4	Çaprazlama Operatörleri	36
3.4.5	Mutasyon Operatörleri	36
3.4.6	Hareketli Engellere Karşı Çarpışma Kontrolü.....	37

4	Bulgular	38
4.1	<i>Statik Sensör Konumlandırma Problemine Yönelik Olarak Yapılan Deneyler</i>	<i>38</i>
4.1.1	Deney Tasarımına Yönelik Genel Bilgiler	38
4.1.2	Algoritmaya Özgü Parametreler ile Yapılan Deneyler	39
4.1.3	Probleme Özgü Parametreler ile Yapılan Deneyler	40
4.2	<i>Performans Değerlendirmesi ve Tartışma.....</i>	<i>43</i>
4.2.1	Algoritmanın Koşum Süresi Analizi:	43
4.2.2	Yerel Arama Yönteminde Sonlandırma Koşulunun Etkisi:	44
4.2.3	Yerel Aramada Alternatif Yöntemlerin Çözümüne Etkisi:	45
4.2.4	Katkı Tabanlı Çaprazlamada Sensör Sayısı Çarpanının Etkisi:	45
4.2.5	Çözümüne Yeni Sensör Eklemenin Çözümünün Kalitesine Etkisi:.....	46
4.2.6	Sensör Görüntüleme Modunun Çözümünün Kalitesine Etkisi:.....	47
4.2.7	Genel Güzergah Planlamaya Özgü Deneysel Çalışmalar.....	52
4.2.8	Yerel Güzergah Planlamaya Özgü Deneysel Çalışmalar.....	53
5	Sonuçlar	57
6	Kaynakça	58
7	Ekler.....	62
7.1	<i>EK-1 Proje Kapsamında Yapılan Yayınlar</i>	<i>62</i>

Şekil Listesi

Şekil 1: 3-boyutlu (a) ve 2-boyutlu (b) arazilerde sensörlerin kapsama alanları	1
Şekil 2: Sistemin yazılım mimarisi.....	4
Şekil 3: Sistem modüllerinin etkileşimleri	5
Şekil 4: (a) Diamond-square, (b) Hills, (c) Particle-deposition algoritmaları kullanılarak üretilen temsili araziler	6
Şekil 5: Temsili arazinin tel-kafes görüntüsü	6
Şekil 6: (a) Doğal ve yapay nesneler ile, (b) atmosfer koşullarının eklenmesi	7
Şekil 7: Önem haritası.....	7
Şekil 8: Field-of-view testi	9
Şekil 9: Yatay görüş alanı	9
Şekil 10: Yatay FOV testi	10
Şekil 11: Yatay FOV testi için çerçeveleyen dikdörtgenin oluşturulması.....	11
Şekil 12: Köşe noktalarının arazi içerisinde yer almasının sağlanması	11
Şekil 13: Dikey görüş alanı	12
Şekil 14: Sensörle nokta arasındaki eğim	13
Şekil 15: Kromozom yapısı.....	21
Şekil 16: Arazi üzerinde sensörlerin durumu.....	23
Şekil 17: Yerel arama evresinin adımları	25
Şekil 18: Erişilebilirlik testinde kullanılacak noktaların yarıçap yöntemiyle seçimi.....	28
Şekil 19: Olasılıksal yol haritası yöntemiyle üretilen noktalar ve bağlantıları	29
Şekil 20: Temsili araziden seçilen kontrol noktaları	30
Şekil 21: Bir sensör güzergahının kromozomdaki yapısı	32
Şekil 22: Güzergah üzerindeki bir doğru parçasına uygulanan yatay FOV testi	33
Şekil 23: Doğru parçasına uygulanan dikey FOV testi	33
Şekil 24: İki kontrol noktası arasında güzergah oluşturmada kullanılan birincil ve ikincil ara noktalar	36

Şekil 25: İterasyon sayısına göre karma evrimsel algoritmanın (a) toplam fayda değeri ve (b) 3 fayda kriterinin performans değerlendirmesi	39
Şekil 26: Görünürlük ağırlıklı misyonda dört etkenin toplam fayda değerine etkisi	40
Şekil 27: Gizlilik ağırlıklı misyonda dört etkenin toplam fayda değerine etkisi	41
Şekil 28: Maliyet ağırlıklı misyonda dört etkenin toplam fayda değerine etkisi	42
Şekil 29: Karma evrimsel algoritmanın çeşitli bölümlerinin görünürlük (a), gizlilik (b) ve maliyet (c) ağırlıklı misyonlarda koşum süresi yüzdeleri	44
Şekil 30: Görünürlük ağırlıklı misyonda yerel arama yönteminin sonlanma koşulu	44
Şekil 31: (a) Engebeli ve (b) Düzlük araziler için: (c,d) engel olmadan elde edilen kapsama alanları, (e,f) yağışlı hava koşullarında elde edilen kapsama alanları.	50
Şekil 32: Önem haritasının yer aldığı (a) Engebeli ve (b) Düzlük araziler için elde edilen kapsama alanları (c,d)	51

Tablo Listesi

Tablo 1: Deneysel çalışmamızda kullanılan sabit parametrelerin varsayılan değerleri	38
Tablo 2: Algoritmaya özgü parametre alternatifleri.....	39
Tablo 3: Görünürlük ağırlıklı misyonda toplam fayda değerinin varyans analizi	40
Tablo 4: İkinci grup deneylerde kullanılan parametre alternatifleri	41
Tablo 5: Gizlilik ağırlıklı misyonda toplam fayda değerinin varyans analizi	42
Tablo 6: Maliyet ağırlıklı misyonda toplam fayda değerinin varyans analizi.....	43
Tablo 7: Yerel arama metodunda alternatif yöntemlerin performans değerlendirmesi	45
Tablo 8: Algoritmada çeşitli sensör sayısı çarpanlarının performans değerlendirmesi	46
Tablo 9: Karma evrimsel algoritmada yeni sensör eklemenin çözüme katkısının karşılaştırılması.....	46
Tablo 10: Düzlük bir arazide farklı sensör görüntüleme modlarının performans değerlendirmesi.....	47
Tablo 11: Engelibeli bir arazide farklı sensör görüntüleme modlarının performans değerlendirmesi ...	48
Tablo 12: Düzlük bir arazide farklı sensör görüntüleme modlarının performans değerlendirmesi.....	48
Tablo 13: Engelibeli arazide farklı sensör görüntüleme modlarının performans değerlendirmesi	49
Tablo 14: Genel güzergah planlama algoritmasındaki sabit parametreler	52
Tablo 15: Genel güzergah planlama algoritmasındaki değişken parametreler ve olası değerleri	52
Tablo 16: Engelibeli arazide farklı askeri misyonlar için belirlenen genel güzergah algoritması parametreleri.....	53
Tablo 17: Düzlük arazide yapısında farklı askeri misyonlar için belirlenen genel güzergah algoritması parametreleri.....	53
Tablo 18: Yerel güzergah planlama algoritmasındaki sabit parametreler	54
Tablo 19: Yerel güzergah planlama algoritmasındaki değişken parametreler ve olası değerleri	54
Tablo 20: Engelibeli arazi yapısında farklı askeri misyonlar için belirlenen yerel güzergah algoritması parametreleri.....	55
Tablo 21: Düzlük arazi yapısında farklı askeri misyonlar için belirlenen yerel güzergah algoritması parametreleri.....	55
Tablo 22: Engelibeli arazi yapısında genel güzergah algoritmasının performans değerlendirmesi.....	56
Tablo 23: Düzlük arazi yapısında genel güzergah algoritmasının performans değerlendirmesi.....	56

Özet

Sensörlerin bir alan üzerine yerleştirilmesi ve parametrelerinin tayini, bilgisayarla görme, otonom sistemler, robotik ve askeri keşif gibi bir çok farklı disiplin için en temel araştırma alanlarından biridir. Projemizde, temsili olarak üretilen 3-boyutlu bir arazi için sensör sayısının ve tipinin seçimi, sensörlerin konumlandırılması, yatay ve dikey yönelme açısı gibi temel sensör parametrelerinin tayinini içeren çok ölçütlü fayda teorisi tabanlı yeni bir model tasarlanmış ve buna yönelik bir yazılım mimarisi gerçekleştirilmiştir. Geliştirdiğimiz modelimizdeki temel varsayım; bir askeri harekâta sensörlerin algılama davranışının birbirleriyle çelişen üç ölçüt arasında rasyonel bir ödünleşmeye dayanması gerektiğidir. Bu üç ölçüt: i) arazide görünen alanın maksimize edilmesi, ii) karşı güç tarafından tespit edilebilecek bilginin minimize edilmesi ve iii) konumlandırılan sensörlerin toplam maliyetinin minimizasyonudur.

Sensörlerin konumlandırılması problemi için önerilen teorik temellerin yanı sıra özgün bir karma evrimsel algoritma da geliştirilmiştir. Probleme özgü yeni operatörlerin tanımlanarak karma olarak kullanıldığı algoritmamızda, probleme özgü sezgisel yöntemlerle başlangıç popülasyonun oluşturulması, probleme özgü bilgiler içeren yeni varyasyon operatörleri (katkı tabanlı ve komşuluk tabanlı çaprazlama operatörleri) ve yerel arama teknikleri yer almaktadır. Deneysel çalışmamız, görünürlük-ağırlıklı, gizlilik-ağırlıklı ve maliyet ağırlıklı ölçütlerin kendi aralarında en uygun dengenin sonuçta sağlandığını göstermektedir. Yapılan deneysel çalışmalar, karma evrimsel algoritma tabanlı çözümümüzün çeşitli askeri keşif uygulamalarında farklı ölçütler için etkin bir şekilde kullanılabileceğini göstermektedir. Sensör konumlandırma problemine ek olarak, kablosuz çoklu ortam sensör ağlarına yönelik bağlı-kapsama problemi için, çalışmamızda geliştirilen çok fonksiyonlu 3-boyutlu arazi ve sensör simülatörü revize edilerek, kapsanan alanın maksimizasyonu ve sensör maliyetinin minimizasyonunu hedefleyen evrimsel algoritma tabanlı özgün bir çözüm de sunulmuştur.

Projemizde, hareketli sensörlerin güzergah planlama problemi için, çıkılabilecek en yüksek eğim ve en düşük dönüş açısı gibi farklı kısıtların dikkate alındığı, yeni bir karma çözüm de geliştirilmiştir. Bu çözüm, genel ve yerel güzergah planlama olmak üzere iki aşamadan oluşmaktadır. Genel güzergah planlama aşaması olasılıksal yol haritası yöntemiyle üretilen bağlantırlık çizgesi ile sensörlerin güzergahlarını oluşturan kontrol noktalarının belirlenmesini ve bu noktaların bir fonksiyon yardımıyla sensör güzergahlarına atanmasını içerir. Yerel güzergah planlama aşamasında ise kontrol noktaları arasındaki ara noktaları belirleyen evrimsel algoritma tabanlı bir yöntem geliştirilmiştir.

Anahtar Kelimeler:

Karma Evrimsel Algoritmalar, Sensör Planlanması, Çok-Ölçütlü Eniyileme, Çok-Ölçütlü Fayda Teorisi, Yerel Arama, Olasılıksal Yol Haritası

Abstract

Deploying and configuring multiple sensors for acquisition of a given area is one of the fundamental research areas of researchers from various fields including computer vision, autonomous systems, robotics and military scouting. In this project, we presented a novel multi-attribute utility theory based model and a framework for determining the types and number of sensors, locating the selected sensors and setting their orientational sensor-specific parameters including heading and tilt angles on a synthetically generated 3-D terrain with multiple objectives. Our model relies on rational trade-off between the three conflicting objectives which are maximizing the coverage area while maintaining the maximum stealth, and minimizing the total acquisition cost of deploying the sensors.

In addition to theoretical foundations, we developed a new hybrid evolutionary algorithm for the sensor placement problem. We incorporated new and specialized operators for hybridization, including problem-specific heuristics for initial population generation, intelligent variation operators (Contribution-Based Crossover operator and Proximity-Based Crossover operator) which comprise problem specific knowledge, and a local search phase. The experimental study validates finding the optimal balance among visibility-oriented, stealth-oriented and cost-oriented objectives. The obtained results significantly indicate that our hybrid evolutionary algorithm can be effectively and robustly deployed in various military scouting missions with different objectives. Our multi-functional 3-D terrain/sensor simulator for the sensor placement problem was modified for the connected-coverage problem in wireless multimedia sensor networks, which was solved with a novel evolutionary algorithm based solution that aims to maximize the coverage of the terrain and to minimize the cost of sensors.

In this project, we also developed a new hybrid solution for path planning of moving sensors with different access characteristics including different slopes of climbing and different minimum turning angles. Our solution is a two-phase technique where the global path planning phase determines control points of sensors' paths with the help of the connectivity graph generated by probabilistic road map (PRM) method and a function for selecting the control points. On the other hand, we developed evolutionary algorithm based method for the local path planning phase, which is for determining the intermediate points that are between control points in order to complete the paths.

Keywords:

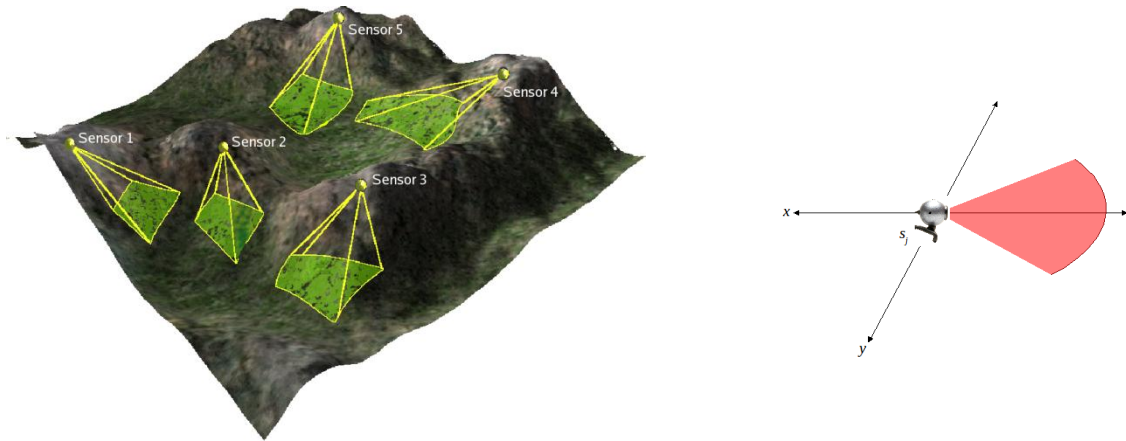
Sensor Planning, Multi-objective Optimization, Multi-Attribute Utility Theory, Hybrid Evolutionary Algorithms, Local Search, Probabilistic Road Maps.

1 Giriş

Bu bölümde, öncelikle sensör konumlandırma problemine yönelik olarak literatürde yer alan farklı alanlardaki çalışmalar kısaca özetlenmiştir. Projemiz kapsamında, sensör konumlandırma probleminin güncel bir uygulama sahası olarak kablosuz çoklu ortam sensör ağlarında, kapsama alanını en fazlaştıracak, maliyeti en aza indirecek ve bağlı olma kısıtını da sağlayabilecek şekilde düğümlerinin yerleştirilmesi ve yönlendirilmesi problemi de ele alınarak modellenmiş ve çözüm yaklaşımı geliştirilmiştir. Bu nedenle, kablosuz çoklu ortam sensör ağlarına yönelik olarak literatürde yapılan çalışmalara da bu bölümde yer verilmiştir.

İleri düzeyde algılayıcıların üretim maliyetinin düşmesi ve uzaktan erişilebilen algılayıcı teknolojisinin birçok alanda yaygınlaşmasına rağmen, algılayıcılar vasıtasıyla daha sağlıklı ve kapsamlı bilgilerin toplanması hala zor bir problem olarak karşımıza çıkmaktadır. Verilen bir alanı kapsamak amacıyla, farklı algılama özelliği ve maliyeti olan çoklu sensörlerin konumlandırılması oldukça önemli ve gerekli bir birleşik optimizasyon problemidir [(WANG, 2008), (MARENGONI, 2000), (TEZCAN, 2008)]. Askeri uygulamalarda, keşif, gözetim ve hedef tayini operasyonlarında gözetim altındaki bir alandan bilgi toplamak amacıyla kullanılan sensör platformları hayati önem taşımaktadır (RISEMAN, 1996). Aynı şekilde, düşman kuvvetlerinin konuşlandığı bölgeleri algılamak için, belirli bir alanın maksimum kapsanacak ve gizliliği de sağlanacak şekilde sensörlerin yerleştirilmesi beklenir.

Askeri uygulamaların haricinde de, sensör konumlandırma problemi birçok alan için temel araştırma konuları arasında yer almaktadır. Bir oda veya bölgeyi kapsamak için gerekli sensör sayısının belirlenmesi hesapsal geometri literatüründe sanat galerisi problemi olarak isimlendirilmektedir (O'ROURKE, 1987). Sanat Galerisi Problemi iki boyutlu alanlar için optimal şekilde çözülebildiği halde, kapsanacak alanın 3-boyutlu olduğu durumlarda problem NP-zor olmaktadır (O'ROURKE, 1992) (Şekil 1). 3-boyutlu Sanat Galerisi Probleminin ancak yaklaşık çözümleri bulunabilmektedir [(MARENGONI, 2000), (HORSTER, 2006)].



Şekil 1: 3-boyutlu (a) ve 2-boyutlu (b) arazilerde sensörlerin kapsama alanları

Kapsama problemi literatürde çok değişik uygulama alanları için çalışılmıştır (HALL, 1988). Video algılayıcıları için kapsama problemi, kapsanacak alanın en az bir gözlemci tarafından kapsanması amacıyla, ihtiyaç duyulacak gözlemci sayısını ve yerlerini belirlemeye yönelik geleneksel Sanat Galerisi Problemi ile yakından ilişkilidir (O'ROURKE, 1987). Video sensör planlama çabaları daha çok odaklanma, görünürlük, bakış açısı, kısıtlama alanları gibi değişik kısıtlamaların uygulanmasına yöneliktir (TARABANIS, 1995).

Genel kamera yerleştirme probleminde, görüntülenecek bir kat planındaki hedef, göreve özgü kısıtların en düşük maliyetle sağlandığı bir kamera yerleşimi sağlamaktır (ERDEM, 2004). Ayrık bir problem uzayında ikili optimizasyon yöntemi, bu probleme bir çözüm olarak verilmiştir. Dinamik olarak hareket eden nesnelerin birbirlerini engelleyebildiği durumları içeren senaryoların analiz edilerek en uygun sensör konfigürasyonlarının belirlenmesi problemi üzerinde de çalışmalar yapılmıştır [(MITTAL, 2004), (MITTAL, 2006)]. Görünürlük ve statik kısıtların entegre edildiği sensör planlama problemi hakkında Mittal yeni bir mimari önermiştir.

Birden çok nesnenin ve bölgenin 3-boyutlu ortamda görüntülenmesini sağlamak için, sensörlerin görüş açısının otomatik olarak ayarlanması, bilgisayarla görmede önemli araştırma alanlarından biridir [(STOMAS, 1998), (REED, 2000), (CHEN, 2002), (CHEN, 2004)]. Bu tip problemlerde, bir grup görüş açısı ve sensör konumlandırması, istenilen tüm özellikleri örneklemek için tasarlanmaktadır. Benzer bir şekilde, otonom hareketli robotlar için navigasyon probleminde, robotların engelleri ve düzlük araziye ayırt edebilmesi için kendi aralarında işbirliği yapmasına olanak sağlayacak bir grup sensörle donatılmıştır. Robotik alanında sensör planlama problemi, sensör tipinin, sensör konfigürasyonunun ve görevlerinin tayinini içerir. Bu problemle ilgili çeşitli yöntemlerin kullanıldığı birçok çalışma literatürde yer almaktadır [(KRISTENSEN, 1997), (MIHAYLOVA, 2002), (SPLETZER, 2002)]. Çok ölçütlü fayda teorisini esas alan, birlikte çalışan sensörlerin planlanmasına yönelik bir sistem de literatürde yer almaktadır (COOK, 1996). Bu sistemde, görme, sensör planlama ve çok-araçlı planlama üzerinde yapılan bazı çalışmalar birleştirilmiştir.

Sensör konumlandırma probleminin güncel bir uygulama sahası olan kablosuz çoklu ortam sensör ağları (Wireless Multimedia Sensor Networks-WMSN), video yada ses gibi çoklu ortam verilerini alabilen ve bir merkez düğüme inderebilen sensör düğümlerinden oluşan yeni bir sensör (algılayıcı) ağ türüdür (AKYILDIZ, 2007). Bağlı-kapsama konsepti WMSN sensörlerinin yerleştirilmesi için en önemli konulardan biridir. WMSN düğümleri öyle bir yerleştirilmelidir ki kapsanan alan en fazlalanmalı ve düğümler birbirine bağlı bir iş ağı oluşturabilmelidirler. Bu problem, sensör konumlandırma probleminin güncel bir uygulaması olduğundan, bu probleme yönelik çalışmaları da aşağıda kısaca özetlemektediriz.

Son yıllarda, çoklu ortam algılayıcılar ile kablosuz iletişim ünitelerinin bir araya getirilmesi ile ortamdaki çoklu ortam verilerini edinebilen WMSN'lerin hayata geçmesi mümkün olmuştur. Video sensör ağları için kapsama ve yönlendirme problemiyle ilgili incelemeler de literatürde yer almaktadır (TEZCAN, 2008). Bu çalışmada, iki boyutlu ortamda en yüksek faydayı sağlayan, engellerin etkisini en aza indirgeyen ve ilgili alan hakkında toplanan bilginin niteliğini arttıran yönelme parametrelerini bulmak için yeni bir algoritma geliştirilmiştir. Kablosuz algılayıcı ağları için, az sayıda sensörün düşük enerji tüketimiyle bir bölgeyi kapsama problemi de literatürde yer almaktadır [(WANG,2008), (JIA, 2009)].

Kapsama sorunu WMSN için önemli olmasına karşın, sensörlerin birbirlerine bağlı olması da bir iş ağı için oldukça önemlidir. Kapsama alanı gibi bağlı olmak da algılayıcıların yerleştirilmesi ile yakından ilişkilidir. Bu yüzden kapsama, bağlı olma ile birlikte değerlendirilmektedir (GHOSH, 2008). Bağlı-kapsama kavramı önce WSN'ler için ele alınmıştır. PEAS, WSN'ler için bağlı-kapsama kavramını ilk ele alan çalışmalardan biridir (YE, 2003). Fakat PEAS, bağlı-kapsamayı analitik biçimde garantileyememektedir. Xing tarafından kapsama ve bağlılık arasında geometrik bir analiz sunulmuştur (XING, 2005). Bir başka çalışmada (ZHOU, 2004), alanın her noktasının en az k tane algılayıcı ile kapsanacak ve algılayıcıların işağı olarak bağlı olabilecek şekilde (bağlı-k-kapsama) aktif algılayıcı alt kümelerinin seçilmesi problemi tartışılmıştır. Yang, WSN'de bağlı-k-kapsama için kritik koşulları çalışmıştır (YANG, 2008).

Ayrıca, projemiz kapsamında, 3-boyutlu ortamda çoklu hareketli sensör platformları için güzergah planlama algoritmaları da tasarlanmıştır. Bu tip ortamların güzergah planlamasında engellere ve diğer

hareketli sensörlere çarpma kontrolünün yanı sıra, hedeflenen yere ulaşmada süre (mesafe), gerçekçilik (düzgünlük) ve arazinin görünürlüğü gibi çeşitli kriterler göz önünde tutulmuştur. Bu nedenle, güzergah planlama problemine yönelik literatür özeti de aşağıya eklenmiştir.

Güzergah planlama, robot navigasyonu, bilgisayar animasyonları, havacılık ve askeri simülasyonlar gibi birçok alanda karşımıza çıkan temel problemlerden biridir [LA VALLE, 2006]. Robotik uygulamalarında güzergah planlama problemindeki amaç; başlangıç ve hedef konfigürasyonları arasında çevredeki engellerden arındırılmış, belirlenen kriterlere göre en uygun güzergahı oluşturacak bir dizi işlemin gerçekleştirilmesidir. Güzergah planlama faaliyeti işlemsel olarak çok yoğun ve konfigürasyon uzayı boyutlarına bağlı olarak üstel artan karmaşıklık düzeyine sahiptir. Bu nedenle üç boyutlu çalışma ortamında kesin çözümü bulma problemi “NP-HARD” zorluk derecesine sahip bir problemdir (XAVIER, 1992). Güzergah planlama probleminin karmaşıklığı serbestlik derecesiyle beraber probleme dahil sensör (robot) sayısına da bağlıdır. Çoklu güzergah planlama probleminde, hareketli sensörler arasında iletişim, dinamik ortamda çarpışma kontrolü gibi hesaplamalar genel güzergah planlamaya göre probleme fazladan yük getirmektedir.

Genel güzergah planlama problemiyle ilgili literatürde Voronoi diyagramlarla güzergah planlama (BHATTACHARYA, 2007), hücre dağılması (cell decomposition) (ZHU, 1991) ve örnekleme tabanlı güzergah planlama [(LATOMBE, 1991), (KAVRAKI, 1996), (LA VALLE, 1998), (BARRAQUAND, 1990)] gibi farklı yöntemlerin kullanıldığı birçok çalışma bulunmaktadır. Çalışma ortamı hakkında genel bilgiye sahip olmak karmaşık ve fazla süre gerektiren bir işlem olduğu için işlem süresini kısaltacak olasılıksal yol haritası (probabilistic roadmap - PRM) (KAVRAKI, 1996) veya hızlı keşifte rasgele bağlantılar (rapidly exploring random tree - RRT) (LA VALLE, 1998) gibi örnekleme tabanlı teknikler kullanılabilir.

Olasılıksal yol haritası metoduyla üretilen güzergahlar, mesafe kriterine göre optimum çözümü vermemesine rağmen bu yöntemin çalışma ortamının engellerden arındırılmış konfigürasyon alanını veren ve çoklu güzergah planlaması için sorgulama imkanı sağlayan en hızlı algoritmalarından biridir.

Kapsamaya dayalı güzergah planlama, güzergah planlama probleminin özel bir çeşididir. Kapsamaya dayalı güzergah planlama ile ilgili oda temizleme (LUO, 2002), mayın tarama (NAJJARAN, 2000), otonom su altı kapsama (HERT, 1996) ve zirai hasat toplama (OLLIS, 1996) gibi birçok robotik uygulama mevcuttur. Bu uygulamaların birçoğunda, kapsama problemi ortamdaki tüm noktalardan geçmeyi hedeflemektedir. Kapsamaya dayalı güzergah planlama problemi ile ilgili çalışmalarda yukarıda belirtilen yöntemlerin haricinde “artificial potential field” (PIRZADEH, 1990) ve sinir ağları (YANG, 2004) gibi yöntemler de kullanılmıştır. Bu çalışmaların bir kısmı sadece tek robotlu sistemlere göre tasarlanmışken bir kısmı çoklu robot sistemlerini göz önüne almıştır. Ayrıca bu tür yöntemler [(ACAR, 2001), (TIAN, 2005)], kullanılan çalışma ortamlarının iki boyutlu olması ve bazı uygulanabilirliği düşük varsayımlar içermesi nedeniyle 3-boyutlu eğimli arazilerde uygulanması mümkün değildir.

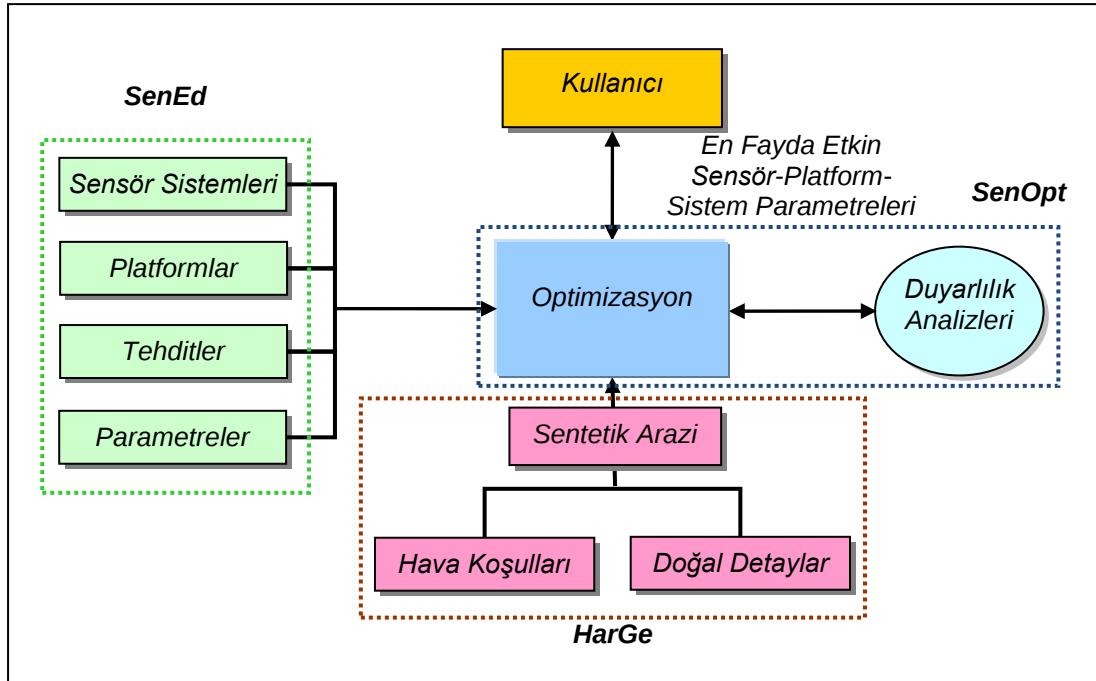
Yukarıda belirtilen nedenlerden ve çalışmamızdaki hedefler arasında mesafe haricinde çeşitli kriterlerin olması sebebiyle, çalışmamızda çoklu hareketli sensör platformları için güzergah planlama problemi için yeni bir karma evrimsel algoritma tabanlı mimari geliştirilmiştir. Bu mimaride arazinin görünürlüğü, güzergahların düzgünlüğü, maliyeti ve erişilebilirliği hedefleri dikkate alınmıştır. Ayrıca, sabit ve hareketli engellere (diğer sensor platformları) karşı çarpışma kontrolü de çalışmamızda yer almaktadır.

2 Genel Bilgiler

Bu araştırma projesinde, temsili olarak üretilen bir arazi bölgesi için sensörlerin konumlandırılması, sensör açısı, eğimi gibi temel sensör parametrelerinin tayini ve bir platforma/araca bağlı hareket kabiliyeti olan sensörler için devriye güzergahlarının belirlenmesi hedeflenmiştir. İlk aşamada, 3-boyutlu temsili arazi verilerinin üretilmesine ve gerçek yaşamdaki farklı çevre koşullarının benzetiminin yapılmasına olanak sağlayacak bir arazi oluşturma simülasyonu ve deney ortamı (testbed) oluşturulmuştur. Bu bölümde, geliştirilen 3-boyutlu simülasyona ait yazılım mimarisi ile 3-boyutlu kapsama analizinde kullandığımız bilgisayar görme yöntemleri hakkında bilgi verilecektir.

2.1 Proje Kapsamında Geliştirilen Sistemin Yazılım Mimarisi

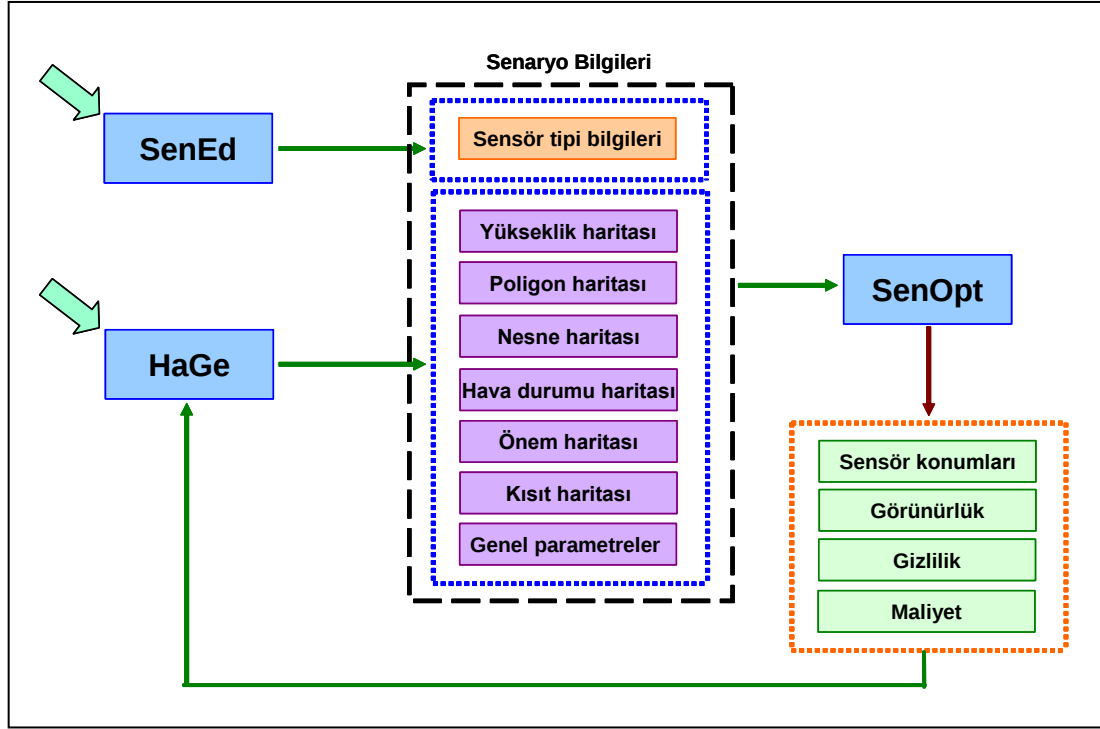
Temsili olarak üretilen bir arazi bölgesi için sensörlerin konumlandırılması, temel sensör parametrelerinin tayini ve bir platforma bağlı sensörler için hareket güzergahlarının belirlenmesini gerçekleyecek olan sistemimiz incelendiğinde, yazılım mimarisi açısından üç ana modülden oluştuğu görülmektedir: 1) Temsili arazi verilerini ve üzerinde yer alan doğal detaylar (fundalık, göl, vadi, vb.) ile hava koşullarının oluşturulduğu Harita Geliştirme modülü (*HarGe*); 2) Genel sistem parametreleri, konumlandırılacak farklı sensör sistemleri (kamera, infra-red, gece görüş, mikro-dalga radar, vb.) ile tehditlere ait verilerin tanımlandığı Senaryo Editör modülü (*SenEd*); 3) Sabit sensörlerin konumlandırılması ve parametrelerinin tayini için eniyileme algoritmalarının kullanıldığı ve duyarlılık analizlerinin de yapıldığı Sensör Optimizasyon modülü (*SenOpt*). Sistemin yazılım mimarisi Şekil 2’de verilmiştir.



Şekil 2: Sistemin yazılım mimarisi

Sistemimizde yer alan ana modüller arasındaki etkileşim, Şekil 3’te görüldüğü gibi gerçekleşmektedir. İlk olarak, *HarGe* modüle vasıtasıyla temsili arazi verisi ve üzerine örtüştürülecek doğal detaylar (fundalık, göl, vadi, vb.) ile hava koşulları oluşturulmaktadır. Daha sonra, konumlandırılacak farklı sensör sistemleri (kamera, infra-red, gece görüş, mikro-dalga radar) ve platformlara ait bilgiler,

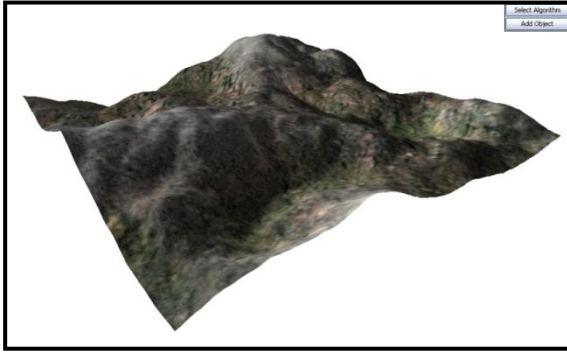
tehditler ve genel parametreler bir senaryo çerçevesinde *SenEd* modülü vasıtası ile modele dâhil edilmektedir (senaryo üretim safhası). Üretilen senaryo bilgileri, *SenOpt* vasıtasıyla karma evrimsel algoritmalar kullanılarak global fayda işlevine göre eniyilenmekte ve sonuçta sensörlerin uygun şekilde konumlandırılması gerçekleştirilmektedir.



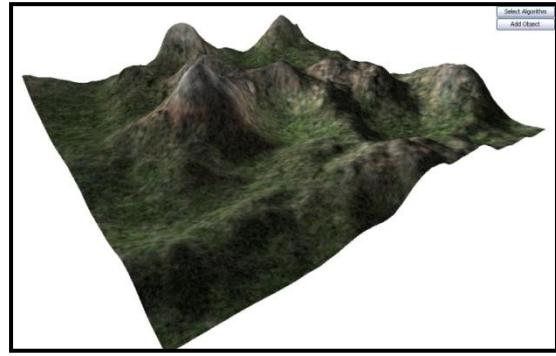
Şekil 3: Sistem modüllerinin etkileşimleri

Geliştirdiğimiz sistem, yukarıda belirtilen amaca erişmek için temsili ortam ve sensör sistemlerinin detaylı modellemesine olanak sağlar. Her ne kadar bir coğrafi bilgi sistemi yazılımı oluşturmak amaç olmasa da, optimizasyon sonuçlarının gerçeği yansıtması için çok detaylı bir ortam modelleme arayüzü hazırlanmıştır. Bu arayüz vasıtasıyla, 3-boyutlu arazi, arazi üzerindeki detaylar ve hava koşulları modellenenmektedir.

- Harita Üretici (HarGen) olarak adlandırabileceğimiz modül ile, temsili arazinin değişik çözünürlüklerdeki haritası, tel-kafes görüntüsü veya yükseklik bilgisi görüntülenebilir (Şekil 4-Şekil 5). Bu şekillerdeki araziler Diamond-square (FOURNIER, 1982), Hills (NYSTROM, 2000), ve Particle-deposition arazi geliştirme algoritmaları ile üretilmiştir.
- Temsili arazinin istenilen bölgelerine doğal ve yapay nesneler istenilen oranda eklenebilir (Şekil 6.a).
- Atmosfer koşulları, görüş mesafesi (sis miktarı), yağış tipi ve miktarı tanımlanıp görselleştirilebilir (Şekil 6.b).
- Önem haritası ve kısıt haritası oluşturulabilir (Şekil 7).



(a)

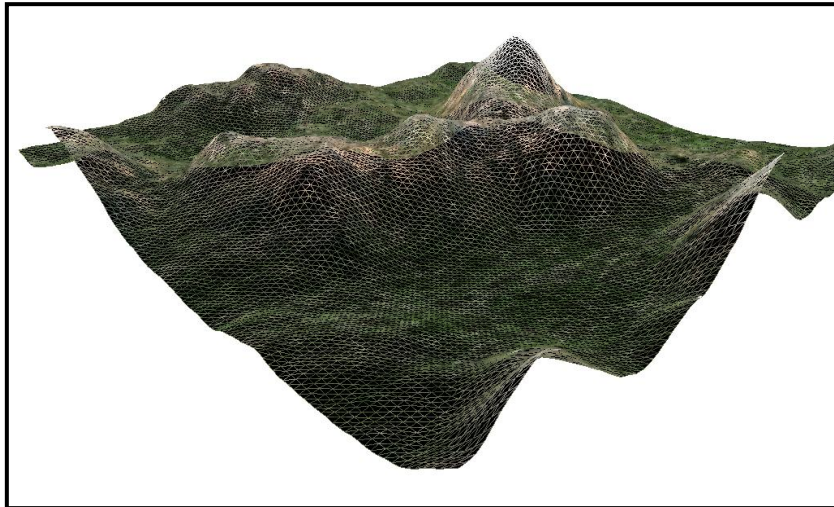


(b)

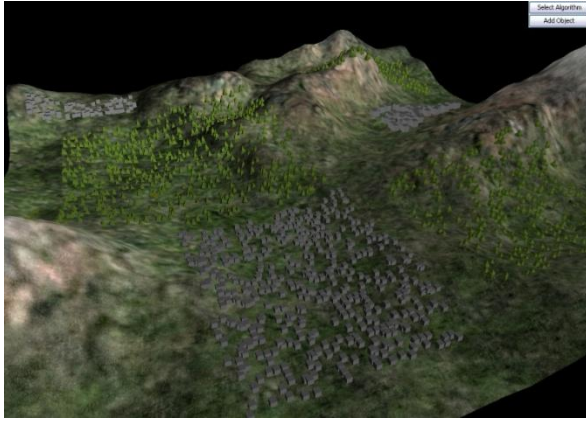


(c)

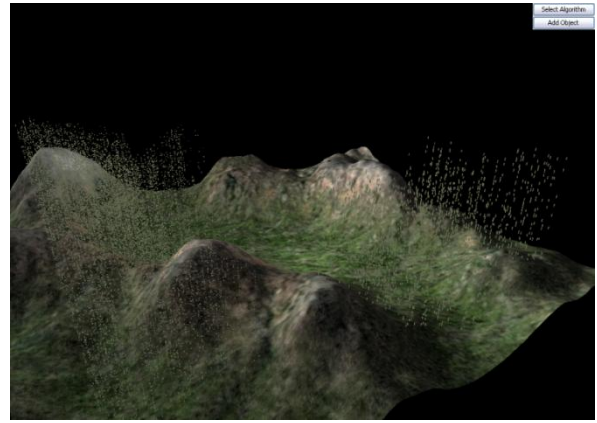
Şekil 4: (a) Diamond-square, (b) Hills, (c) Particle-deposition algoritmaları kullanılarak üretilen temsili araziler



Şekil 5: Temsili arazinin tel-kafes görüntüsü

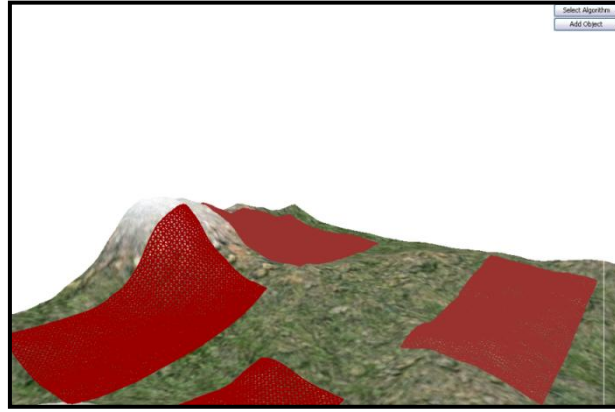


(a)



(b)

Şekil 6: (a) Doğal ve yapay nesneler ile, (b) atmosfer koşullarının eklenmesi



Şekil 7: Önem haritası

Senaryo tanımlama sistemi (*SenEd*), temsili arazinin savunması ve/veya devriye güzergahının belirlenmesine yönelik bir senaryonun *HarGe*'de belirtilen sınırlamalar, nesneler çerçevesinde hazırlanabilmesi için geliştirilmiştir.

Sensör sistemleri, bir ya da daha fazla bileşenin hiyerarşik bir yapı altında bir araya getirilmesiyle oluşturulmaktadır. Sensör sistemleri, koyulabilirlik haritasına ek olarak kendi kullanılabilirlik ve koyulabilirlik kısıtlarına da sahiptir.

Temsili ortam ve sensör sistemlerinden oluşturulan senaryo, optimizasyon modülüne (*SenOpt*) girdi olarak verilmektedir. *SenOpt* kapsamında geliştirilen sistemde maliyet/etkinlik optimizasyonu, etkinlik optimizasyonu ve maliyet optimizasyonu olarak iki alt grubu ayrılır. Etkinlik optimizasyonu modülü, sensör sistemlerinin verilen bir bütçeyi aşmadan etkinliği maksimize edecek en uygun tip, sayı ve konumlarını bulur. Benzer şekilde maliyet optimizasyonu modülü, sensör sistemlerinin verilen bir etkinlikte korunma sağlarken maliyeti minimize edecek en uygun tip, sayı ve konumlarını bulmayı sağlar.

Optimizasyon neticesinde elde edilen sonuçların farklı başarımlar ölçütlerine göre duyarlılık analizlerini gerçekleştirmek amacıyla deneysel tasarımlar dizayn edilmiş, farklı sentetik ortamlarda bu deneyler tekrarlanarak geliştirilen yöntemin etkinliği ölçülmüştür.

2.2 Sensör Sistemlerine Yönelik Geliştirilen Algılama Yöntemleri

Proje kapsamında gerçekleştirilen arazi geliştirme algoritmalarından birisi yardımı ile üretilen arazinin toplam görünürlük değeri, araziye oluşturan poligonların görünürlük değerlerinin poligonların önem değerleri ile ağırlıklı olarak toplanmasıyla bulunmaktadır. Toplam gizlilik bileşeninde de sensörlere yönelik olarak benzer hesaplamalar olmasından dolayı raporun bu kısmında sadece toplam görünürlük ile ilgili açıklamalar yer almıştır. Toplam görünürlük değeri, çözümü oluşturan her bir sensörün arazi üzerindeki poligonlara yönelik görünürlük hesaplamasının yapılıp, tüm sensörler için bu görünürlük değerlerinin birleştirilmesi ile elde edilir.

2.2.1 Sensör Görünürlük Değerinin Hesaplanma Aşamaları

Sensör algılama probleminin çözümüne yönelik olarak, öncelikle seçilen sensörün algıladığı arazi bölümü ve bu arazi bölümü içerisinde yer alan poligonlardan oluşan küme oluşturulur. Daha sonra da bu kümede yer alan her bir poligonun görünürlük değeri hesaplanır. Spesifik olarak, görünürlük değerini hesaplamaya yönelik yaptığımız çalışma iki aşamadan oluşmaktadır:

Görülebilir nitelikteki poligon listesini oluşturmaya yönelik *görüş konisi (view-cone) oluşturma algoritma aşaması*

Görüş konisi içerisindeki her bir poligonun görülebilirlik değerini bulmaya yönelik olan *görüş çizgisi (line-of-sight, LOS) algoritma aşaması*. (Proje kapsamında Bresenham ve Janus LOS Algoritmaları gerçekleştirilmiş olup detayları aşağıda verilmiştir).

Gerçeklenen bu aşamaların detayına geçmeden önce, çalışmamız kapsamında belirlenen bazı varsayımlar aşağıda listelenmiştir:

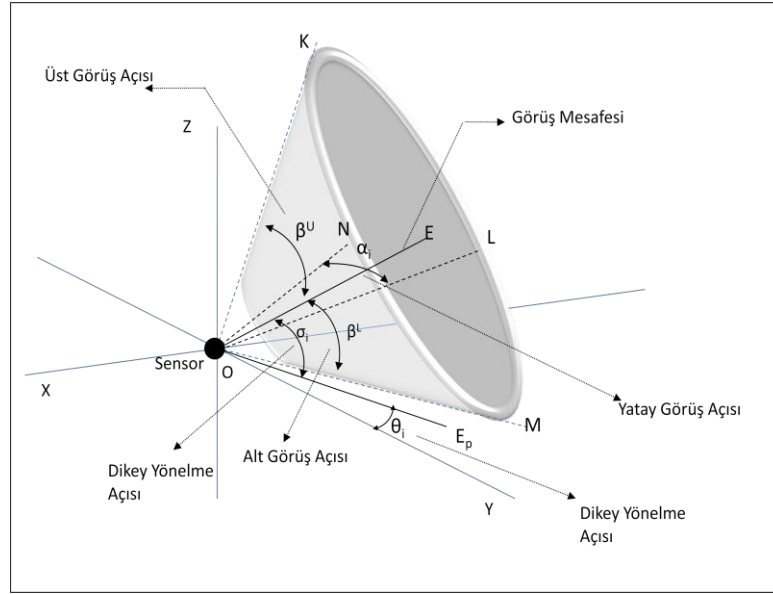
- Her bir sensör, seçilen poligonun ağırlık merkezi üzerine yerleştirilir.
- Sensör, seçilen poligonun ağırlık merkezin üzerinde sabit yükseklikte (GUI veya konfigürasyon dosyamızda belirlenen yükseklikte) bir platforma yerleştirilmiştir. Bu platform sayesinde dikey yönelme açısı (tilt açısı) daha geniş bir aralıktan atanabilecektir.
- Koordinat sistemimiz yatay XZ ve yükseklik Y olacak şekilde tanımlanmıştır.
- Görüş çizgisi (LOS) algoritmalarında her bir poligonun görünürlük değeri, poligonun üç köşe noktası ve ağırlık merkezinin görünürlük değerlerinin ortalaması ile bulunmaktadır.
- Çalışmamızda tespit, teşhis ve tanıma katmanları olup, sensörlerin karakteristikleri bu üç katmana göre verilmiştir. Optimizasyon adımında ise, hangi katman seçilirse ona yönelik parametreler göz önüne alınacaktır.
- Çalışmamızda, görme konisi oluşturma sürecini içeren aşamalarda (Yatay FOV testi, dikey FOV testi), gereksiz tekrar oluşturacak hesaplamaları azaltmak için, her bir testi geçen poligonları saklamak yerine gerçekleştirmemizde, testi geçen arazi bölümü içinde yer alan köşe noktaları ve ağırlık merkezi noktaları saklanmaktadır.

2.3 Görüş Konisi Oluşturma Algoritması

Çalışmamızda gerçekleştirilen görüş konisi oluşturma algoritması iki aşamadan oluşmaktadır:

1. En geniş yatay görüş alanı içinde kalan sensörlere yönelik olarak yatay görüş alanı (field-of-view - FOV) testi yapılarak, yatay FOV testinden geçen arazi parçası (poligonlardan oluşan bir set) belirlenir (Şekil 8).

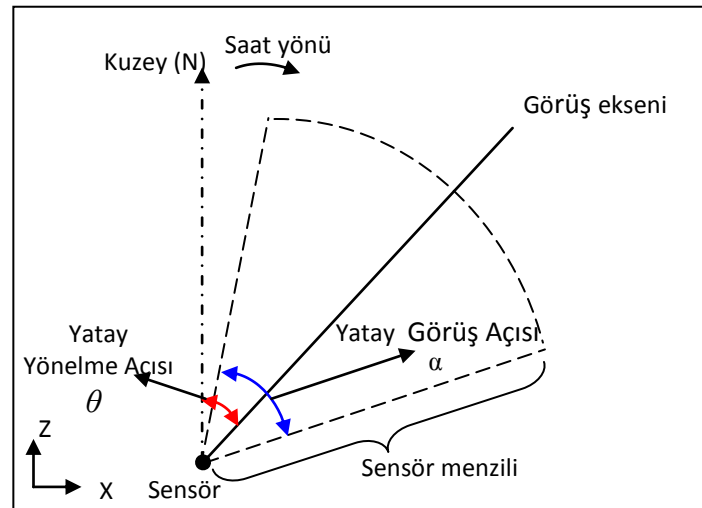
2. Yatay FOV testinden geçen poligonlara dikey FOV testi uygulanarak, görüş konisinde yer alan poligon kümesi belirlenmiş olur.



Şekil 8: Field-of-view testi

2.3.1 Yatay FOV Testi

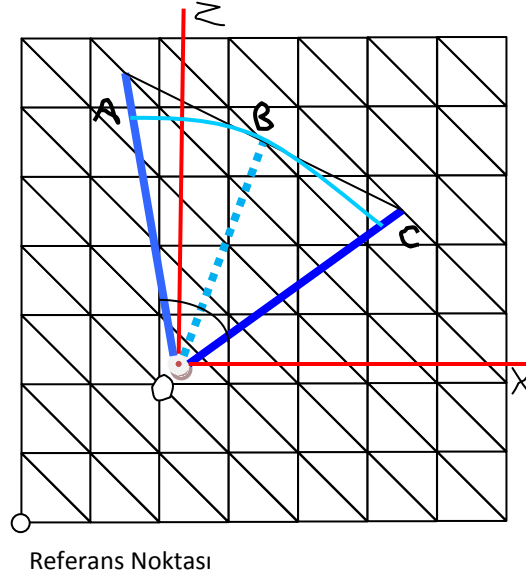
Yatay FOV testinin temel amacı, XZ koordinat eksenine uygun bir filtreleme sağlamak olup bu fonksiyonumuz için gereken girdiler; sensörün konumsal parametrelerinin bir kısmı (üzerinde bulunduğu poligon numarası ve yatay yönelme açısı [θ]) ve davranışsal parametrelerinin bir kısmı (sensörün tipi, seçilen katmana uygun olarak yatay görüş mesafesi [Δ] ve yatay görüş açısı [α]), arazinin boyutu ve poligonları tuttuğumuz dizinin adresidir (Şekil 9). Fonksiyonun çalışması tamamlandığında ise FOV testini geçen noktalar kümesi çıktı olarak döndürülmektedir. Aşağıda yatay FOV testi adımları özetlenmiştir.



Şekil 9: Yatay görüş alanı

Adım 1: Açılarla İlgili Hesaplamalar

Aşağıda verilen arazi poligonlar olarak gösterilmiş olup, XZ düzlemini ifade etmektedir (Şekil 10). Yatay FOV testi de XZ düzlemini baz aldığı için yüksekliklere yer verilmemiştir. Burada sensör O noktasına yerleştirilmiş olup, ZOB açısı sensörün yatay yönelme açısını (θ), AOC açısı ise sensörün yatay görüş açısını (α) vermektedir.



Şekil 10: Yatay FOV testi

Buna göre, XOB açısı $\hat{XOB} = (450 - \theta) \bmod 360$ formülü ile hesaplanır. Daha sonraki adımlarda gereken XOA ve XOC açıları da benzer şekilde aşağıdaki formüller yardımı ile hesaplanır:

$$\hat{XOA} = (\hat{XOB} + \frac{\alpha}{2}) \bmod 360 \quad \text{ve} \quad \hat{XOC} = (\hat{XOB} - \frac{\alpha}{2}) \bmod 360 \quad (1)$$

Adım 2: A, B ve C Noktalarının Koordinatlarının Hesaplanması

Arazinin en sol ve en alt noktası (referans noktası) (0,0) noktası kabul edilip, A, B ve C noktalarının koordinatları hesaplanır. Burada O noktasına konan sensör için $|OB|$, sensörünün yatay görüş mesafesini vermektedir, $|OB|=L$. Sensörün hangi poligon üzerinde olduğu bilindiğinden, O.x ve O.z değerleri o poligonun ağırlık merkezinin koordinatlarını vermektedir. Buna göre, B noktasının koordinatlarını aşağıda verilen formüller ile hesaplamaktayız:

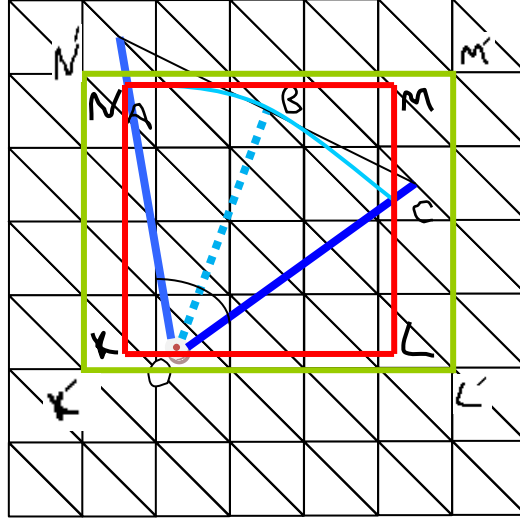
$$\begin{aligned} B.x &= O.x + (|OB| * \cos(\hat{XOB})) \\ B.z &= O.z + (|OB| * \sin(\hat{XOB})) \end{aligned} \quad (2)$$

Sensörün görüş eksen (OB doğrusu), yatay görüş açısının açıortayı olduğu için, $|OB|=|OA|=|OC|$. Bu nedenle, A ve C noktalarının arazi üzerindeki koordinatları aşağıdaki formüller yardımı ile hesaplanır.

$$\begin{aligned} A.x &= O.x + (|OB| * \cos(\hat{XOA})) & A.z &= O.z + (|OB| * \sin(\hat{XOA})) \\ C.x &= O.x + (|OB| * \cos(\hat{XOC})) & C.z &= O.z + (|OB| * \sin(\hat{XOC})) \end{aligned} \quad (3)$$

Adım 3: Çerçeveleyen Dikdörtgen Oluşturulması

O, A, B, C noktaları göz önüne alınarak minimum-maksimum x ve z değerleri bulunur. Bu dört noktanın (K,L,M,N) birleştirilmesi ile bir dikdörtgen oluşturulur (Şekil 11).



Şekil 11: Yatay FOV testi için çerçeveleyen dikdörtgenin oluşturulması

Dört nokta bulunduktan sonra, noktanın dikdörtgen üzerindeki konumuna bağlı olarak, üzerinde bulunduğu poligonun en köşe noktası seçilir. Aşağıdaki dört noktayı gösteren her bir seçenek için, nokta alt veya üst poligonlardan hangisinde olursa olsun, her durumda şekilde işaretlenen nokta seçilir. Bu işlem sonucunda bir önceki sayfada verilen KLMN dikdörtgenini içine alan ve onu çevreleyen yeni bir dikdörtgen (K'L'M'N' dikdörtgeni) çizilmiş olur. Bu adımda ayrıca, dikdörtgenin köşe noktalarının arazi alanının içinde olması da sağlanmaktadır (Şekil 12).



Şekil 12: Köşe noktalarının arazi içerisinde yer almasının sağlanması

Bu adımda, yapılacak işlemin doğruluğunu bozmadan, yatay FOV testine girecek poligonların (poligon noktalarının) sayısını azaltarak, yatay FOV testinin gerektireceği hesaplama zamanını azaltmak hedeflenmiştir.

Adım 4: Noktalara Yatay FOV testi uygulanması

Bu adımda, önceki adımda oluşturulan dikdörtgenin üzerinde bulunan her bir poligonun köşe noktaları ve ağırlık merkezinin, sensörün yatay görüş alanı içinde olup olmadığını test ederiz. Göz önüne alınan her noktanın, AOB veya BOC üçgenlerinin birinin içerisinde olup olmadığı kontrol edilmektedir. Bu iki üçgen yerine AOC üçgeni kullanılırsa, yatay FOV alanı içinde olan bazı noktalar kaybedilmiş olacaktır; daha hassas bir filtreleme yapabilmek için iki ayrı üçgen göz önüne alınmıştır.

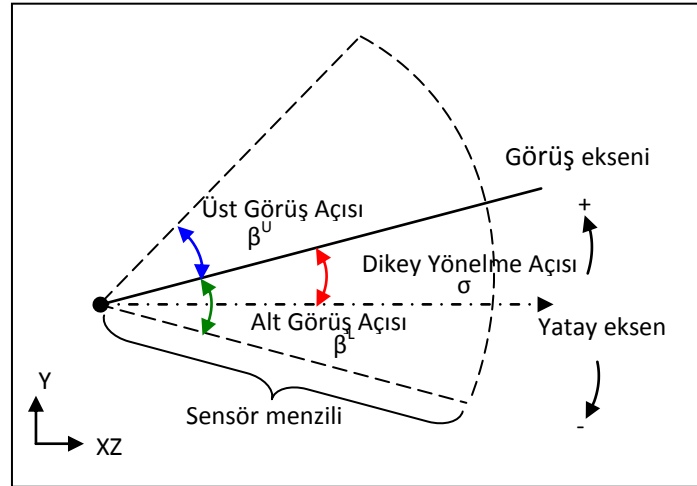
Poligonların ağırlık merkezleri, öncelikle her bir ağırlık merkezi noktasının hangi poligon üzerinde olduğu bulunup, o poligonun hafızada yer alan poligon listesinin ağırlık merkezi ile ilgili alanına erişilmesi ile bulunabilir. X ve Z eksenlerine göre yanyana yer alan herhangi iki nokta arasındaki

mesafe (iki poligonun birleşmesi ile oluşan karenin kenar uzunluğu) sabit olup, ölçek değeri olarak konfigürasyon dosyasında belirlenmiştir. Ağırlık merkez noktalarının koordinatlarını belirlemek için yaptığımız alternatif yöntemde, bir kareyi oluşturan alt poligon için poligonun alt sol köşe noktaya uzaklığı her iki eksen için ölçek/3 iken, üst poligon için bu değer 2*ölçek/3 olacaktır.

2.3.2 Dikey FOV Testi

Yatay FOV testi tamamlandığında, yatay FOV'a göre algılanan arazi noktalarının listesi dikey FOV testi için girdi olacaktır. Bu fonksiyonumuz için gereken diğer girdiler; sensörün üzerinde bulunduğu poligonun numarası, sensörün dikey üst görüş açısı (β^U) ve dikey alt görüş açısı (β^L), sensörün dikey yönelme açısı (σ), poligon haritası ve nokta haritasıdır (Şekil 13). Doküman boyunca, dikey üst görüş açısı ve dikey alt görüş açısı ibarelerinde bulunan dikey kelimesi, (yatay görüş açısı için alt ve üst limit olmadığı için) çıkarılmıştır.

Dikey FOV testi tamamlandığında, geride kalan nokta kümesi sensörün görme konisini oluşturacaktır. Bir sonraki bölümde, o kümenin elemanları için seçilecek olan görüş çizgisi algoritmaları uygulanacaktır. Aşağıda dikey FOV testi adımları özetlenmiştir:



Şekil 13: Dikey görüş alanı

Adım 1: Yatay FOV testini geçen noktalar için yükseklik değerlerinin yüklenmesi

Yatay FOV testini geçen noktaların bir kısmı poligon köşe noktaları (kesişim noktaları), bir kısmı ise poligon ağırlık merkezi noktalarıdır. Yapılan gerçeekte, poligonların kesişen noktalarının yükseklikleri nokta haritasından, ağırlık merkezlerinin yükseklik değerleri ise poligon haritasından alınmaktadır.

Adım 2: Sensör Eğimlerinin Hesaplanması

Sensörün konumsal parametrelerinden olan dikey yönelme açısı (σ), davranışsal parametrelerinden olan alt görüş (β^L) ve üst görüş açıları (β^U) kullanılarak, sensöre yönelik alt FOV eğimi (ρ^L) ve üst FOV eğimi (ρ^U) değerleri bulunur.

$$\rho^L = \tan(\sigma - \beta^L) \quad \rho^U = \tan(\sigma + \beta^U) \quad (4)$$

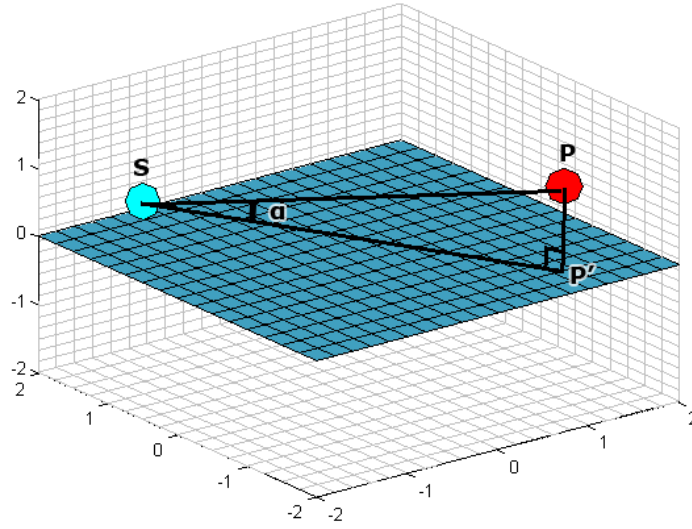
Bu hesaplamalarda, $\tan(90)$ ve $\tan(-90)$ durumlarında açığa çok küçük bir değer ($\varepsilon = 0.05$) eklenir.

Adım 3: Sensörle Nokta Arasındaki Eğimin Hesaplanması

Bu adımda, dikey FOV testi uygulanacak her nokta için sensörle o nokta arasındaki eğim

hesaplanacaktır. Örnek olarak, aşağıda verilen şekilde, sensörün S ve dikey FOV testi yapılacak poligon noktasının P olduğunu düşünelim. Bu adımda bulacağımız eğim değeri $\tan(\alpha)$ 'ya eşit olacaktır (Şekil 14).

S noktasını (x_1, y_1, z_1) olarak ve P noktasını (x_2, y_2, z_2) olarak gösterelim. P' ise P noktasının dikey izdüşümü olup, S ile aynı düzlemedir. P' noktası da (x_2, y_1, z_2) olarak gösterilir.



Şekil 14: Sensörle nokta arasındaki eğim

Bu durumda, aşağıdaki formüller kullanılarak noktanın eğimi hesaplanır:

$$|SP'| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad |PP'| = |y_2 - y_1|$$

$$\tan(\alpha) = \frac{|PP'|}{|SP'|} \quad (5)$$

Adım 4: Noktaya Dikey FOV Testi Uygulanması

Bir önceki adımda bulunan noktanın eğimi, ilk adımda bulunan sensör eğim değerleri ile karşılaştırılır. Noktanın eğimi aşağıdaki eşitsizliği sağlaması durumunda, o nokta dikey FOV testini geçmiş olarak kabul edilir.

$$\min(\rho^L, \rho^U) \leq \tan(\alpha) \leq \max(\rho^L, \rho^U) \quad (6)$$

Üçüncü ve dördüncü adımlar, yatay FOV geçen tüm noktalara uygulanır; sonuçta kalan noktalar görme konisini oluşturur.

2.4 Görüş Çizgisi (Line-of-Sight [LOS]) Algoritmaları

LOS algoritmalarının temel amacı, arazi üzerinde herhangi bir noktanın sensörden hedefe yönelen bir veya daha fazla ışını bloke edip etmediğini belirlemektir (PROCTOR, 2004). Blokajı belirlemek için sensörden hedefe yönelen üç-boyutlu ışının farklı noktalarından seçilen bir *değerlendirme kümesi* göz önüne alınır. Bu küme içindeki her (x, z) noktası için (yükseklik boyutunu göz önüne almadan XZ

düzleminde) arazinin o noktadaki yüksekliği ile ışının o noktadaki yüksekliği karşılaştırılır. Bu karşılaştırma sonucunda, arazinin o noktada daha yüksek olması durumunda, ışının bloke olduğuna karar verilir ve değerlendirme kümesi içindeki diğer noktalara bakılmaz. Aksi durumda ise, bir sonraki nokta değerlendirilir.

Her bir değerlendirme noktasında yapılan testin bir alternatifi de, sensörden o noktaya olan çizginin eğiminin sensörden hedefe olan çizginin eğimi ile karşılaştırılmasıdır. Arazinin verilen her bir (x, z) noktasındaki yükseklik değerini bulmak için literatürdeki iki farklı metot (PROCTOR, 2004) tasarlanmış olup, biz de çalışmamızda her iki metottan birer algoritma kullanmaktayız:

1. *Interpolation*: Enterpolasyon yöntemi ile yükseklik değerinin bulunması. Çalışmamızda gerçekleştirilen Janus LOS algoritması bu kategoridendir.
2. *Nearest Post*: Yakın nokta yöntemi ile yükseklik değerinin bulunması. Çalışmamızda gerçekleştirilen ikinci algoritma olan Bresenham algoritması bu kategoridendir.

2.4.1 Janus LOS Algoritması

Janus algoritması kullanılan çalışmalarda, göz önüne alınan arazi ızgara-tabanlı bir arazi ise, verilen bir (x, z) noktasındaki yükseklik değerini bulmak için 4-noktalı enterpolasyon (four-post interpolation) gerçekleştirilir (HANDERSON, 1999). Projemiz kapsamında geliştirdiğimiz araziler ise, üçgensel poligon yapısında olduğu için, poligonun belirlediği düzlemin verilen (x, z) noktasındaki yükseklik değeri bulunur. Diğer bir ifade ile bulacağımız değer, (x, z) noktasının verilen poligon düzlemine olan en kısa uzaklık değeridir. Bu bölümde, çalışmamızda gerçekleştirdiğimiz Janus LOS algoritmasının adımlarından kısaca bahsedeceğiz. Burada bahsedilen adımlar literatürde (HANDERSON, 1999) geçen algoritmanın çalışmamıza uyarlanmış halidir.

M nolu poligonun ağırlık merkezi üzerinde bir sensör olduğunu düşünelim; bu nokta P noktası olsun, $P(x_1, y_1, z_1)$. Ayrıca hedefteki nokta (LOS testi yapacağımız nokta) da $Q(x_1, y_1, z_1)$ noktası olsun. Çalışmamızın daha önceki kısımlarında anlatıldığı üzere, görme konisi testi sonucunda görme konisi içerisinde olan her nokta için LOS testi yapılacaktır.

Önce bu iki noktanın (P ve Q), X ve Z 'ye göre ofset farkları alınarak hangisi maksimumsa o değer referans kabul edilir, $\Delta = \max(|x_1 - x_2|, |z_1 - z_2|)$. Burada çıkan Δ değeri arazinin ölçek değerine (çalışmamızda bu poligonun iki kısa kenar uzunluğudur) bölünerek çıkan sayı en yakın tam sayıya yuvarlanır. Bu işlem sonunda elde ettiğimiz değere k diyelim; k sensörden hedefe kadar arada bakılacak nokta sayısı (değerlendirme kümesindeki nokta sayısı) olacaktır. Buradaki önemli nokta k 'nın çok büyük değer olması, hesaplama zamanının artıracığı gibi, k 'nın çok küçük bir değer olması da sonucunun hassasiyetini ve doğruluğunu azaltacaktır.

Bir sonraki adımda PQ doğrusu $k+1$ eşit doğru parçasına bölünecek şekilde arada gereken k noktanın XZ düzlemine göre koordinatları hesaplanır. Daha sonra aradaki her bir k noktası için öncelikle o noktanın hangi poligon üzerinde olduğu belirlenerek o poligonun belirlediği düzlemin verilen noktasındaki yükseklik değeri bulunur. Bu yükseklik değeri arazinin o noktasındaki yükseklik değeridir. Daha sonra, PQ doğrusun o noktadaki yükseklik değeri ile arazinin bulunan yükseklik değeri karşılaştırılarak, eğer arazinin yükseklik değeri daha fazla ise hedefteki noktanın (Q noktasının) görülemeyeceği ortaya çıkar ve aradaki noktalara bakılmadan iterasyondan çıkılarak hedef noktanın görme değeri olarak 0 döndürülür. Aksi durumda ise, arazi oluşturulması aşamasında seçilmiş olabilecek nesneler ve hava koşullarına yönelik testler yapılarak hedef noktanın görme değeri $[0,1]$ aralığında olacak şekilde hesaplanır.

2.4.2 Bresenham LOS Algoritması

Çalışmamızda gerçekleştirilen diğer LOS algoritması olan Bresenham Algoritması ise yakın nokta yöntemine (PROCTOR, 2004) göre çalışmakta olup, bu algoritmada Janus'daki gibi enterpolasyon aşaması yoktur. Enterpolasyon ile ilgili hesaplamalar olmaması ve daha az sayıda nokta (aradaki kontrol noktası) göz önüne alındığından dolayı, hızlı olmasına rağmen Janus kadar hassasiyete sahip olmayabilir. Bu algoritmada, herhangi bir (x, z) noktasının yüksekliği, (yükseklik değeri set edilmiş) ona en yakın olan noktanın yükseklik değeri ile aynı kabul edilir. Literatürde kullanılan dört-noktalı arazi gösteriminde, sensör ile hedef noktası arasındaki köşe noktalar göz önüne alınırken, çalışmamızda üçgensel poligonlar kullanılması nedeni ile sensörden hedef noktasına olan çizgiye en yakın olan ağırlık merkez noktalarını kontrol noktaları olarak almaktayız.

3 Gereç ve Yöntem

Sensörlerin konumlandırılması ve oryantasyonunun ayarlanması sırasında çok fazla sayıda parametrenin kullanılması, temsili alanın üzerindeki doğal, çevresel ve diğer faktörlerin birçok parametre ile belirtilmesi neden ile çok büyük arama uzayına sahip bir problemdir. Çok büyük arama uzayına sahip problemlerin çözümünde klasik metotlara göre daha etkin ve iyi sonuç veren arama ve eniyileme (optimizasyon) metotlarının başında evrimsel algoritmalar gelmektedir. Projemizde problemin modellenmesinde ve çözümünde, karar analizi tekniklerinden çok ölçütlü fayda teorisini temel alan evrimsel algoritmalarla yerel aramayı (local search) birarada kullanan karma çözüm stratejileri yöntem olarak kullanılmıştır.

3.1 Çok - Ölçütlü Fayda Teorisi

Çok ölçütlü eniyileme yöntemleri Skalarizasyon ve Pareto yaklaşımları olmak üzere iki grupta incelenir (ANDERSSON, 2001). Birinci kategoride tüm eniyileme fonksiyonları tek bir fonksiyonda yazılır. Pareto yönteminde ise çok ölçütlü fonksiyonun elemanları eniyileme süreci boyunca birbirinden ayrı değerlendirilir. Von Neumann ve Morgenstein (VON NEUMANN, 1964) tarafından ortaya atılan *fayda teorisine* göre; karar verici, her karar seçeneği için bir fayda fonksiyonu tanımlar ve beklenen değer kriterine benzer bir yöntemle seçenekten beklediği faydayı hesaplar.

$$U(a_i) = \sum_j P(O_j) U(V_{ij}) \quad (7)$$

Yukarıdaki denklemde $U(a_i)$ i 'inci seçeneğin fayda değerini, $P(O_j)$ j 'inci olayın meydana gelme olasılığını, $U(V_{ij})$ ise (i, j) çıktısının fayda değerini verir.

Bu ölçütün beklenen değer kriterinden farkı, formülden de anlaşılacağı gibi, çıktının kendisi yerine çıktının faydasının dikkate alınmasıdır. Sonuçta da beklenen faydası en yüksek olan seçenek tercih edilecektir. Fayda teoreminin geçerliliğinin sağlanabilmesi için bazı temel noktaların gerçekleşmesi gerekecektir ([CLEMEN, 1991), (KEENEY, 1993)]. Beklenen fayda kriterine göre bir seçim yapabilmek için, önce tüm çıktıların bir fayda fonksiyonuna dönüştürülmesi, daha sonra da seçeneklerin beklenen faydalarının saptanması gereklidir. Aslında sonuçta yapılan, karar vericinin fayda eğrisinden hareketle saptanan tercih yapısının belirlenip, o yapı ile tutarlı bir şekilde en yüksek belirlilik eşdeğerini veren seçeneğin seçilmesidir.

Fayda kuramı, özellikle askeri uygulamalarda, belirsizlik veya kısmi bilgi ortamında farklı senaryoların (risk alan, riskten kaçınan, vs.) denenmesini mümkün kıldığından tercih edilmektedir. Özellikle tek bir amaç yerine, bizim projemizde de olduğu gibi, birden fazla amacın var olması durumunda fayda analizi son derece karmaşıklaşmakta ve bazı ayrıştırma şartlarını (toplanabilirlik, bağımsızlık, vs.) sağlaması halinde uygulanabilmektedir [(KEENEY, 1993), (DYER, 1990)].

Birbirleriyle çelişen ancak aynı zamanda birbirlerini tamamlayan birden fazla ölçütün bulunması durumunda, eğer ölçütler birbirinden bağımsız ise çarpımsal ya da çoklu-doğrusal fayda fonksiyonu kullanılabilir. Bizim temel varsayımımız; bir askeri harekatta sensörlerin algılama davranışının birbirleriyle çelişen üç ölçüt (öncelik) arasında rasyonel bir ödünleşmeye (trade-off) dayanması gerektiğidir. Bu üç ölçüt: i) arazi hakkında elde edilen bilgilerin maksimum yapılması, ii) düşman tarafından toplanan bilginin (tespit edilme) minimum yapılması ve iii) konumlandırılan sensörlerin toplam maliyetlerinin minimum yapılmasıdır. Çok Ölçütlü Fayda (ÇÖF) analizinde ilk adım, performans ölçütlerinin ve her bir alternatife göre tahmini değerlerinin belirlenmesidir. Daha sonra, her bir performans ölçütü için, 0 ile 1 arasında değer alacak tek-ölçütlü fayda fonksiyonu değeri bulunur. Bir ÇÖF fonksiyonu her bir ölçümün toplam performans üzerinde nasıl bir etkisinin olacağını görece önemlerine veya belirlenmiş ağırlıklarına göre hesaplar.

Karar vericinin tercihlerinin doğru koşulları sağlayıp sağlamadığını belirlemek için, ölçütlerin birbirinden bağımsız olduğu varsayımıyla global fayda fonksiyonu olarak çarpımsal fayda fonksiyonu kullandık. Askeri bir keşif görevi için, maksimum yapmaya çalıştığımız global fayda fonksiyonu, $U(A,S,P)$, P pozisyonunda bulunan bir S sensörü ile taranan arazi A ise, aşağıdaki gibi formüle edilmiştir:

$$\begin{aligned} U(A, S, P) = & w_{vis} U_{vis}(A, S, P) + w_{st} U_{st}(A, S, P) + w_{cost} U_{cost}(S, P) \\ & + w_{vis} w_{st} U_{vis}(A, S, P) U_{st}(A, S, P) \\ & + w_{vis} w_{cost} U_{vis}(A, S, P) U_{cost}(S, P) \\ & + w_{st} w_{cost} U_{st}(A, S, P) U_{cost}(S, P) \\ & + w_{vis} w_{st} w_{cost} U_{vis}(A, S, P) U_{st}(A, S, P) U_{cost}(S, P) \end{aligned} \quad (8)$$

Bu denklemde; $U_{vis}(A, S, P)$, P noktalarına konumlandırılmış bir S sensör kümesi ile taranan arazi A 'nın görünürlük faydası, $U_{st}(A, S, P)$, S sensör kümesinin gizlenme faydası ve $U_{cost}(S, P)$ ise sensörlerin maliyetlerinin faydasıdır. w_{vis} , w_{st} , w_{cost} değerleri ise ağırlık değerleri olup $[0, 1]$ aralığında değer alır ve bu değerler deneysel olarak veya askeri harekâtın önceliklerine (sızma, intikal, keşif, taktik akın, sıyırılma, pusu, pusuya karşı koyma, vs.) göre karar verici tarafından belirlenir.

$$w_{vis} + w_{st} + w_{cost} = 1 \quad (9)$$

3.1.1 Görünürlük - Faydasının Hesaplanması

Tasarladığımız modelde yer alan toplam alan görünürlük bileşeni, alanı oluşturan poligonların her birinin görünürlüğünün toplamına eşittir. Aşağıda verilen denklemlerde toplam alanın görünürlüğü (10) ve bir poligonun görünürlüğünün (11) hesaplanması belirtilmiştir.

$$U_{vis}(A, S, P) = \frac{\sum_{p_i \in A} V(S, P, p_i) \times W_{p_i}}{\sum_{p_i \in A} W_{p_i}} \quad (10)$$

$$V(S, P, p_i) = \frac{\sum_{b \in p_i} \max_{s_j \in S} \{V_{s_j}(s_j, p_k, b)\}}{n} \quad (11)$$

Yukarıdaki denklemde W_{p_i} , p_i poligonun ağırlığını ve $V(S, P, p_i)$ ise P poligon setindeki noktalara konumlandırılmış S sensör seti kullanılarak p_i poligonunun görünürlük değerini gösterir. Poligonun ağırlığı W_{p_i} iki farklı deneysel yöntem ile belirlenebilir:

- Temsili arazinin oluşturulması sırasında, arazinin bir veya daha fazla bölgesi (ya da tamamı) seçilerek bu bölgedeki bütün poligonlara aynı ağırlık değeri verilebilir. Bu ağırlık değerleri $\{0, 0.25, 0.50, 0.75, 1.00\}$ değerlerinden birisine eşit olacaktır. Eğer bir poligon seçilmemişse ağırlık değeri 0.50 olarak atanır.
- Ağırlıklar temsili arazinin özelliklerine göre belirlenir. Poligonların yükseklik ve eğimlerine göre ağırlık değerleri $(0, \dots, 1)$ arasında olacak şekilde bir fonksiyon tanımlanmıştır. Temsili arazideki diğer poligonlara ve belirlenen bir eşik değere göre daha büyük bir yükseklik değerine sahip olan herhangi bir poligona ağırlık değeri olarak 1'e çok yakın bir değer atanır.

Belirli bir sensörün (s_j) görünürlük faydası da benzer şekilde aşağıda verilen denkleme göre hesaplanmıştır (12). $V_{s_j}(s_j, p_k, p_i)$, p_k poligonuna konumlandırılmış s_j sensörü kullanılarak bulunan p_i poligonunun görünürlük değeridir.

$$U_{vis}^j(A, s_j, p_k) = \frac{\sum_{p_i \in A} V_{s_j}(s_j, p_k, p_i) \times W_{p_i}}{\sum_{p_i \in A} W_{p_i}} \quad (12)$$

$$V(s_j, p_k, p_i) = \frac{\sum_{b \in p_i} V_{s_j}(s_j, p_k, b)}{n} \quad (13)$$

Görünürlük faydası, sensör özellikli görünürlük faydalarının toplamına eşit değildir. $V_{s_j}(s_j, p_k, b)$ terimi, p_i poligonuna ait bir nokta olan b noktasının, p_i poligonunun kütle merkezine konuşlandırılmış olan s_j sensörü tarafından görünürlük değeridir. b noktası birden fazla sensör tarafından farklı görünürlük değeri ile teşhis edilebileceğinden, Denklem 13'de b noktasının maksimum görünürlük değeri kullanılır. Bu denklemde, p_i poligonunun görünürlüğü seçilen poligon noktalarının ortalama görünürlüğüne eşit kabul edilmiştir. Projemizin bu aşamasında, üç köşe noktası ile kütle merkezi olmak üzere bir poligon için dört noktanın görünürlüğü'nün ortalaması alınmıştır.

Yukarıdaki terimlere ek olarak, *görünürlük etkinliği* diye adlandırdığımız yeni bir terim kullanılmaktadır. Bu terimin hesaplanmasında, sensör görünürlük faydası için kullanılan denklemin payda kısmı, p_k poligonuna yerleştirilmiş en iyi sensörün yatay FOV testinden geçen poligonların ağırlık değerleri toplamı ile değiştirilir. En iyi sensörün yatay FOV testinden geçen poligonlar, p_k poligonuna yerleştirilmiş bir sensörün olabilecek en yüksek görünürlük fayda değerini verecek sensör konfigürasyonudur. Böylece bir sensörün *görünürlük etkinliği*, sensörün normalize edilmiş görünürlük değerini verir.

$V_{s_j}(s_j, p_k, b)$ terimi, a kaynak noktasına (p_k poligonunun kütle merkezi) konuşlandırılmış s_j sensöründen, hedef nokta olan b 'nin görünürlüğü aşağıdaki gibi hesaplanır.

$$V_{s_j}(s_j, p_k, b) = (1 - \eta_{s_j} \times \frac{D(a, b)}{\Delta_{s_j}}) \times (1 - \max_{p_c \in a \rightarrow b} \psi_c^W) \times (1 - \max_{p_c \in a \rightarrow b} \psi_c^O) \quad (14)$$

Bu formülasyonda, $D(a, b)$ a ve b noktaları arasındaki mesafe, Δ_{s_j} görüş mesafesi ve ρ_{s_j} ise s_j sensörünün menzile bağlı etki katsayısıdır. İkinci terim havanın geçirgenlik değeridir ve atmosferik koşullara bağlı olarak değişen yoğunluk değerlerine eşittir. ψ_c^W terimi, a ve b noktaları arasındaki bir poligonun (p_c) üzerindeki havanın yoğunluğudur. Havanın karlı, yağmurlu veya sisli olmasına bağlı olarak bu terim bir yoğunluk değeri alır. Yüksek yoğunluk değeri düşük görüş değerine neden olur. a ve b noktaları arasındaki bütün poligonların atmosferik koşullarına bağlı olarak aldıkları yoğunluk değerleri LOS algoritmasında dikkate alınır ve eğer a ve b noktaları arasındaki herhangi bir ara nokta engellenmemişse maksimum değer kullanılır.

Denklemdaki son terim ise nesnenin geçirgenlik değeridir ve nesnelerin yoğunluğuna göre belirlenir. Temsili arazi üzerinde doğal (orman, çalılık, vs.) ve yapay (binalar, vs.) nesneler olabileceğinden, poligon üzerine bir nesne konuşlandırıldığında, bu nesnenin yükseklik ve yoğunluk değeri poligona eklenir.

3.1.2 Gizlenme - Faydasının Hesaplanması

Düşman tarafından tespit edilemeyen sensörlerin askeri uygulamalarda çok büyük yararı bulunmaktadır. Gizlilik, sensörlerin varlığının düşman tarafından gözlenmesini engelleyecek şekilde arazi görüntüleme işlemini gerçekleştirilmesini sağlayarak, askeri hareketlerin engellenme riskini azaltır.

Belirli bir arazi üzerine konumlandırılan veya belirli bir güzergah üzerinde keşif görevi yapan sensörlerin toplam gizlenme faydası, düşman sensörleri kullanılarak konuşlandırılmış sensörlerin toplam görünürlük maliyetinin 1'den çıkartılmasıyla bulunur. Bu amaçla, belirli bir m senaryosuna göre önceden belirlenmiş sayıda düşman objeleri arazi üzerine rasgele dağıtılır. Bu objeler düşman sensörlerini taşıyan araçlardır. Deneysel çalışmamızda, düşman sensörlerinin açığa bağlı konumsal nitelikleri (görme açısı, görme menzili, vs.) en kötü sensör değerlerine göre belirlenmiştir.

Tek bir senaryoya göre P poligon setine konuşlandırılmış S sensörün gizlenme faydası aşağıdaki gibi hesaplanır:

$$U_{s,m}(A, S, P) = 1 - \sum_{s_i \in S} V_E(E, P, p_{s_i}) \times R_U(A, s_i, p_{s_i}) \quad (15)$$

$V_E(E, P, p_{s_i})$ terimi, p_{s_i} poligonuna konumlandırılmış s_i sensörünün, P pozisyonundaki E düşman araçından maksimum görünürlüğüdür. Sensör görünürlüğünde olduğu gibi:

$$V_E(E, P, p_{s_i}) = \max_{e_j \in E} \{V_E(e_j, p_k, p_{s_i})\} \quad (16)$$

hesaplanır. $V_E(e_j, p_k, p_{s_i})$ terimi, s_i sensörünün konuşlandırıldığı arazi üzerindeki tek bir noktanın, p_k poligonuna konuşlu bir düşman sensörü tarafından görünürlüğüdür. Bu terim, $V_{s_j}(s_j, p_k, b)$ 'nin dualidir ve bu nedenle de Denklem 14'e göre hesaplanır. Diğer yandan, verilen senaryo için tek bir s_j sensörünün gizlenme faydası Denklem 15'deki toplam ifadesi kaldırılarak hesaplanır.

$R_U(A, s_i, p_{s_i})$ terimi s_i sensörünün görünürlük fayda oranı olup, bu oran verilen sensörün görünürlük faydasının bütün sensörlerin görünürlük faydası toplamına oranına eşittir.

$$R_U(A, s_i, p_{s_i}) = \frac{U_{vis}(A, s_i, p_{s_i})}{\sum_{s_j \in S} U_{vis}(A, s_j, p_{s_j})} \quad (17)$$

Yukarıdaki ifadenin paydası genel görünürlük faydası değil, görünürlük faydalarının toplamıdır. Bu durum, genel görünürlük faydasının tek bir sensörün görünürlük faydasına bile eşit olabileceği gerçeğinden kaynaklanır. Eğer yüksek görünürlük fayda değerine (dolayısıyla yüksek fayda oranı) sahip bir sensör düşman sensörü tarafından tamamen görünürse, bu durum gizlenme faydasını önemli ölçüde azaltacaktır.

Sensörlerin dağıtılması ve hesaplamaların yapılması daha önceden belirlenmiş sayıda tekrarlanır ve bu senaryolardan elde edilen gizlenme faydalarının ortalaması toplam gizlenme faydası olarak alınır. Eğer r düşman nesnelerinin konumlandırılması için belirlenen farklı senaryo sayısı ise, toplam gizlenme faydası:

$$U_{s,m}(A, S, P) = \frac{\sum_{x=1}^r U_{s,m}(A, S, P)}{r} \quad (18)$$

Program çalıştırılmadan önce $r * |E|$ düşman sensörünün konumsal parametreleri (poligon numarası, yatay ve dikey yönelme açıları, sensör tipi) belirlenir. Düşman sensörlerinin parametreleri programın çalışması esnasında değiştirilmemiştir.

3.1.3 Maliyet - Faydasının Hesaplanması

Tasarladığımız modelde yer alan maliyet faydası bileşeninin hesaplanmasında, sensörlerin değeri ile ilgili iki farklı anlam göz önüne alınmıştır: (a) sensörün normalize edilmiş finansal maliyeti ($NCost_F(s_i)$); (b) sensörün verilen bir noktaya yerleştirme maliyetinin normalize edilmiş değeri ($NCost_L(s_i, p_k)$). Maliyet-faydasının hesaplanması aşağıda verilen denklem yardımı ile yapıldı:

$$U_{cost}(S, P) = 1 - \left\{ \frac{\sum_{s_i \in S} (w_F \times NCost_F(s_i) + w_L \times NCost_L(s_i, p_k))}{|S|} \right\} \times N|S| \quad (19)$$

Finansal maliyet ile konumlandırma maliyetleri birbirlerinden bağımsızdır ve w_F ile w_L bu iki maliyet teriminin 0 ile 1 arasında değer alan ağırlıklarıdır. Konuşlandırma maliyeti $NCost_L(s_i, p_k) = P_L(s_i, p_k) \times \varphi_{s_i}$ denklemi ile hesaplanır. Bu denklemde $P_L(s_i, p_k)$, p_k poligonundaki s_i sensörünün konuşlandırılma olasılığı, φ_{s_i} ise sensör tipine bağlı olarak belirlenen bir sabittir. Konuşlandırma olasılık değeri, temsili arazideki bütün poligonların yükseklik ve eğimleri ile sensörlerin karakteristiklerine bağlı olarak belirlenir. Denklem 19'daki $N|S|$ sensör kullanımının normalize edilmiş değeridir ve $N|S| = \frac{S}{E(S)}$ formülü ile hesaplanır.

3.2 Karma Evrimsel Algoritma Tabanlı Çözüm Yöntemi

Evrimsel Algoritma, aslında Genetik Algoritma ve diğer varyasyonlar için bir çatı (kategori) ismi olması nedeni ile raporumuzun bundan sonraki bölümlerinde Genetik Algoritma terimi kullanılacaktır. Basitleştirilmiş genetik algoritmalar birçok eniyileme problemi için yaygın bir şekilde kullanılmasına rağmen özellikle karmaşık ve birleşik eniyileme problemleri için verimli çözüm üretmez (MICHALEWICZ, 2000). Genetik algoritmaların performansını yükseltmek için, bazı probleme özgü tekniklerin [(EIBEN, 2003), (FOX, 1990), (GREFFENSTETTE, 1987), (RAIDL, 2002)] genetik algoritmalarla birleştirilmesi gereklidir. Genetik algoritmaların diğer tekniklerle beraber karma olarak kullanılması durumunda bu yöntem, Memetik Algoritma (Memetic Algorithm) veya Karma Evrimsel Algoritma olarak adlandırılmaktadır [(MOSCATO, 1989), (RADCLIFFE, 1994)].

Bu proje kapsamında geliştirdiğimiz genetik algoritmamız da, aşağıda belirtilen üç farklı yolla diğer operatör tipleri ve probleme özgü bilgi içermesi nedeni ile Karma Evrimsel Algoritma olarak ele alınabilir (EIBEN, 2003).

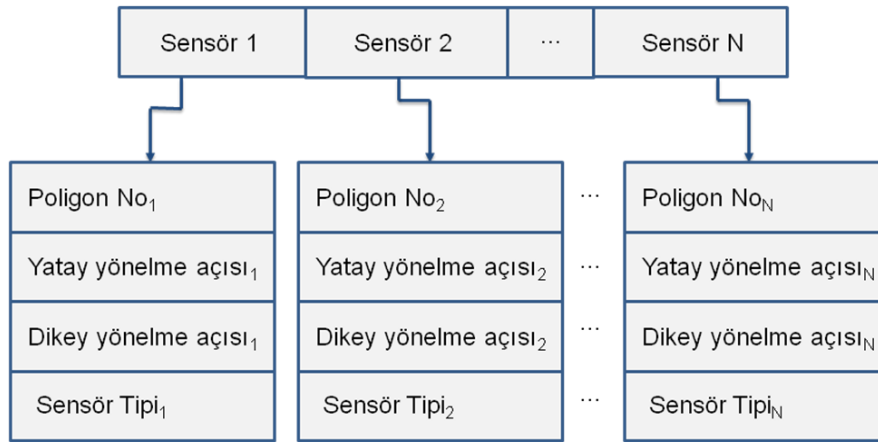
Sezgisel ve akıllı metotlarla başlangıç popülasyonun oluşturulması: Aşağıda detaylı anlatıldığı üzere ilk popülasyonun üretilmesi işleminde sezgisel yaklaşımlar kullanılmıştır.

Varyasyon operatörleri içerisinde hibridizasyonun gerçekleştirilmesi: Çalışmamızda çaprazlama ve mutasyon operatörlerine yönelik farklı hibridizasyonlar (karma teknikler) ve probleme özgü adımlar ile problem bilgisi eklenmiştir.

Varyasyon operatörlerinin sonucuna yerel arama uygulanması: Çalışmamızda çaprazlama operatöründen sonra yerel arama işlemi gerçekleştirilmiştir.

3.2.1 Kromozom Yapısı

Çalışmamızda, her bir çözümü içeren kromozom yapısı sensörlerden oluşmakta olup, her bir sensör, sensör tipi ile poligon no, yatay yönelme açısı, dikey yönelme açısı gibi sensörün konumsal özelliklerini içerir (Şekil 15). Sensör sayısı üzerine bir kısıt getirilmeyen problemlerde, farklı çözümlerde (farklı kromozomlarda) farklı sayıda sensörler olabilmektedir. Yapılan çalışmada, sensörlerin kromozom üzerinde yerleştirilmesine yönelik hiç bir sınırlama olmayıp, sensörlerin (kromozom genlerinin) sırası önemli değildir.



Şekil 15: Kromozom yapısı

Çalışmamızda steady-state (sabit-popülasyonlu) Genetik Algoritma (GA) yaklaşımı kullanılmış olup, her nesilde tek bir birey üretilir ve çaprazlama ve mutasyon bir orana bağlı olmadan uygulanır. Sonuçta, iki ebeveyn bireyden üretilen çözüm popülasyondaki en kötü bireyin yerine yazılır. Bireylerin uygunluk fonksiyonu değerleri için çok ölçütlü fayda teorisi kısmında açıklanan toplam fayda değeri kullanılmıştır.

3.2.2 Başlangıç Popülasyonunun Oluşturulması

Başlangıç popülasyonu için öncelikle birey üretme aşamasında her birey için gerekli sensör sayısı belirlenir. Bir bireydeki sensör sayısı, aşağıdaki denklemle belirlenen sensör alt ve üst limitleri arasından rasgele belirlenir. Senaryo türü (tespit, teşhis, tanıma) ve verilen senaryo türüne göre sensörün görüş mesafesi (Δ) kullanılarak sensör alt ve üst limitleri belirlenir. Denklemde yer alan $\Delta_{\max}$ tüm sensör tiplerinin görüş mesafeleri arasında en yüksek görüş mesafesini gösterirken $\Delta_{\min}$ en düşük görüş mesafesini gösterir.

$$\left(\frac{\text{Terrain_Dim}}{\Delta_{\max}}\right)^2 * \gamma_1 \leq |S| \leq \left(\frac{\text{Terrain_Dim}}{\Delta_{\min}}\right)^2 * \gamma_2 \quad (20)$$

Bağıntıda “Terrain_Dim” ifadesi, oluşturulan arazinin XZ düzleminde bir boyutunun uzunluğudur. Çalışmamızda kullanılan tüm alan üretme algoritmalarında kare şeklinde araziler oluşturulduğundan X ve Z ekseninin boyutları aynıdır. Bağıntıda geçen γ_1 ve γ_2 değerleri sabit olup arazinin karmaşıklığına göre, arazi üretme algoritmalarının parametrelerinden faydalanılarak, $0.5 \leq \gamma_1 \leq 1.25$ ve $1.0 \leq \gamma_2 \leq 1.75$ aralıklarında atanır.

Bir bireydeki sensör sayısı belirlendikten sonra, konumlandırılacak olan her bir sensörün hangi tipte olacağı, sensörlerin kapasiteleri ile ters orantılı olarak rasgele belirlenir. Kapasite değeri ise sensörün üç farklı katmana göre görüş mesafesi veya görüş açıları kullanılarak hesaplanır. Sonuç olarak konumlandırılacak sensörlerin yüksek olasılıkla düşük maliyetli (ve düşük kapasiteli) sensör tipinde olması ve düşük olasılıkla yüksek maliyetli (yüksek kapasiteli) olması hedeflenir. Sensörlerin konumları ise rasgele olarak veya aşağıda belirlenen sezgisel teknik yardımı ile belirlenir.

3.2.3 Sezgisel Olarak Sensör Konumlarının Belirlenmesi

Burada düşünülen yaklaşım sensörlerin homojen olarak arazi üzerine dağıtılmasıdır. Toplam k sensör olduğu düşünülürse öncelikle arazi, satır ve sütunda $\lfloor \sqrt{k} \rfloor$ sayıda olacak şekilde alanlara bölünerek her bir $\lfloor \sqrt{k} \rfloor * \lfloor \sqrt{k} \rfloor$ alana bir sensör (daha önce belirlenen tipte) atanır; geride kalan $k - \lfloor \sqrt{k} \rfloor * \lfloor \sqrt{k} \rfloor$ sensör ise $\lfloor \sqrt{k} \rfloor * \lfloor \sqrt{k} \rfloor$ alana rasgele olarak yerleştirilir.

Burada her bir alanın XZ ekseninde içerdiği poligonların eşit olarak dağılması sağlanır. Her bir sensör atanacağı alan belli olduktan sonra, o alandaki hangi poligona atanacağı şu şekilde belirlenir: [0,1] aralığında rasgele bir sayı üretilerek bu sayının 0.20'den küçük olması durumunda sensör, alanın merkezindeki poligona, 0.20'den büyük olması durumunda ise (%80 oranında) alandan rasgele seçilecek bir poligon üzerine yerleştirilir. Bu şekilde merkeze yerleştirme yüzdesi oldukça düşük olur ve bazı sensörler ilgili alanın merkezinde olurken, bir kısmı da farklı noktalara atanabilecektir.

3.2.4 Seçim Yöntemi

Geliştirilen algoritmada *turnuva seçim tekniği* (tournament selection) kullanılmaktadır. Turnuva seçim tekniğinde kullanılan tek parametre turnuva büyüklüğüdür. Örneğin, turnuva büyüklüğü 2 olan bir deneyde popülasyondan rasgele iki birey seçilerek bunlardan en iyi uygunluk değerine sahip olan ilk çözüm ebeveyn olarak seçilir. Bu işlem tekrarlanarak çaprazlama için gereken ikinci birey belirlenmiş olur. Deneysel çalışma sırasında problem, farklı turnuva büyüklüğü değerleri ile test edilmiştir.

3.2.5 Varyasyon Operatörleri

Popülasyondan *turnuva seçim metodu* ile seçilen iki ebeveyn çözüme öncelikle çaprazlama operatörü uygulanarak tek bir offspring (yavru çözüm) oluşturulur. Bu noktada uygulanan çaprazlama işlemi, sensörün bulunduğu poligon numarasını sezgisel metotlar da kullanarak değiştirmeye yöneliktir. Çaprazlamadan sonraki ikinci adımda ise, konumu belirlenen offspringin diğer konumsal özellikleri (yatay yönelme açısı, dikey yönelme açısı) yerel arama teknikleri ile değiştirilerek, oluşacak çözümün performansı artırılmaya çalışılır. Yerel iyileştirme aşaması tamamlandığında ise mutasyon operatörü ile çözümdeki sensörlerin konumsal özellikleri veya çözümde yer alan davranışsal özelliği (sensör tipi) değiştirilerek popülasyon içindeki çeşitliliğin artırılması hedeflenmiştir.

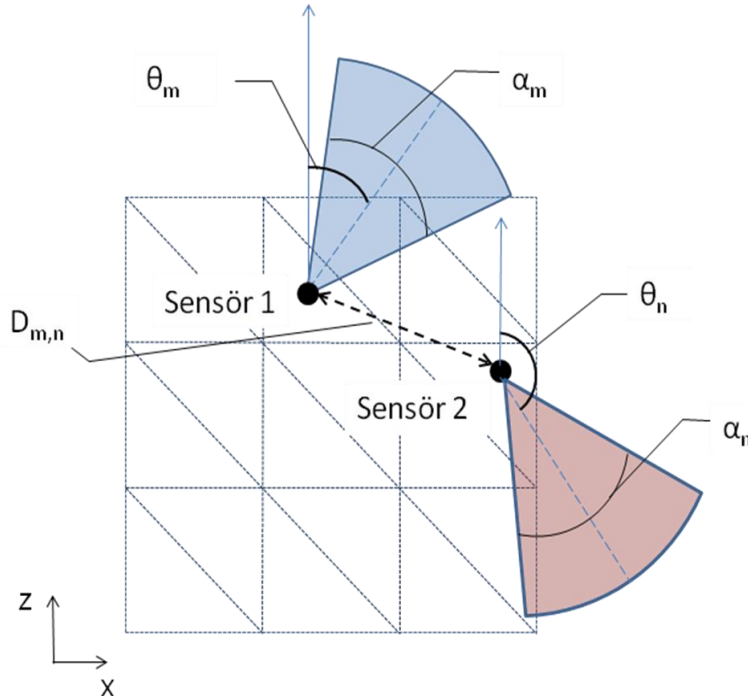
3.2.6 Çaprazlama İşlemi

Çalışmamızda sunduğumuz Karma Genetik Algoritma tabanlı çözümde, üç farklı tipte çaprazlama operatörü kullanılmıştır: i) Katkı Tabanlı Çaprazlama, ii) Komşuluk Tabanlı Çaprazlama ve iii) Kes-Birleştir Çaprazlama. Buradaki her üç metotta da sensörlerin bir çözüm içinde yerleştirilmelerine yönelik herhangi bir sınırlandırma yoktur. Sensörler gen dizisi üzerinde herhangi bir sırada olabilirler. Deneysel çalışmamızda seçilen çaprazlama operatörünün çalışma boyunca sabit olmaması ile her bir çözüm için bu üç operatörden birinin rasgele seçilmesi durumu göz önüne alınmıştır. Çaprazlama sürecinde sensörlerin konumları değiştirilirken, diğer konumsal özelliklerini yansıtan yatay ve dikey yönelme açılarında bir değişiklik yapılmaz. Ebeveynlerden seçilen gen, konum, yatay ve dikey aç bilgilerini de içerecek şekilde tüm alanları ile offspringe taşınır.

i. Katkı Tabanlı Çaprazlama

Bu operatör aşağıdaki adımlardan oluşmaktadır:

1. Her iki ebeveynden rasgele birer sensör seçilir ve seçilen sensörler incelenerek toplam uygunluk değeri (görünürlük, gizlilik ve maliyeti içerecek şekilde, toplam katkı değeri) daha yüksek olan sensör (ilgili gen) offspringe ilk sensör olarak kopyalanır.
2. Offspringe taşınmamış tüm sensörler arasından her iki ebeveynden rasgele birer sensör seçilir ve toplam uygunluk değeri daha yüksek olan ve yeni oluşturulan çözüme hâlihazırda yerleştirilmiş olan sensörlerle arasındaki mesafe, kabul edilebilir komşuluk mesafesinin (proximity distance) altında olmayan sensör offspring'e kopyalanır. Kabul edilebilir komşuluk mesafesinin nasıl hesaplanacağı aşağıda verilmiştir:
 - İki ebeveyn çözüm içerisinde değerlendirilecek katkı değeri en yüksek olan sensör, m nolu poligonda θ_m yatay yönelme açısıyla yerleştirilmiş olsun. Yavru çözümde (offspringde) ise daha önce yerleştirilmiş olan bir sensör n nolu poligon üzerinde θ_n yatay yönelme açısıyla bulunsun. Bu sensörlerin yatay görüş açıları da α_m ve α_n olsun (Şekil 16).



Şekil 16: Arazi üzerinde sensörlerin durumu

m nolu poligon üzerinde olan sensörün seçilip seçilmeyeceğine karar verilirken öncelikle aşağıdaki birinci eşitsizlik göz önüne alınır. Bu eşitsizlikte sağ taraf kabul edilebilir komşuluk mesafesi (*proximity distance*) olarak tanımlanabilir; sol taraf ise m ve n nolu poligonların ağırlık merkezleri arasındaki 3 boyutlu mesafedir. Kabul edilebilir komşuluk mesafesinin hesaplanmasında m ve n nolu poligonlara konumlandırılmış sensörlerin görüş mesafesi (Δ_m , Δ_n) ve probleme özgü μ sabiti kullanılır. (Deneylerde μ değeri 0.4 alınmaktadır.)

$$D_{n,m} > (\Delta_n + \Delta_m) \times \mu \quad (21)$$

$$|\theta^m - \theta^n| > \frac{\alpha^m + \alpha^n}{2} \times \kappa \quad (22)$$

Birinci eşitsizlikte, aradaki mesafe kabul edilebilir komşuluk mesafesinden büyük ise m nolu poligon üzerindeki sensör offspringe taşınır.

Birinci eşitsizliği sağlamayan durumlar için ikinci eşitsizlik kontrol edilir, ikinci eşitsizliği sağlanması durumunda da sensör offspringe eklenir. İkinci eşitsizlikte iki sensörün yatay yönelme açıları arasındaki fark, yatay görüş açılarının ortalamasının κ oranında büyükse, bu durumda sensörler arasındaki mesafe kabul edilebilir komşuluk mesafesinden küçük olsa bile, farklı yönlere baktıklarından sensörün offspringe eklenmesi uygundur.

3. Bir önceki adımda seçilen sensör offspringe kopyalanır. Offspringe yerleştirilen sensör sayısı ($|S|$) aşağıdaki eşitsizlikte verilen üst limitten küçük ise ikinci adıma geri dönülür.

$$|S| \leq (|S_1^P| + |S_2^P|) * \varepsilon \quad (23)$$

Burada $|S_1^P|$ ve $|S_2^P|$ birinci ve ikinci ebeveyndeki sensör sayısını (gen sayısını), ε ise 1'den küçük bir sabit değeri göstermektedir (Deneylerde $\varepsilon = 0.5$ olarak ele alınmıştır). Bu aşamada, offspringdeki sensör sayısı için bir üst limit konulmasına rağmen bir alt limit belirlenmemiştir. Komşuluk mesafesi ve yatay görüş açısı kısıtları nedeniyle offspringdeki sensör sayısı iki ebeveyndekinden de az olabilecektir.

ii. Komşuluk Tabanlı Çaprazlama

Komşuluk Tabanlı Çaprazlamanın Katkı Tabanlı Çaprazlamadan farkı, başlangıçta ilk ebeveyn gen dizisi üzerinde rasgele bir noktanın seçilmesi ve sürecin bu noktadan itibaren devam ettirilmesidir. Hangi ebeveynde minimum sayıda sensör var ise ondaki sensör sayısı (m olsun) referans alınarak, $(1, m)$ arasında rasgele seçilen sayı, çaprazlama noktasını vermiş olur. Birinci ebeveynin ilk m sensörü offspringe taşınır. Daha sonra, diğer ebeveynin ilk geninden başlayarak, önceki bölümde verilen iki eşitsizliğe göre, komşuluk mesafesi veya yatay açı kısıtlarının sağladığı ve offspringdeki sensör sayısı bir önceki metotta verilen üst sınırdan büyük olmadığı sürece, sensörler offspringe aktarılır. Bu yöntemde genlerin aktarılma sırası, genlerin toplam uygunluk değerinden bağımsız olarak sensörlerin ebeveyndeki diziliş sırasındır. Bu çaprazlama yönteminde de offspringe aktarılacak sensör sayısına yönelik bir alt limit yoktur.

iii. Kes-Birleştir Çaprazlama

Literatürde daha önce yapılan çalışmalara benzer olarak, seçilen iki ebeveyn çözüm üzerinde bir çaprazlama noktası seçilerek, birinci ebeveynden o noktaya kadar olan sensörler offspringe taşınır. İkinci ebeveynden ise belirlenen noktadan gen dizisinin sonuna kadar olan tüm sensörler offspringe kopyalanır. Bu metot literatürde diğer problemlerde de kullanılan bir metot olup, burada komşuluk mesafesi veya açı testleri yapılmamaktadır.

3.2.7 Yerel İyileştirme Prosedürü

Çaprazlama sonunda oluşturulan offspringin açısız özellikleri ile ilgili aşağıdaki adımları içeren yerel iyileştirme çalışması yapılır (Şekil 17):

```

1. Offspringteki sensörler  $S$  kümesinde tutulur.
2.  $I_R \leftarrow 0$  // sensörlerin görünürlük değerlerindeki toplam iyileştirme oranı.
3.  $q \leftarrow 0$  // çözümde incelenen toplam sensör sayısı.
4.  $\delta_H \leftarrow$  yatay yönelme açısı artış miktarı.
5.  $\delta_T \leftarrow$  dikey yönelme açısı artış miktarı.
6. En düşük  $\min s_i \in S U_v(A, s_i, p_k)$  değerine sahip  $s_m$  sensörü seçilir
7. while ( $I_R < 0.20$  yada  $q < 0.10 \times |S|$ ) do
8.   while ( $U_v(A, s_m, p_k)$  artış olmadığında) ve ( $\theta < 360$ ) do
9.      $\theta_m = \theta_m + \delta_H$ 
10.   Eğer görünürlük fayda değerinde artış yoksa,  $\theta_m$  başlangıç değerini alır.
11.   for ( $\sigma = -90 + \beta_L$ ;  $\sigma \leq 90 - \beta_U$  ;  $\sigma = \sigma + \delta_T$  ) do
12.     Yeni  $\sigma$  değeri ile  $U_v(A, s_m, p_k)$  hesapla; ve en yüksek değeri sakla.
13.   Eğer görünürlük fayda değerinde artış yoksa,  $\sigma_m$  başlangıç değerini alır.
14.    $q \leftarrow q + 1$ 
15.   En düşük  $\min s_i \in S U_v(A, s_i, p_k)$  değerine sahip bir başka  $s_m$  sensörü seçilir.
16. endwhile

```

Şekil 17: Yerel arama evresinin adımları

1. Bireydeki en düşük görünürlük değerine sahip sensör seçilir.
2. Seçilen sensörün yatay yönelme açısı (θ), δ_H artırılarak her bir artırma sonucunda o sensörün görünürlük değeri hesaplanır ($\theta + \delta_H$, $\theta + 2\delta_H$, $\theta + 3\delta_H$, ...). İlk durumdaki yatay yönelme açısı ve ulaşılan tüm alternatifler arasından en iyi görünürlük değerine sahip sonuç seçilir.

Buradaki artış miktarının (δ_H) belirlenmesinde ölçek (arazi uzunluğu) ve arazi üreten algoritmalarından sağlanacak olan arazinin karmaşıklık boyut parametresi göz önüne alınarak her arazi için sabit bir açı değeri bulunur. Örnek olarak, az sayıda poligon var ise δ_H büyük olmalı, poligon sayısının fazla olması durumunda ise δ_H değeri azaltılmalıdır. Deneyisel çalışmalarımızda δ_H değeri 25° olarak hesaplanıp kullanılmıştır.

3. Bir önceki adımda yatay yönelme açısı belirlenen sensör için bu kez de dikey yönelme açısı (tilt angle) için alternatifler denenir. Normalde bir sensörün dikey yönelme açısı (σ) aşağıdaki aralıkta olabilir:

$$-90 + \beta^L \leq \sigma \leq 90 - \beta^U \quad (24)$$

Yukarda verilen eşitsizlikte, β^L (dikey) alt görüş açı değerini ve β^U (dikey) üst görüş açı değerini göstermektedir. Bu aşamada yatay yönelme açısına benzer şekilde dikey yönelme açısı için bir

artış miktarı (δ_T) belirlenir. Sensör dikey yönelme açısı, başlangıç değerinden başlayarak her adımda yukarıdaki aralıkta olacak şekilde δ_T miktarı arttırılarak veya azaltılarak belirlenir. Bu işlem δ_T artış miktarı ile belirlenebilecek tüm alternatifler için tekrarlanır.

4. İkinci ve üçüncü adımlarda, yatay ve dikey yönelme açılarında yapılan değişiklik sonucunda sensörün görünürlük değerindeki değişiklik %20'den az ise veya incelenen sensör sayısı çözümdeki toplam sensörün %10'undan az ise çözümdeki birinci adıma geçilerek bir sonraki (görünürlüğe göre) en kötü sensör seçilir.

Görünürlük değerindeki değişiklik bulunurken görünürlük değer toplamları ile aynı sensörlerin yeni durumdaki görünürlük değer toplamları karşılaştırılarak %20 iyileşme sağlanıp sağlanmadığı kontrol edilir. Bu adımda, farklı sensörlerin aynı poligonları görme işlemi göz önüne alınmadığından toplam değerler, çözümün toplam uygunluk değerini yansıtmayacaktır.

3.2.8 Mutasyon Operatörleri

Mutasyon için düşünülen operatörler aşağıda verildiği şekli ile iki ana kategoride toplanmıştır. Deneysel çalışmamızda bir çözüm boyunca sabit bir kategorinin belirlenmesi ve kategorinin rasgele seçilmesi alternatifleri için performans ölçümü gerçekleştirilmiştir. Her iki durumda da, kategori içinden alternatiflerin seçilmesi rasgele yapılmıştır.

i. Konum Tabanlı Operatörler

Bu grupta sensörlerin konumu (poligon no) dikkate alınarak mutasyon işlemi gerçekleştirilmiştir. Sensörlerin yatay ve dikey yönelme açısı gibi diğer niteliklerini konum tabanlı mutasyon operatörleri değiştirmemektedir. Mutasyon işlemi için bu grup seçildiğinde aşağıdaki üç metottan biri rasgele olarak çözüme uygulanır:

- *Güncelleme.* Mutasyon uygulanacak bireydeki bir sensörün atandığı poligon değiştirilerek başka bir poligona atanır. Bu aşamada, sensörün diğer konumsal özellikleri (yatay ve dikey yönelme açıları) ve tipi değiştirilmez.

Sensör Seçimi: Rasgele olarak veya çözüm içindeki görünürlük değeri en kötü olan sensör seçilir.

Poligon Seçimi: İki yolla gerçekleşir:

1) Sensör arazi üzerinde rasgele seçilen bir poligona taşınır.

2) Sensör arazi üzerinde hangi bölgede bulunuyorsa, aynı bölgeden bir poligon rasgele seçilerek sensör o poligona yerleştirilir. Burada kullanılan bölge kavramı ilk popülasyonu sağlama adımındakinin aynıdır.

- *Silme.* Mutasyon uygulanacak bireyden bir sensör çıkartılır. Çıkarılacak sensör, rasgele seçilebileceği gibi, çözüm içindeki görünürlük fayda değeri en küçük olan da olabilir.
- *Ekleme.* Mutasyon uygulanacak bireye yeni bir sensör eklenir. Poligon seçimi yukarıdakine benzer olarak rasgele bir poligon üzerine veya görünürlük değeri en az olan sensörün bulunduğu bölge içerisinden rasgele bir poligon seçilerek yapılmıştır. Yeni oluşturulan sensörün davranışsal özellikleri (yatay, dikey açıları ve sensörün tipi) rasgele atanır.

ii. Açık Tabanlı Operatörler

Bu grupta, sensörlerin açılal özellikleri (yatay ve dikey yönelme açısı) dikkate alınarak mutasyon işlemi gerçekleştirilmiştir. Mutasyon işlemi için sensör seçimi ya rasgele ya da görünürlük fayda değerinin düşüklüğüne göre seçilir. Bu grupta aşağıda verilen iki farklı metot bulunur ve operatör seçim işlemi rasgele gerçekleşir:

- *Metot 1.* Sensörün yatay yönelme açısının belirlenen aralıkta rasgele değiştirilmesi.
- *Metot 2.* Sensörün dikey yönelme açısının belirlenen aralıkta rasgele değiştirilmesi.

Bu seçim işleminde metotlara farklı önem değerleri için ağırlık (veya önem değeri) verilerek, ağırlıklı olarak yatay yönelme açısının değiştirilmesi, daha sonra dikey yönelme açısının değiştirilmesi ve en az ağırlıklı olarak da sensör tipinin değiştirilmesi göz önüne alınmıştır. Yukarıda verilen her metot için göz önüne alınacak sensör rasgele olarak veya görünürlük değeri en düşük olan seçilebilir. Seçilen tipteki değişiklik de tamamen rasgele olarak sağlanır.

Ayrıca, sensör nitelikleri arasında yer alan sensör tipi için bir mutasyon operatörü tanımlanmıştır. Deneysel çalışmalarımızda her 100 nesilde bir defa o nesilde üretilen bireyden rasgele seçilen bir sensörün sensör tipi rasgele değiştirilir. Bu işlem yukarıda tanımlanan iki grup mutasyon operatörlerinden herhangi biriyle birlikte uygulanır.

3.3 Hareketli Sensörler için Genel Güzergah Planlama Adımı

Projemizin ikinci evresinde, bir bölgeyi maksimum kapsayacak biçimde hareketli sensörlere yönelik güzergahlarının planlanması probleminin gerçekleşmesi sağlanmıştır. Güzergah planlama problemi için geliştirdiğimiz çözüm iki adımdan oluşmaktadır.

1. Genel Güzergah Planlama Adımı. Bu adımda arazinin yapısına göre ulaşılabilir her bölge dikkate alınarak sadece görünürlük ve mesafe kriterlerine göre sensör güzergahlarının kontrol noktaları oluşturulur.
2. Yerel Güzergah Planlama Adımı. Yerel güzergah planlama adımında ise genel güzergah planlama aşamasında oluşturulan güzergah kullanılarak görünürlük ve mesafe kriterlerinin yanı sıra maliyet, düzgünlük ve hareketli nesnelere (sensörler dahil) karşı çarpışma kontrolü dikkate alınmaktadır. Bu adım sonunda kontrol noktaları arasında yer alan ara noktalar belirlenerek sensörler güzergahlarının belirlenmesi tamamlanır.

Raporumuzun bu bölümünde genel güzergah planlama adımı ayrıntılı olarak anlatılmıştır. Genel güzergah planlama adımı aşağıda verildiği şekilde üç aşamada gerçekleştirilmiştir:

1. “Olasılıksal Yol Haritası” yöntemi ile arazi üzerinde erişilebilecek noktalar ve noktalar arası bağlantıların oluşturulduğu bağlantırlık çizgesinin oluşturulması
2. Seçilen noktalar arasından arazide erişilmesi güç olan bölgedeki noktalar öncelikli olacak şekilde, arazinin genelinden sınırlı sayıda noktanın “kontrol noktası” olarak seçilmesi
3. Başlangıç ve hedef noktaları belirlenmiş hareketli sensörler için, mesafe ve görünürlük değerleri kullanılarak oluşturulan VPD (visibility-per-distance) fonksiyonu yardımıyla seçilen kontrol noktalarının sensörlerin güzergahlarına atanması

3.3.1 Olasılıksal Yol Haritasının Oluşturulması

Çoklu güzergah planlamasında aynı ortam içerisindeki çoklu sorgular problemin karmaşıklığını çok fazla arttırdığı için, sorgu sayısını dikkate almayan ve örnekleme tabanlı bir yöntem olan olasılıksal yol

haritaları (probabilistic road map - PRM) çalışmamızın bu bölümünde kullanılmıştır (KAVRAKI, 1996). Bu yöntemde hedef “yol haritası” diye adlandırdığımız topolojik bir çizge (graf) veri yapısını oluşturmaktır (LA VALLE, 2006). Başlangıç ve hedef noktası verilen bir problemde güzergah, üretilen yol haritasını kullanarak sabit engellere çarpma kontrolüne gerek duyulmadan oluşturulur. Yol haritalandırma yaklaşımı, koşum süresi açısından da oldukça etkilidir.

Projemiz kapsamında gerçekleştirilen olasılıksal yol haritası yöntemi, literatürde de verildiği üzere iki aşamadan oluşmaktadır (KAVRAKI, 1996):

- *Ön hazırlık aşaması:* Yol haritası için gerekli olan arama alanını simgeleyen çizgenin oluşturulma aşamasını içerir.
- *Sorgu aşaması:* Bu aşamada güzergah için başlangıç ve hedef noktaları alınır. Çizge üzerindeki en yakın noktalarla bu noktalar birleştirilir ve belirlenen kriterlere göre (mesafe, gizlilik, görünürlük vs. gibi) çizge üzerindeki noktalardan bir kısmı kullanılarak güzergah oluşturulur.

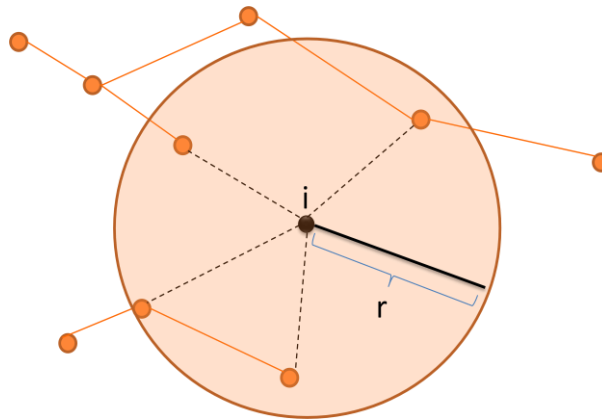
Ön Hazırlık Aşaması

Öncelikle, PRM’de kullanılacak aday nokta sayısı belirlenir ve bu sayıda aday nokta arazi üzerine aşağıda verilen iki farklı yöntemden biri kullanılarak yerleştirilir:

- Hiçbir kısıt gözetmeksizin aday noktalar arazi üzerine rasgele yerleştirilir.
- Arazi, aday nokta sayısı kadar kare şeklinde bölgelere ayrılır. Her bölgeye bir aday nokta %80 olasılıkla rasgele, %20 olasılıkla bölgenin merkezine yerleştirilir.

Daha sonra, belirlenen aday noktalar için uygulanacak erişilebilirlik testinin hangi noktalar arasında yapılacağı belirlenir. İki nokta arasında yapılan erişilebilirlik testinde kullanılacak ikili noktalar kümesi, aşağıda verilen komşuluk tabanlı yöntemlerden biri kullanılarak belirlenebilir (LA VALLE, 2006).

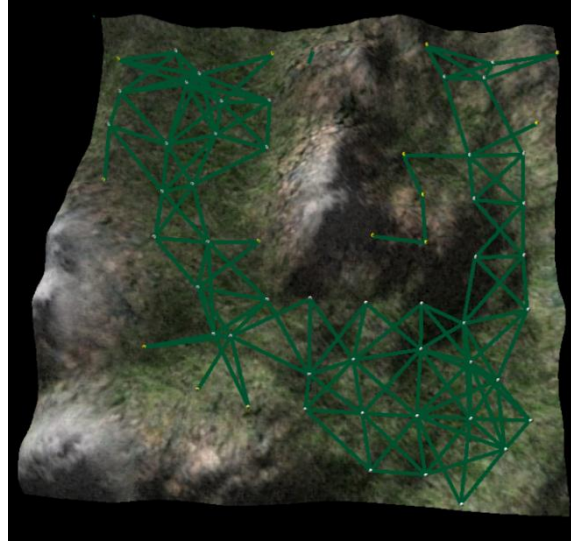
1. *En yakın K komşu:* Belirlenen i noktasına en yakın K adet nokta, i noktasına komşuluğu olan noktalar olarak belirlenir.
2. *Yarıçap:* i noktasının merkez alındığı r yarıçapındaki bir dairenin içerisinde yer alan noktalar i noktasının komşusu olarak belirlenir (Şekil 18). Çalışmamızda bu yöntem gerçekleştirilmiştir.



Şekil 18: Erişilebilirlik testinde kullanılacak noktaların yarıçap yöntemiyle seçimi

Yarıçap yöntemi ile oluşturulan ikili noktalar kümesi erişilebilirlik testinden geçirilmektedir (Şekil 18). Bir noktadan diğer bir noktaya erişilmesi, bu iki nokta arasındaki poligonların eğiminin önceden belirlenmiş bir üst limitten (örneğin, 45o) daha yüksek olmaması ve

herhangi bir sabit engelin bulunmaması anlamına gelmektedir. Erişilebilir her ikili nokta grubu, yol haritası için kullanılacak olan çizgede birer kenarı ifade etmektedir. Komşuluğundaki hiçbir noktayla bağlantısı olmayan noktaları ise çizgeye yerleştirmemekteyiz. Sensör platformlarına bağlı kısıtlar göz önüne alındığı için bu yöntemle çizgedeki noktaları kullanarak oluşturacağımız güzergahların gerçekçiliği sağlanmış olur. Şekil 19’de temsili bir arazi için üretilmiş çizge örneği ile arazideki erişilebilir alanlar görülmektedir.



Şekil 19: Olasılıksal yol haritası yöntemiyle üretilen noktalar ve bağlantıları

Sorgu aşaması:

Ön hazırlık aşamasında oluşturulan çizge veri yapısı kullanılarak, istenilen başlangıç ve hedef noktaları arasında güzergah oluşturma işlemi bu aşamada gerçekleştirilmektedir.

Önceki adımda oluşturulan çizgenin bütün noktaların birbirine bağlı bir çizge olması, her türlü sorguya cevap verebilmesi açısından zorunludur. Başlangıç ve hedef noktalarına göre güzergah oluşturulurken, çizge üzerindeki hangi nokta ve kenarların kullanılacağı, Dijkstra algoritması gibi farklı kısa yol algoritmaları yardımı ile veya A^* gibi bazı sezgisel algoritmalar yardımı ile belirlenebilir. Çalışmamızda Dijkstra kısa yol algoritmasından yararlanılmıştır. Sonuçta, istenilen sorguya göre üretilen her türlü güzergah, arazinin eğimli yapısı veya ortamdaki nesneler gibi etkenlerden arındırılmış olacaktır.

Olasılıksal Yol Haritasında Çeşitli Parametrelerin Etkisi

Olasılıksal yol haritasının oluşturulması aşamasında aşağıda verilen parameterlerin etkisin olduğu gözlemlenmiştir.

- **Aday nokta sayısı:** Aday nokta sayısı yol haritasının arazi yapısına ne kadar uyumlu olduğunu belirlemede önemli bir etkidir. Fazla sayıda nokta dar bölgeler için bile bir güzergah oluşturmada yardımcı olurken, az sayıda noktadan oluşturulan yol haritası sadece düzlükleri gösterebilecek kadar basit kalabilir.
- **Komşuluk mesafesi:** Komşu nokta seçiminde “yarıçap” yöntemini kullandığımızda, yarıçap uzunluğu çizgedeki kenar sayısını doğrudan etkilemektedir. Yarıçap değeri düşük alındığında erişilebilen noktalar erişilemez olabilirken, çok yüksek alındığında ise kenar sayısındaki fazlalıktan dolayı özellikle sorgu aşamasının çalışma süresi oldukça artmaktadır.

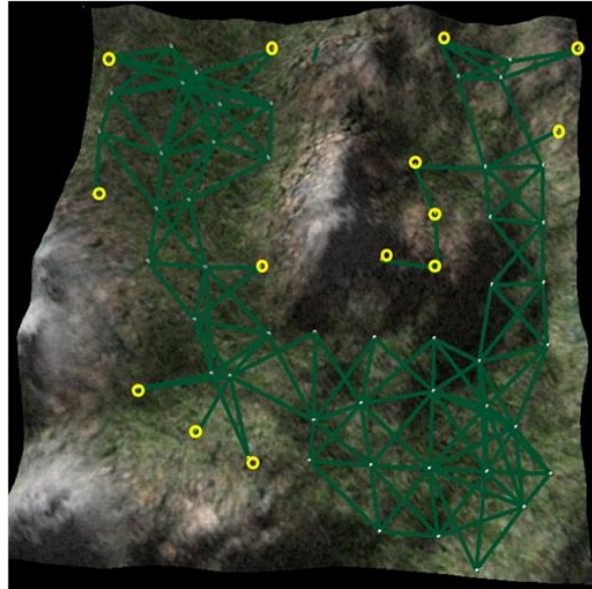
- *Poligon eğimi üst limiti:* Sensörün üzerinden geçebileceği en dik poligonun eğimi arttıkça üretilen çizgedeki nokta sayısı ve erişilebilen bölgenin alanı artmış olur. Maksimum eğim değeri azaldıkça, nokta sayısı ve erişilebilen bölgenin büyüklüğü, dolayısıyla sensörün görüntüleyebileceği alan azalmış olur.

Arazide erişilebilen (kullanılabilen) alanın büyüklüğünü etkileyen bu parametrelerin tayini deneysel ön çalışmamızda yer almıştır.

3.3.2 Kontrol Noktası Seçimi

Geliştirdiğimiz sistemde, oluşturulacak güzergahlar için kontrol noktaları, olasılıksal yol haritası yöntemi (PRM) ile belirlenen noktalar arasından seçilmektedir. Bu aşamada, arazinin erişimi mümkün olan her bölgesinden kontrol noktası seçimi hedeflenmiştir. Kontrol noktası seçim işlemi iki aşamada gerçekleştirilmektedir:

1. İlk aşamada, çizgedeki noktalar arasından en az kenar sayısına sahip olan noktalar, aralarında belirli bir mesafenin sağlanması durumunda, kontrol noktası olarak seçilir. Kenar sayısı en az olan noktalar, genellikle dağ eteği, vadi vs. gibi sensörlerin tırmanamayacağı yüksek eğimli bölgelerin başlangıcında olduğundan dolayı bu bölgelerin etrafında hareket ederek arazinin görünürlük değerini arttırmak için, kontrol noktaları bu bölgelerden seçilir. Seçilen noktalardan bazılarının arazinin sınırlarına çok yakın olması durumunda ise, sınır bölgelerde görünürlük değeri diğer bölgelere göre daha düşük olacağından, güzergahın etkinliğini arttırmak amacıyla seçilen o noktalar, komşuluğundaki noktalardan biri ile yer değiştirir.
2. İkinci aşamada, güzergahların daha düzgün (gerçekçi) olması ve arazinin geneline ulaşabilmek için, seçilen noktaların bulunduğu bölgeler haricinde arazideki diğer bölgelere yeni noktalar eğer yakınlarında kontrol noktası yoksa eklenir. Öncelik, erişilmesi güç olan bölgelerdeki noktalar olduğu için, kontrol noktaları ağırlıklı olarak bu noktalardan seçilmektedir (Şekil 20).



Şekil 20: Temsili araziden seçilen kontrol noktaları

3.3.3 Kontrol Noktalarının Güzergahlara Atanması

Projemizin genel güzergah planlama adımındaki son aşama, seçilen noktaların başlangıç ve hedef noktaları belirlenmiş güzergahlara atanması işlemidir. Seçilen noktalar teker teker tüm güzergahlara

katkı değerleri sıranmak için eklenir. Katkı değerlerini hesaplamak için benzer bir çalışmada (MOHAN, 2008) yer alan M_{PDV} hesaplamasını problemimize uyarlayarak oluşturduğumuz *VPD (visibility-per-distance) fonksiyonu* kullanılmıştır. *VPD fonksiyonu* mesafe ve toplam görünürlük değerlerindeki değişim oranı kullanılarak oluşturulurken M_{PDV} belirli bir noktanın görünürlük değeri ve o noktaya erişim için gereken mesafeyi dikkate almaktadır. Eğer seçilen nokta bir güzergaha belirli bir mesafeden daha yakınsa, *VPD fonksiyonu* ile test edilmeden doğrudan güzergaha eklenir. Diğer seçilen noktalar sırasıyla fonksiyon değerine göre en uygun sensörün güzergahına atanır.

VPD (visibility-per-distance) Fonksiyonu:

Çözümdeki herhangi bir sensörün, $s_i \in S = \{s_1, s_2, \dots, s_n\}$ güzergahına atanacak kontrol noktaları, $k_q \in K = \{k_1, k_2, \dots, k_p\}$ ve kontrol noktaları ile oluşturulan güzergahtaki sensörün görünürlük değeri $V(s_i, K_i)$ kullanılarak oluşturulan *VPD fonksiyonu* aşağıdaki formülde tanımlanmıştır (25). Formüldeki $\Delta V(S, k_q)$, s_i sensörüne k_q kontrol noktası atandıktan sonra oluşan yeni güzergahla birlikte çözümün yeni görünürlük değerinin eski değerine oranını, $\Delta d(s_i, k_q)$ ise s_i sensörünün güzergahına k_q kontrol noktası eklendikten sonra oluşan güzergah ile eski güzergahın uzunlukları arasındaki oranı ifade eder.

$$VPD = \frac{\Delta V(S, k_q)^\alpha}{\Delta d(s_i, k_q)^{5-\alpha}} \quad (25)$$

Formüldeki α parametresi ($0 \leq \alpha \leq 5$), oluşturulan güzergahın mesafe ve görünürlük kriterlerinden hangisine ne kadar ağırlık verildiğini göstermektedir. Örneğin, $\alpha = 5$ olursa kontrol noktası seçiminde sadece görünürlük değeri dikkate alınırken, $\alpha = 0$ olması durumu da kontrol noktası seçiminde sadece güzergahın uzunluğunun dikkate alındığını gösterir. Deneysel çalışmamızda (0,5) aralığında çeşitli α değerleriyle testler gerçekleştirilmiştir.

3.4 Hareketli Sensörler için Yerel Güzergah Planlama Adımı

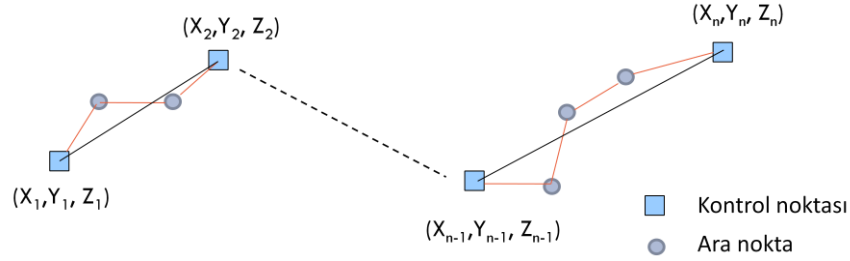
Genel güzergah planlama aşaması ile üretilen kontrol noktalarına ek olarak, çalışmamız kapsamında sunulan karma evrimsel algoritma tabanlı çözüm yardımı ile kontrol noktaları arasında yer alacak olan ara noktalar üretilerek, her sensor için uygun güzergah oluşturulmaktadır. Bu gerçeklemede, görünürlük ve mesafe kriterlerinin yanısıra maliyet, düzgünlük ve hareketli nesnelere karşı çarpışma kontrolü (sensörler dahil) dikkate alınmaktadır. Algoritmadaki melezleme işlemi, başlangıç popülasyonu oluşturulmasında kullanılan probleme özgü yöntemlerle yine probleme özgü çaprazlama ve mutasyon operatörleri yardımı ile sağlanmıştır.

Yerel güzergah planlama aşamasında steady-state (sabit-popülasyonlu) evrimsel algoritma yaklaşımı kullanılmış olup, her nesilde iki ebeveyn bireyden çaprazlama sonucu tek bir birey üretilmekte ve mutasyon ile güzergahın uygunluğuna göre onarım metodu uygulanmaktadır. Sonuçta üretilen yeni çözüm popülasyondaki en kötü bireyin yerine yazılmaktadır. Ebeveyn seçimi ise turnuva metodu ile gerçekleştirilmektedir. Bu bölümde evrimsel algoritma tabanlı çözümümüz ile ilgili detaylar sunulacaktır.

3.4.1 Kromozom Yapısı

Öncelikle, her güzergah kromozomda güzergahı oluşturan kontrol noktalarının koordinatları ile temsil edilir. Kontrol noktaları kromozomda başlangıç noktasından hedef noktasına doğru koordinatları dikkate alınarak sıralanır. Arazi üzerinden seçilen kontrol noktaları arasında oluşturulan güzergahlar

herhangi bir sensöre ait güzergahın alt güzergahını oluşturur. Her alt güzergahta ise belirlenen hedefler doğrultusunda ara noktalar bulunmaktadır ve bu noktalar da kromozomu oluşturan diğer noktalardır (Şekil 21).



Şekil 21: Bir sensör güzergahının kromozomdaki yapısı

3.4.2 Uygunluk Fonksiyonu

Kromozomun uygunluk fonksiyonu sensör güzergahlarının değerlendirilmesiyle hesaplanmaktadır. Güzergah değerlendirmesinde görünürlük, maliyet, düzgünlük ve erişilebilirlik etkenleri dikkate alınmakta olup, bu etkenlerle oluşturulan güzergah için uygunluk fonksiyonu aşağıda verilmiştir:

$$f(R) = V(A, S, P)^{w_v} (w_d D(R) + w_s S(R))^{1-w_v} - w_c C(R) \quad (26)$$

Yukardaki formüldeki w_v , w_d , w_s , w_c sembolleri sırasıyla görünürlük ($V(A, S, P)$), maliyet ($D(R)$), düzgünlük ($S(R)$) ve erişilebilirlik ($C(R)$) kriterlerinin çözüme olan etki oranıdır ve $0 \leq w_v, w_d, w_s \leq 1$, $w_c > 1$. Çalışmamızın temel hedefi arazi üzerinde kapsama olduğu için çözümdeki maliyet ve düzgünlük parametre grubu ile görünürlük parametresi problemimizde ayrı düşünülerek w_v parametresi bu iki grubun çözümdeki ağırlık oranı için kullanılmaktadır.

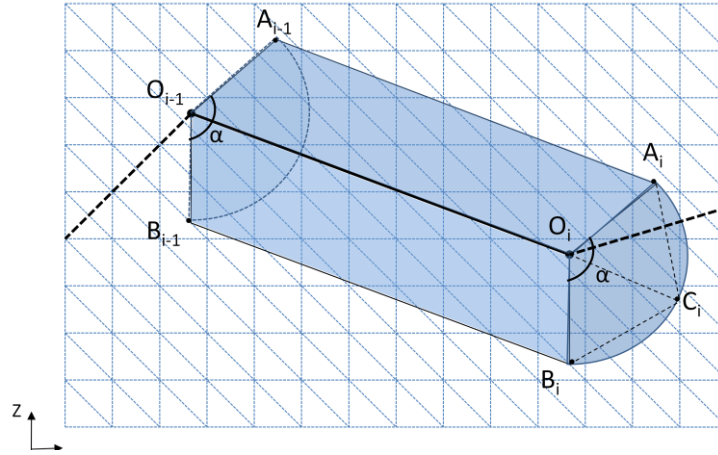
Görünürlük Bileşeni

Sabit sensörlerde görünürlük değerinin hesaplanması için geliştirdiğimiz yöntemi esas alarak, hareketli sensörler için yeni bir görünürlük değeri hesaplama yöntemi belirledik. Hareketli sensörlerin algılama problemi, sabit sensörler için algılama problemine getirilen çözüm gibi, öncelikle seçilen sensörün algıladığı arazi bölümü ve bu arazi bölümü içerisinde yer alan poligonları içeren kümenin oluşturulmasını içermektedir. Daha sonra da bu kümede yer alan her bir poligonun görünürlük değeri hesaplanır.

Güzergahlar sınırlı sayıda noktalardan oluştuğu için, bir güzergahı doğru parçalarının birleşimi şeklinde ifade edebiliriz. Hareketli sensörün algıladığı alanın hesaplanması, yatay FOV ve dikey FOV testlerinden oluşmaktadır. Bu testler güzergah üzerindeki her doğru parçası için ayrı ayrı uygulanır.

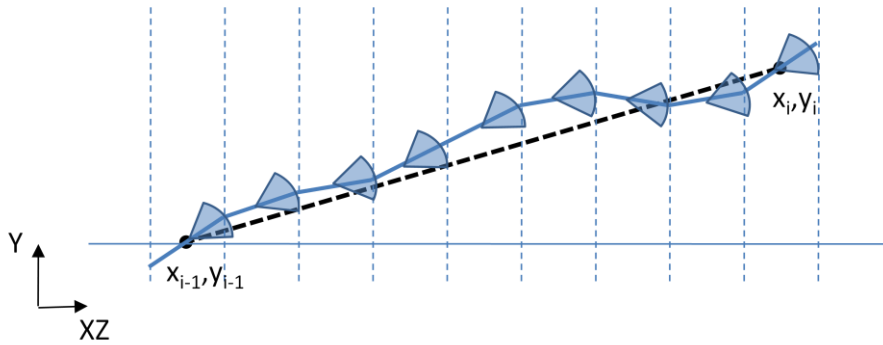
- **Yatay FOV testi.** Yatay FOV testi, sabit sensörler için uygulanan yatay FOV testine benzer bir işlemden oluşur. Yatay FOV testi, XZ düzleminde sensör spesifik parametreler olan yatay görüş açısı, yatay yönelme açısı ve görüş mesafesi parametreleri ile sensörün hareket ettiği doğru parçasının uzunluğuna göre yapılan filtreleme işlemini içerir.

Sabit sensörler için uygulanan yatay FOV testinde görüş alanı içerisindeki noktaların seçimi, noktanın çizilen iki üçgen içerisinde olup olmadığıyla test edilirken, Şekil 22’de görüldüğü gibi O_{i-1} noktasında bulunan bir sensörün O_i noktasına doğru hareket etmesiyle oluşan görüş alanındaki nokta seçimi, çizilen iki yamuk ($O_{i-1}A_{i-1}A_iC_i$ ve $O_{i-1}B_{i-1}B_iC_i$) arasında olup olmadığıyla kontrol edilir.



Şekil 22: Güzergah üzerindeki bir doğru parçasına uygulanan yatay FOV testi

- **Dikey FOV testi.** Sabit sensörlerde geliştirdiğimiz gibi, bir doğru parçası için yatay FOV testi tamamlandığında, yatay FOV'a göre algılanan arazi noktalarının listesi dikey FOV testi için girdi olarak kullanılır. Dikey FOV testi sabit sensörlerde uygulanan dikey FOV testi ile aynı adımları içermesine rağmen, dikey FOV testinde kullanılan parametrelerden dikey yönelme açısı güzergah üzerindeki poligonların eğimine göre farklılık göstermektedir. Bu aşamada, hareketli sensörler için dikey FOV testi, güzergahtaki doğru parçasının üzerinden geçtiği her poligon için farklı dikey yönelme açıları ile tekrarlanır (Şekil 23).



Şekil 23: Doğru parçasına uygulanan dikey FOV testi

Güzergah üzerindeki herhangi bir poligon üzerinden yapılan dikey FOV testinden geçen noktalar, görüş çizgisi algoritmaları yardımıyla (bkz. Janus ve Bresenham LOS algoritmaları) sensör ile nokta arasında kesintisiz görüşün olup olmadığı test edilir. Son aşamada, görüş çizgisi testinden geçen noktaların görünürlük değerleri sabit sensörler için kullanılan görünürlük hesaplama yöntemi ile hesaplanır.

- **Görünürlük hesaplaması.** Çözümde yer alan tüm sensör güzergahları için, yukarıda açıklanan hareketli sensörlerde algılama yöntemleri kullanılarak, hesaplanan arazi üzerinde görüntülenen noktaların görünürlük değerlerinin toplamı ile bulunur.

$$V(A, S, P) = \frac{\sum_{p_i \in P} V(S, p_i) \times W_{p_i}}{\sum_{p_i \in P} W_{p_i}} \quad (27)$$

Formülde yer alan W_{p_i} p_i poligonunun ağırlığını, $V(S, p_i)$ poligonun görünürlük değerini ifade eder. Formüldeki bir poligonun görünürlüğünün hesaplanması aşağıda belirtilmiştir.

$$V(S, p_i) = \frac{\sum_{b \in p_i} \max_{s_j \in S, p_k \in R_{s_j}} \{V_{s_j}(s_j, p_k, b)\}}{n} \quad (28)$$

Bu formülde, çözümdeki sensörler (S) güzergahları (R) üzerinde hareket halindeyken, p_i poligonunun en yüksek görünürlük değerini alan sensör (s_j) ve sensörün güzergahı (R_{s_j}) üzerindeki poligon (p_k) seçilir ve p_i poligonunun görünürlük değeri hesaplanır. p_i poligonunun görünürlük değeri poligonu oluşturan dört noktanın görünürlük değerlerinin ortalaması ile hesaplanır.

Maliyet Bileşeni

Bu bileşen değerini çözümdeki güzergahların uzunluğuna göre hesaplamaktayız. Çözümün maliyetinin hesaplanmasında, güzergahların maliyetlerinin ($d(p_s)$) ortalaması ile en uzun ($\max(d(p_s))$) ve en kısa güzergahın ($\min(d(p_s))$) aralarındaki oran kullanılır. Çözümdeki tüm sensörlerin birbirlerine yakın güzergah uzunluklarına sahip olması, görevin daha kısa sürede ve daha etkin şekilde tamamlanmasına yardımcı bir etkidir. Formüldeki λ sabiti, bu parametrenin çözümün maliyet hesaplanmasındaki etkisini düzenlemek için kullanılmıştır. Deneyler, aksi belirtilmedikçe, $\lambda=0.50$ olarak alınmıştır.

$$D(R) = \left(\sum_{s \in S} d(R_s) / |S| \right) \times \left(\min_{s \in S} \{d(R_s)\} / \max_{s \in S} \{d(R_s)\} \right)^\lambda \quad (29)$$

$$d(R_s) = 2 - \left(\sum_{i \in R_s} d(m_i, m_{i+1}) / E(r) \right)$$

Güzergahın normalize edilmiş maliyet hesaplaması, güzergahtaki her doğru parçasının uzunlukları toplamının kullanılan sensörün yatay görme açısı, arazinin büyüklüğü ve arazideki poligonların ortalama eğimleri kullanılarak oluşturulan tahmini güzergah uzunluğuna bölünmesi ile hesaplanır.

Tahmini güzergah uzunluğu ($E(r)$) hesaplamasında, öncelikle arazinin iki boyutlu alanı arazideki poligonların ortalama eğimi ile çarpılıp sensör sayısına bölünerek, sensör başına tahmini görüntülenecek alan (APS) hesaplanır. Daha sonra, görüntülenecek alan yatay FOV testinde hesaplanan alana eşit olacak şekilde bir doğru parçası uzunluğu hesaplanır. Hesaplanan bu uzunluk κ sabiti ($\kappa=1.75$) ile çarpılarak tahmini güzergah uzunluğu bulunur.

$$E(r) = (APS - \Delta_s^2 \times \sin(\alpha / 2)) / (2 \times \Delta_s \times \sin(\alpha / 2)) \times \kappa \quad (30)$$

Düzgünlük Bileşeni

Sensörlerin hareket kabiliyetinin çok yüksek olduğunu farzetsek de, özellikle yön değiştirirken çıkabilecek sorunlar nedeniyle, sensörlerin dönüş açılarının önceden belirlenmiş bir değerden (mesela 30° gibi) daha düşük olmaması gerekir. Düzgünlük hesaplamasında, güzergahtaki tüm ardışık ikili doğru parçalarının arasındaki açılar ($\alpha(l_i, l_{i+1})$) içinden sensörün hareketini engelleyecek düşüklükte olan açı değerleri göz önüne alınır. Formülde toplam yerine çarpımın kullanılmasının nedeni, güzergah üzerinde uygun açı değerine sahip olmayan noktalar arttıkça düzgünlük değerinin o oranda azalmasına sebep olmasıdır.

$$s(R_j) = 1 \times \prod_{i \in R_j} \alpha(l_i, l_{i+1}) / \min(\alpha), \text{ where } \alpha(l_i, l_{i+1}) < \min(\alpha) \quad (31)$$

Çözümün toplam düzgünlük değeri ise sensör güzergahlarının düzgünlük değerlerinin ortalamasıyla hesaplanır.

$$S(R) = \sum_{i=1}^n s(R_i) / n \quad (32)$$

Erişilebilirlik Bileşeni

Güzergah üzerindeki her doğru parçasının erişilebilirlik kontrolü, doğru parçasının başlangıç ve varış noktaları arasındaki erişilebilirlik testi ile gerçekleşir. Olasılıksal yol haritasında da kullanılan bu yöntemde, doğru parçasının üzerinden geçtiği poligonların eğimi ve sabit engellerin varlığının kontrolü yapılır.

Çözümün erişilebilirlik değeri, güzergahların üzerindeki doğru parçalarından erişilebilirlik testinden geçemeyen poligonun sayılarının ($o(p_i)$) toplamının güzergahların üzerindeki toplam poligon sayısına (R_s) bölünmesi ile hesaplanır.

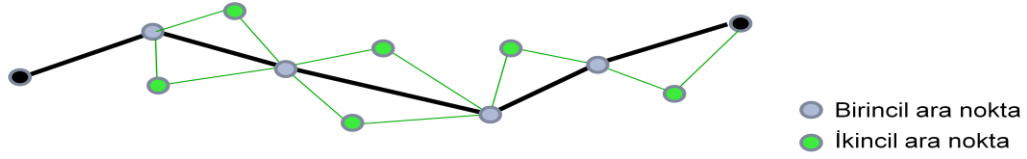
$$C(R) = \frac{\sum_{s \in S} \sum_{i \in R_s} o(p_i)}{\sum_{s \in S} R_s} \quad (33)$$

3.4.3 Başlangıç Popülasyonun Oluşturulması

Çalışmamız kapsamındaki evrimsel algoritma tabanlı çözümün başlangıç popülasyonunun oluşturulması üç aşamada gerçekleştirilmiştir:

- *Kontrol noktalarının seçimi.* Genel güzergah planlama aşamasında popülasyondaki çeşitliliği arttırmak için başlangıçta üretilecek her bir bireye ait kontrol noktaları, sensörlerin güzergahlarına atanan kontrol noktaları ile bu noktaların komşuluğundaki noktalar arasından rasgele seçilir.
- *Ara noktaların seçimi.* Seçilen kontrol noktaları arasında oluşturulan alt güzergahlar için ara noktaların seçimi, birincil ve ikincil ara noktaların seçimi ile gerçekleştirilir. Birincil ve ikincil ara noktalar ise olasılıksal yol haritasında üretilen noktalar arasından seçilir. Birincil ara noktalar, en kısa yol algoritması kullanılarak üretilen alt güzergahtaki ara noktalardır. Birincil ara noktalardan herhangi ikisi arasına, olasılıksal yol haritası yönteminde üretilen çizgedeki noktalardan biri eklendiğinde bu iki nokta arasında alternatif bir bağlantı sağlanıyorsa, bu nokta ikincil ara noktalar grubuna kaydedilir (Şekil 24). Birincil ve ikincil ara noktalar arasından nokta seçimi işleminde noktalar öncelikle başlangıç noktasından hedef noktasına

doğru yakınlıklarına göre sıralanır ve sırlanmış şekilde iki gruba ayrılır. Seçilecek nokta sayısı belirlendikten sonra iki gruptan eşit sayıda olacak şekilde noktalar rasgele seçilir.



Şekil 24: İki kontrol noktası arasında güzergah oluşturmada kullanılan birincil ve ikincil ara noktalar

- *Yatay ve dikey yönelme açılarının belirlenmesi.* Bir güzergah üzerindeki sensörün en yüksek görünürlük değerine erişmesini sağlamak için yatay ve dikey yönelme açılarının güzergahtaki her doğru parçası için farklı değerler alması gerekebilmektedir. Örneğin sensörün güzergahı üzerindeki bir bölge başka bir sensör tarafından da görüntülenebiliyorsa sensörün yatay yönelme açısını değiştirerek farklı yerleri görüntüleyebiliriz.

Başlangıç popülasyonunda dikey ve yatay yönelme açılarını rasgele belirleyebileceğimiz gibi yatay yönelme açısını üzerindeki doğru parçası ile aynı doğrultuda ayarlarken dikey yönelme açısını üzerinden geçtiği poligonlarla paralel olacak şekilde de belirleyebiliriz.

3.4.4 Çaprazlama Operatörleri

Kontrol noktaları üzerinde yapılan çaprazlama işlemi, seçilen iki ebeveyn çözümdeki tüm sensör güzergahları için tekrarlanır. Çalışmamızda iki farklı çaprazlama operatörü tanımlanmıştır:

- *Görünürlüğe Bağlı Çaprazlama.* Çaprazlama işlemi; seçilen her iki ebeveyndeki sensörlere ait güzergahlardaki kontrol noktalarının sayısı aynı ve kontrol noktalarının yerleri birbirine yakın olduğundan, alt güzergahlar arasında gerçekleştirilebilir. Çaprazlama işlemi, bir güzergahtaki ardışık her iki kontrol noktası arasındaki alt güzergahın toplam fayda değerlerine göre gerçekleştirilir.

Öncelikle, her iki ebeveyndeki tüm güzergahların toplam görünürlük değeri hesaplanır ve iki ebeveyn arasında sırayla seçilen alt güzergahlardan en yüksek görünürlük değerine sahip alt güzergah, çaprazlama sonucu oluşacak yeni bireye taşınır. Bu işlem tüm sensörler için gerçekleştirilebileceği gibi, rasgele seçilen bir güzergaha da uygulanabilir.

- *Rastgele Çaprazlama.* Çaprazlama işlemi önceki yöntemdeki gibi alt güzergahlar arasında gerçekleştirilir. Çaprazlama işleminde ebeveynlerden alt güzergah seçimi rasgele gerçekleşir ve güzergahtaki alt güzergahların belirli kısmına uygulanır (40%). Geride kalan alt güzergahlar için ebeveynlerden toplam uygunluk değeri yüksek olan alt güzergah seçilerek güzergahın kalan kısmı tamamlanır.

3.4.5 Mutasyon Operatörleri

Güzergahta yer alan herhangi iki ardışık doğru parçası arasındaki dar açıdan veya güzergahtaki belli bir bölgeye erişimin sabit engel veya arazideki yüksek eğimin etkisiyle engellenmesinden dolayı sensörün hareket etmesine engel oluşturabilecek durumları düzenlemek için mutasyon operatörlerini uygulamaktayız. Üç farklı mutasyon operatörü gerçekleştirilmiştir:

- *Ara nokta ekleme:* Yukarıdaki kriterlere göre seçilen iki ardışık noktanın çevresine rasgele yeni bir nokta eklenir. Eğer güzergah üzerinde sensörün hareket etmesine engel olabilecek durum bulunmuyorsa, görünürlüğü arttırmak amacıyla en düşük görünürlüğe sahip alt güzergaha, ara noktalara belirli bir mesafeden yakın olması koşuluyla belirli bir olasılıkla, yeni nokta rasgele eklenir. Bu işlem bireydeki tüm güzergahlar için tekrarlanır.
- *Ara nokta güncelleme:* Yukarıdaki kriterlere göre seçilen iki ardışık ara noktadan biri komşuluğunda bir yere taşınır. Eğer güzergahtaki tüm doğru parçaları yukarıdaki kriterlere uygunsa, arazinin görünürlüğünü arttırmak için en düşük görünürlüğe sahip alt güzergahtan rasgele seçilen bir noktanın yeri komşuluğundan seçilen rasgele bir noktaya taşınır. Bu işlem bireydeki tüm güzergahlar için tekrarlanır.
- *Ara nokta silme:* Yukarıdaki kriterlerden sadece aç kısıtını sağlamayan noktalardan biri rasgele seçilerek silinir.
- *Kontrol noktası güncelleme:* Rasgele seçilen bir kontrol noktasının yeri komşuluğundaki bir nokta ile değiştirilir.

3.4.6 Hareketli Engellere Karşı Çarpışma Kontrolü

Çalışmamızda geliştirdiğimiz çarpışma kontrolü modülü, bir çözümde yer alan sensörlerin kendi aralarında çarpışmayı engelleyici yöntemleri içerir. Bu amaçla, çözümdeki sensörlerden herhangi ikisinin güzergahında bir kesişme varsa, sensörlerin başlangıç noktasıyla kesişim noktası arasındaki varış mesafeleri hesaplanır. Eğer varış mesafeleri arasındaki fark önceden belirlenmiş bir sınır (threshold) değerinin altında ise, sensörlerin birbirlerine çarpma olasılığı yüksektir. Bu durumda güzergahlardan birine kesişme noktasını değiştirecek şekilde, sensörlerin o noktaya varış mesafelerini de dikkate alarak, yeni bir ara nokta eklenir. Böylece sensörlerin kesişme noktasına varış mesafeleri değişeceğinden çarpışma engellenmiş olur. Bir sonraki bölümde anlatılacak olan hareketli sensörlere ait deneysel çalışmada, gerçekleştirilen bu modül belirli aralıklarla tüm popülasyona uygulanmaktadır.

4 Bulgular

4.1 Statik Sensör Konumlandırma Problemine Yönelik Olarak Yapılan Deneyler

Bu bölüm, farklı karakteristiğe sahip araziler ve farklı askeri misyonlar (görevler) için, geliştirilen karma evrimsel algoritmanın performansını ölçmeye yönelik tamamladığımız deneyleri içermektedir. Bu deneysel çalışmalarımızı iki gruba ayırabiliriz. Birinci grup deneyler, çaprazlama operatör türü, mutasyon operatör türü gibi geliştirilen GA-tabanlı algoritmaya özgü parametrelerle yapılan deneyleri içermektedir. İkinci grup deneylerde ise hem algoritmaya bağlı diğer etkenlerin (yerel arama gibi) hem de tanımlanan probleme bağlı etkenlerin (arazi yapısı, misyon türü, sensör görüntüleme türü gibi) çözüme olan katkısı test edilmiştir.

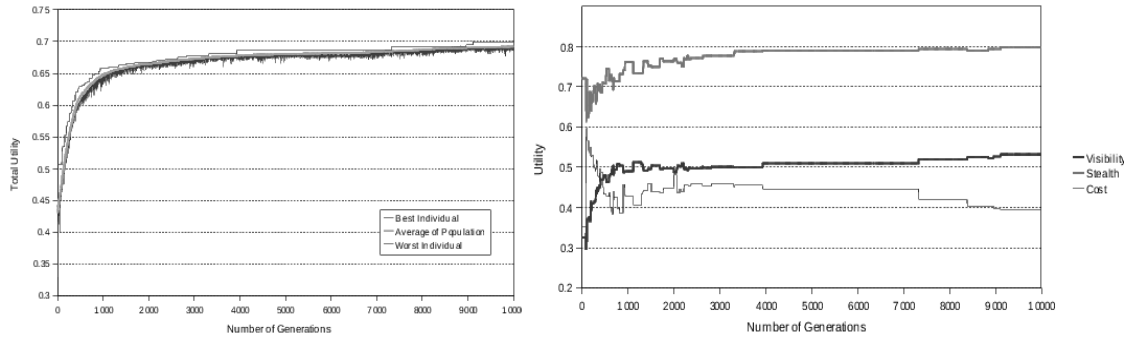
4.1.1 Deney Tasarımına Yönelik Genel Bilgiler

Tasarlayıp C programlama dili ile gerçekleştirdiğimiz karma evrimsel algoritmanın parametreleri ile, problem (üretilen arazi) ile ilgili parametreler ve askeri misyonla ilgili parametrelerin sayısının çok fazla olması ve her bir setin koşum zamanının iş istasyonlarında en az 10 dakika civarında sürmesi toplam deney süresinin çok uzun olmasına yol açmıştır. Her bir deney kombinasyonu için elde edilecek sonuçların istatistiksel açıdan anlamlı olabilmesi için en az 30 tekrarın yapılmasının da zorunlu olduğu düşünüldüğünde, en az 100.000 defa programın çalıştırılması gerekmiş ve bu amaçla İTÜ Ulusal Yüksek Başarımlı Hesaplama Merkezi (UYBHM)'nin olanaklarından faydalanılmıştır. Projemiz kapsamında geliştirilen karma evrimsel algoritma, İTÜ UYBHM'deki her biri Intel Xeon 2.33 GHz işlemcili ve Linux işletim sistemi kurulu clusterda (grup sisteminde) aynı anda birden fazla deneye yönelik olarak çalıştırılması için, MPI (Message Passing Interface) kütüphanesinde yer alan fonksiyonlardan yararlanılmıştır. Grup sisteminde yapılan deneylere yönelik olarak, karma evrimsel algoritmamızda sabit değer alan parametreler Tablo 1'de verilmiştir.

Tablo 1: Deneysel çalışmamızda kullanılan sabit parametrelerin varsayılan değerleri

Parametreler	Değerler
Başlangıç popülasyonu oluşturmada sensör sayısı katsayıları	$\gamma_1=1.00, \gamma_2=1.75$
Sensörün bir bölgeye atanması (bölgenin merkezine, rasgele)	(%20, %10)
Katkı ve komşuluk tabanlı çaprazlamada görüş mesafesi çarpanı	$\mu=0.4$
Offspring oluşturmada ebeveynlerden taşınan sensör sayısı çarpanı	$\varepsilon=0.5$
Yerel aramada iyileştirme oranı	$I_R = \%20$
Yerel aramada bakılan sensör oranı(çözümdeki sensör sayısının en fazla yüzde kaçına bakılacağı)	$S_R = \%10$
Düşman sensör sayısı (tespit, teşhis, tanıma)	(10, 40, 100)
Turnuva büyüklüğü	5
Toplam nesil sayısı	1000

Algoritmanın sonlandırma şartını belirlemek için, görünürlüğün ağırlıklı olduğu bir misyona (göreve) yönelik olarak 10.000 iterasyonluk bir deneyle algoritmanın performansı ölçülmüştür (Şekil 25a). Şekilden de görüleceği üzere 1000'inci iterasyona kadar olan kısımdaki performans artışı ile 1000'inci iterasyondan sonraki kısımdaki performans artışı arasında çok büyük bir fark vardır. Toplam fayda hesaplamasında kullanılan görünürlük, gizlilik ve maliyet kriterleri arasındaki çelişmeden dolayı, testlerde görünürlük ve gizlilik değerleri yükselirken maliyet değeri düşmektedir ve bir noktadan sonra toplam fayda değerinde çok büyük değişiklik olmamaktadır (Şekil 25b). Bu nedenlerden dolayı diğer tüm deneylerimizde çalışmamız 1000 iterasyona kadar test edilmiştir.



Şekil 25: İterasyon sayısına göre karma evrimsel algoritmanın (a) toplam fayda değeri ve (b) 3 fayda kriterinin performans değerlendirmesi

4.1.2 Algoritmaya Özgü Parametreler ile Yapılan Deneyler

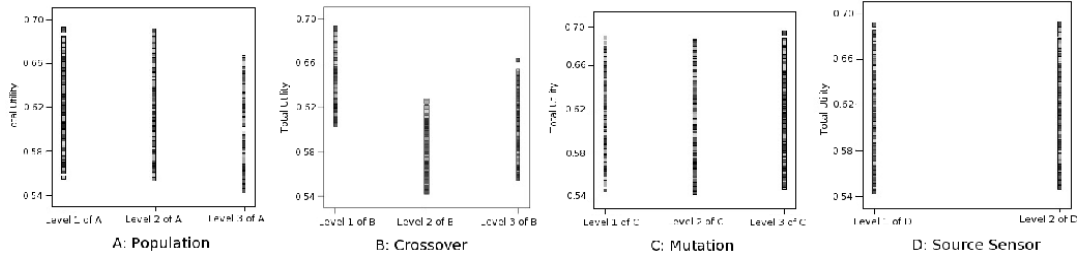
İlk grup deneyler algoritmaya özgü parametrelerin belirlenmesini içermektedir. Tablo 2’de görüldüğü üzere dört farklı algoritmaya özgü parametre bulunmaktadır: a) Popülasyonun büyüklüğü, b) çaprazlama türü, c) mutasyon türü ve d) mutasyon işleminde kullanılan parametrelerden biri olan sensör seçim türü.

Tablo 2: Algoritmaya özgü parametre alternatifleri

Popülasyon Büyüküğü (A)	Çaprazlama (B)	Mutasyon (C)	Mutasyonda Seçilen Sensör (D)
1- 30	1- Katkı tabanlı Çaprazlama	1- Konum tabanlı	1- Rastgele
2- 50	2-Komşuluk tabanlı	2- Açık tabanlı	2- Minimum
3- 100	Çaprazlama	3- Rastgele (1/2) (konum	görünürlük
	3- Kes-Birleştir Çaprazlama	veya açık tabanlı)	

Tablo 2’deki parametreler 54 farklı kombinasyon oluşturmaktadır ve her kombinasyon deneylerde 30’ar defa tekrarlanmıştır. Deneylerdeki tekrarlar bütün kombinasyonlar test edildiği için çalışmamızda “full factorial design” kullanıldı (MONTGOMERY, 2005). 1620 tane testten oluşan bu deney grubunda diğer parametreler varsayılan değerlere ayarlanmıştır. Başlangıç popülasyonu özellikle hiçbir sezgisel yöntem kullanılmadan rasgele üretilmiştir. Yatay yönelme açısı artış miktarı sensör türleri arasında en düşük yatay görme açısının dörtte biri değeri olan 25° olarak belirlenmiştir. Dikey yönelme açısı artış miktarı ise 3° olarak ayarlanmıştır. Bu gruptaki deneylerde zorluk derecesi yüksek bir arazi ile görünürlüğün baskın olduğu bir görev ve teşhis modu dikkate alınmıştır.

Şekil 26’de bu dört algoritmaya bağlı parametrelerin toplam fayda değerine etkisi görülmektedir. Şekilde yer alan düzey (level) ifadesi Tablo 2’de yer alan algoritmaya bağlı parametre değerlerinin karşılığı olmaktadır. Deney boyunca tüm parametreler sabit olduğu için parametrelerin toplam fayda değerine etkileri ve birbirleri arasındaki etkileşimleri hakkında oluşturulan hipotezler kolayca formüle edilip test edilebilir.



Şekil 26: Görünürlük ağırlıklı misyonda dört etkenin toplam fayda değerine etkisi

Görünürlük ağırlıklı misyonda toplam fayda değerinin varyans analizi (ANOVA) Tablo 3'de özetlenmiştir. Tablodan görüleceği üzere, dört ana etken ve başlangıç popülasyonu ve çaprazlama işleminin arasındaki etkileşim toplam fayda değerini ciddi oranda etkilemiştir. “Prob > F” sütunundaki tüm verilerin 0.0500 değerinden düşük olması A, B, D ve AB terimlerinin istatistiksel olarak anlamlılığını (bağımlı değişkeni belirlemede etken olduğunu) göstermektedir.

Tablo 3: Görünürlük ağırlıklı misyonda toplam fayda değerinin varyans analizi

Source	Sum of Squares	df	Mean Square	F Value	p-Value Prob>F	
Model	1.52	9	0.17	1250.65	< 0.0001	significant
A-Population	0.28	2	0.14	1035.91	< 0.0001	
B-Crossover	1.17	2	0.59	4350.85	< 0.0001	
D-Source Sensor	0.011	1	0.011	78.42	< 0.0001	
AB	0.054	4	0.014	100.97	< 0.0001	
Residual	0.22	1610	0.0001348			
Lack of Fit	0.013	44	0.0002956	2.27	< 0.0001	significant
Pure Error	0.20	1566	0.0001302			
Cor Total	1.73	1619				
Std.Dev.	0.012	R-Squared	0.8749			
Mean	0.61	Adj. R-Squared	0.8742			
C.V.	1.90	Pred. R-Squared	0.8733			
PRESS	0.22	Adeq. Precision	101.647			

En iyileme sonuçları toplam fayda değeri için hedeflerin belirlenmesi ve en uygun şartların sağlanmasıyla elde edilir. Görünürlük tabanlı görevde bu uygun şartlar için, özellikle, popülasyon büyüklüğü 50 olarak belirlenirken çaprazlama operatörü olarak katkı tabanlı çaprazlama, mutasyon işleminde kullanılan sensör seçiminde ise en düşük görünürlük değerine sahip sensör seçilmiştir. Algoritmaya özgü parametreler içinde bunlara karşı sadece mutasyon operatörünün toplam fayda değerine önemli bir katkısı yoktur.

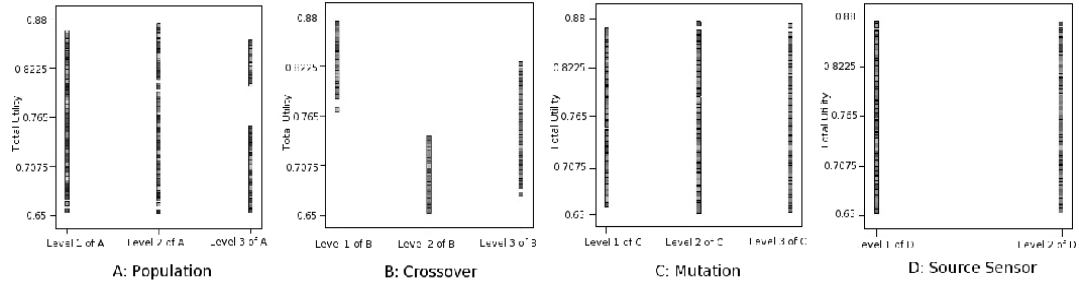
4.1.3 Probleme Özgü Parametreler ile Yapılan Deneyler

İkinci grup deneylerde birinci grup deneylerde yer almayan algoritmaya özgü parametreler (başlangıç popülasyonu seçimindeki strateji, yatay ve dikey yönelme açılarının artış miktarı) ile probleme bağlı parametreler (arazinin yapısı, toplam fayda değerinin hesaplanmasında kullanılan kriterlerin ağırlık değerleri) kullanılmıştır. İkinci grup deneylerde kullanılan bu parametrelerin değerleri Tablo 4'de listelenmiştir. Ayrıca ilk grup deneylerde elde edilen algoritmaya bağlı parametrelerin en iyi değerleri bu gruptaki deneylerde de kullanılmıştır.

Tablo 4: İkinci grup deneylerde kullanılan parametre alternatifleri

Başlangıç Popülasyonu Seçim Stratejisi	Yatay Yönelme Açısı Artış Miktarı	Dikey Yönelme Açısı Artış Miktarı	Fayda Ağırlık Oranları	Arazi Yapısı
(%75 Sezgisel-%25 Rasgele)	25	3	(0.6, 0.3, 0.1)	Terrain 1 (düzlük)
(%50 Sezgisel-%50 Rasgele)	50	5	(0.3, 0.6, 0.1)	Terrain 2 (düzlük)
(%100 Rasgele)	75	7	(0.25, 0.25, 0.5)	Terrain 3 (engebeli)
		10		Terrain 4 (engebeli)

Varyans analizi testlerine göre, Tablo 4'deki ilk üç parametrenin toplam fayda değerine önemli bir katkısı olmadığı görülmektedir. Arazinin engebelilik derecesi ve fayda kriterlerine verilen farklı ağırlıklarla oluşturulan görev türleri ise toplam fayda değerine önemli katkı sağlamaktadır. Böylece, ilk grup deneyler diğer alternatif görevler (gizlilik ağırlıklı ve maliyet ağırlıklı) için genişletilebilmiştir.



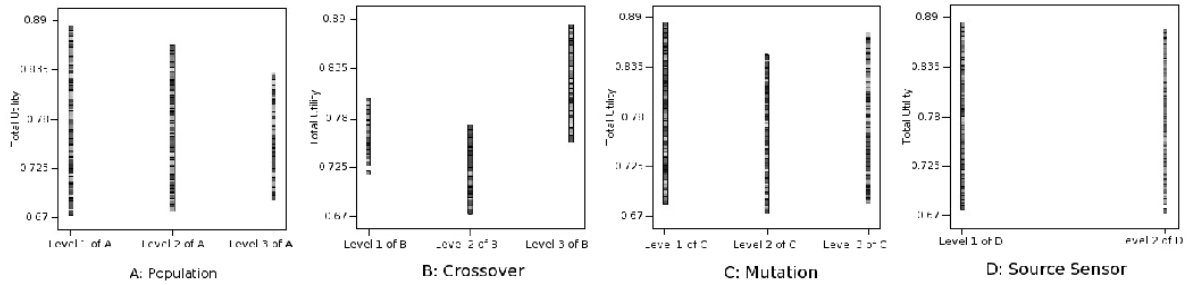
Şekil 27: Gizlilik ağırlıklı misyonda dört etkenin toplam fayda değerine etkisi

Gizlilik ağırlıklı görevde algoritmaya bağlı parametrelerin çözüme olan etkisi Şekil 27'de ve yine aynı görev için varyans analizi sonuçları Tablo 5'de verilmiştir. Tablodan da görüleceği üzere, popülasyon büyüklüğü, çaprazlama ve mutasyon türü, ve bu parametreler arasındaki etkileşim toplam fayda değerine önemli katkılar sağlamaktadır. Bu ilişki tablodaki “R-Squared” değerleri ile de doğrulanmıştır. Sayısal en iyilemeye göre gizlilik ağırlıklı görev için Tablo 2'deki parametrelerin değerleri görünürlük ağırlıklı görevde kullanılan değerlerle aynı değerlere ayarlanır.

Tablo 5: Gizlilik ağırlıklı misyonda toplam fayda değerinin varyans analizi

Source	Sum of Squares	df	Mean Square	F Value	p-Value Prob>F	
Model	5.72	26	0.22	1154.28	< 0.0001	significant
A-Population	0.28	2	0.14	721.86	< 0.0001	
B-Crossover	5.22	2	2.61	13697.54	< 0.0001	
C-Mutation	0.013	2	0.006627	34.76	< 0.0001	
AB	0.19	4	0.047	245.27	< 0.0001	
AC	0.007771	4	0.001943	10.19	< 0.0001	
BC	0.007448	4	0.001862	9.77	< 0.0001	
ABC	0.008026	8	0.001003	5.26	< 0.0001	
Residual	0.30	1593	0.0001907			
Lack of Fit	0.007416	27	0.0002747	1.45	0.0631	not significant
Pure Error	0.30	1566	0.0001892			
Cor Total	6.03	1619				
Std.Dev.	0.014		R-Squared	0.9496		
Mean	0.76		Adj. R-Squared	0.9488		
C.V.	1.82		Pred. R-Squared	0.9479		
PRESS	0.31		Adeq. Precision	91.579		

Maliyet tabanlı görevde algoritmaya bağlı parametrelerin çözüme olan etkisi Şekil 28'de ve yine aynı görev için varyans analizi sonuçları Tablo 6'da verilmiştir. Algoritmaya bağlı parametrelerin tümü ve kendi aralarındaki etkileşim toplam fayda değerine önemli katkılar sağlamıştır. Bu ilişki tablodaki "R-Squared" değerleri ile de doğrulanmıştır. Sayısal en iyilemeye göre popülasyon büyüklüğü 30 olarak belirlenirken çaprazlama operatörü olarak kes-birleştir çaprazlama ve mutasyon operatörü olarak konum tabanlı mutasyon operatörü seçilmiştir.

**Şekil 28:** Maliyet ağırlıklı misyonda dört etkenin toplam fayda değerine etkisi

Tablo 6: Maliyet ağırlıklı misyonda toplam fayda değerinin varyans analizi

Source	Sum of Squares	df	Mean Square	F Value	p-Value Prob>F	
Model	2.99	35	0.085	478.55	<0.0001	significant
A-Population	0.044	2	0.022	122.60	< 0.0001	
B-Crossover	2.60	2	1.30	7276.74	< 0.0001	
C-Mutation	0.062	2	0.031	174.90	< 0.0001	
D-Source Sensor	0.000768	1	0.000767	4.30	0.0382	
AB	0.20	4	0.051	283.21	< 0.0001	
AC	0.016	4	0.004104	23.02	< 0.0001	
AD	0.000253	2	0.000127	0.71	0.4915	
BC	0.038	4	0.009589	53.77	< 0.0001	
BD	0.000796	2	0.000398	2.23	0.1078	
ABC	0.022	8	0.002771	15.54	< 0.0001	
ABD	0.004675	4	0.001169	6.55	< 0.0001	
Residual	0.28	1584	0.000178			
Lack of Fit	0.004112	18	0.000229	1.29	0.1874	not significant
Pure Error	0.28	1566	0.000178			
Cor Total	3.27	1619				
Std.Dev.	0.013		R-Squared	0.9136		
Mean	0.77		Adj. R-Squared	0.9117		
C.V.	1.74		Pred. R-Squared	0.9096		
PRESS	0.30		Adeq. Precision	78.930		

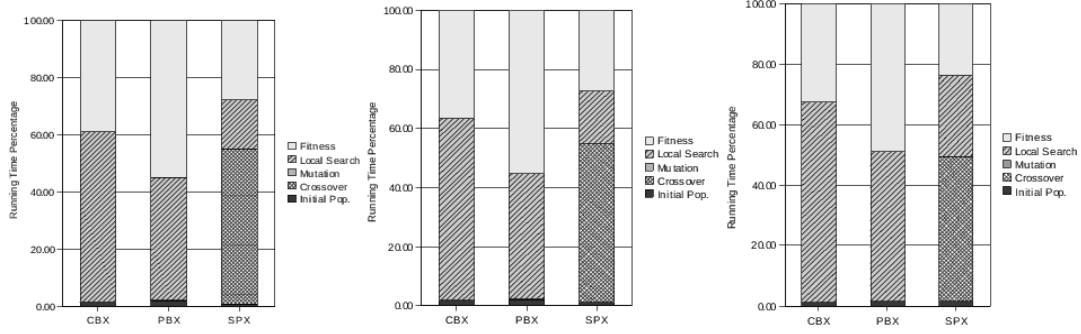
4.2 Performans Değerlendirmesi ve Tartışma

Bu bölümde çeşitli unsurların çözümün kalitesine olan etkisi değerlendirilmiş ve ilgili tartışmalar sunulmuştur.

4.2.1 Algoritmanın Koşum Süresi Analizi:

Karma Evrimsel Algoritmanın beş farklı bölümünün (başlangıç popülasyonu, çaprazlama ve mutasyon operatörleri, yerel arama ve uygunluk hesaplaması) çalışma sürelerinin yüzdesel oranları, üç farklı görev için farklı çaprazlama operatörleri dikkate alınarak Şekil 29'da gösterilmiştir. Katkı tabanlı ve komşuluk tabanlı çaprazlama operatörleriyle mutasyon operatörünün çalışma süresinin yüzdeleri çok küçük (< 0.01) olduğu için çizgede gösterilmemektedir. Tüm sonuçlara genel olarak baktığımızda; yerel arama metodu ve uygunluk hesaplaması, katkı tabanlı ve komşuluk tabanlı çözümlerin toplam çalışma sürelerinin %98'ini oluşturmaktadır.

Buna karşın kes-birleştir çaprazlama operatörü, iki yeni bireyin uygunluk hesaplamasından dolayı toplam çalışma süresini %33 oranında artırmıştır. Yerel arama işleminin toplam çalışma süresindeki oranı, bireylerdeki ortalama sensör sayısının her çaprazlama operatörü için farklılık göstermesinden dolayı farklı değerler vermektedir. Kes-birleştir çaprazlama operatörünün kullanıldığı çözümlerdeki birey sayısının diğer çaprazlama operatörlerinin kullanıldığı çözümlerdeki birey sayısından daha az olduğu görülmektedir. Bunlara ek olarak, katkı tabanlı ve komşuluk tabanlı çaprazlama operatörlerinin kullanıldığı çözümlerdeki popülasyon büyüklüğü kes-birleştir çaprazlama çözümüne göre daha büyük olduğu için, başlangıç popülasyon üretiminin çalışma süresi kes-birleştir çaprazlama operatörlerinin kullanıldığı çözümlerde diğerlerine göre daha küçük orana sahiptir.

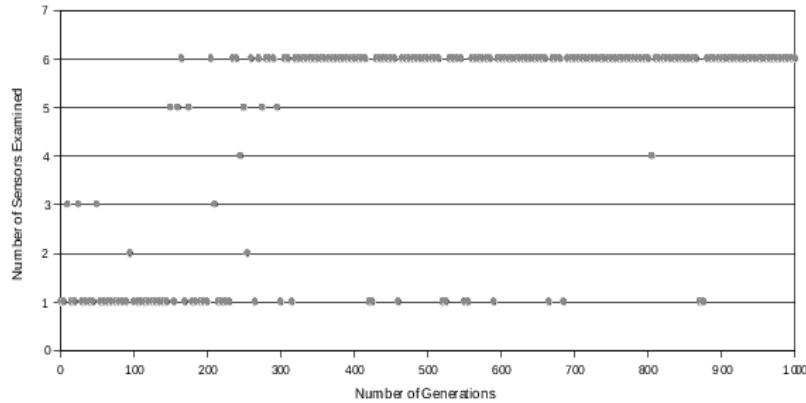


Şekil 29: Karma evrimsel algoritmanın çeşitli bölümlerinin görünürlük (a), gizlilik (b) ve maliyet (c) ağırlıklı misyonlarda koşum süresi yüzdeleri

4.2.2 Yerel Arama Yönteminde Sonlandırma Koşulunun Etkisi:

Başlangıç popülasyonunda sensörlerin açılal özellikleri rasgele belirlendiği için, sensörler belirlenen arazi üzerindeki konumlarında en ideal açılal değerlere sahip olmayabilirler. Bu nedenle başlangıç popülasyonundaki birçok sensörün görünürlük fayda değeri 0.01'in altındadır. Şekil 30'de detaylı şekilde açıklanan yerel arama yöntemi, sensörlerin yatay ve dikey yönelme açılarını değiştirerek görünürlük fayda değerini yükseltmeyi hedeflemektedir.

Deneylerde yerel arama yöntemi iki farklı şekilde sonlanmaktadır: sensörlerin görünürlük fayda değerlerinde toplam %20' lik ilerleme olması veya görünürlük fayda değeri iyileştirilmeye çalışılan sensör sayısının çözümdeki toplam sensör sayısının %10' una ulaşması. "Teşhis" arama modunda görünürlük ağırlıklı bir deneyde her iterasyondaki yerel arama yönteminin sonlanma sebebi Şekil 36'da gösterilmiştir.



Şekil 30: Görünürlük ağırlıklı misyonda yerel arama yönteminin sonlanma koşulu

Şekil 30'da görüleceği üzere ilk 200 iterasyon boyunca çözümlerdeki en kötü sensörün görünürlük fayda değerindeki iyileştirme oranı tanımlanan %20' lik koşulu sağlayıp yerel arama yöntemi sonlanmaktadır. 200 ve 300' üncü iterasyon arasında açılal özellikleri değiştirilip test edilen sensörlerin sayısı iterasyon sayısı ilerlerken hafifçe artmaktadır. Teşhis arama modunda bireylerdeki ortalama sensör sayısı 6 olduğu için 300' üncü iterasyondan sonra toplam sensör sayısının %10' u olan 6 sensörde yapılan testlerden sonra yerel arama yöntemi %10' luk sensör limitine ulaştığı için

sonlanır. Sonuç olarak, ilk iterasyonlarda yerel arama yöntemi iyileştirme oranı üst limitini sağladığı için sonraki iterasyonlarda ise test edilen sensör sayısı üst limitini sağladığı için sonlanmıştır. Benzer gözlemler diğer arama modları için de geçerlidir.

4.2.3 Yerel Aramada Alternatif Yöntemlerin Çözüm Etkisi:

Önceki kısımda belirtildiği gibi, yerel arama metodu sensörlerin açılal özelliklerini değiştirerek çözümün kalitesini arttırmayı hedeflemektedir. Tablo 7'de yerel arama metodu için üç alternatif yöntemin çözüme katkısı görülmektedir. “Heading → Tilt” olarak adlandırılan ilk alternatif yöntemde, sensörün yatay yönelme açısı sensörün görünürlük değerinde ilk ilerleme sağlanana kadar değiştirilir; daha sonra dikey yönelme açısı tüm alternatif açı değerleriyle test edilir ve en yüksek görünürlük fayda değerine sahip açı değeri bulunur. “Tilt → Heading” olarak adlandırılan ikinci yöntemde, ilk yöntemdeki yatay ve dikey yönelme açı testlerinin yerleri değiştirilmiştir. Üçüncü yöntemde ise, yerel arama yöntemi kaldırılıp toplam iterasyon sayısı 1000' den 2000' e çıkarılmıştır.

Tablo 7: Yerel arama metodunda alternatif yöntemlerin performans değerlendirmesi

Yerel Aramadaki Güncellemeler	Misyon Tipi	Sensör Sayısı	Görünürlük	Gizlilik	Maliyet	Toplam Fayda	Koşum Süresi (dk.)
Heading→Tilt	M1	64.2	0.4953	0.7549	0.4476	0.6621	26
	M2	63.8	0.4192	0.9183	0.5491	0.8418	25
	M3	12.6	0.1186	0.9683	0.9117	0.8618	9
Tilt→Heading	M1	64.2	0.4992	0.7479	0.4425	0.6614	18
	M2	62.8	0.4263	0.9112	0.5516	0.8408	17
	M3	12.0	0.1191	0.9660	0.9114	0.8610	8
Yerel Arama Hariç	M1	72.4	0.5041	0.7645	0.3770	0.6615	24
	M2	67.0	0.4176	0.9224	0.5049	0.8363	21
	M3	7.1	0.1123	0.9949	0.9447	0.8901	8

Tablodaki ikinci sütun görev türünü belirtmektedir. M1, M2, M3 sırasıyla görünürlük, gizlilik ve maliyet ağırlıktaki görevleri simgeler. Her üç görev için test edilen yerel aramadaki üç alternatif yöntem birbirine yakın değerler vermektedir. Tüm yatay yönelme açısı alternatiflerinin sayısı dikey yönelme açısı alternatiflerinin sayısından daha az olduğu için, “Heading → Tilt” yöntemi “Tilt → Heading” yöntemine göre daha uzun sürede sonlanır. Üçüncü yöntemde ise yerel arama metodu çıkarılmasına karşın iterasyon sayısı iki katına çıkarıldığı için toplam fayda değeri ilk iki yöntemle yakın değerlere sahiptir. Çalışma süresinin düşük olmasından dolayı kalan diğer deneylerimizde “Tilt → Heading” yöntemi yerel arama metodu olarak kullanılmıştır.

4.2.4 Katkı Tabanlı Çaprazlamada Sensör Sayısı Çarpanının Etkisi:

Çözümdeki sensör sayısının çözümün fayda değerlerindeki (görünürlük, gizlilik, maliyet) etkisini ölçmek için katkı tabanlı çaprazlama yönteminde kullanılan sensör sayısı çarpanı (ϵ) görünürlük (M1) ve gizlilik (M2) ağırlıklı görevlerde farklı değerlerle test edilmiştir (bkz.Tablo 8). Sensör sayısındaki artış görünürlük fayda değerini doğrudan olumlu etkilerken maliyet ve gizlilik fayda değerlerindeki düşüşten dolayı toplam fayda değerini olumsuz etkilemektedir. Toplam fayda değerinin hesaplanması birbiriyle çelişen etkenlere bağlı olduğu için, görünürlük fayda değeri yapılan tüm deneylerde 0.60 değerini geçememiştir. Toplam fayda değerindeki fayda etkenlerinin kendi aralarındaki çelişen ilişki Tablo 8'de de görülmektedir.

Tablo 8: Algoritmada çeşitli sensör sayısı çarpanlarının performans değerlendirmesi

Sensör Sayısı Çarpanı	Misyon Tipi	Sensör Sayısı	Görünürlük	Gizlilik	Maliyet	Toplam Fayda
$\varepsilon = 0.6$	M1	102.2	0.5350	0.7234	0.2562	0.6488
	M2	102.2	0.4837	0.8417	0.3392	0.7819
$\varepsilon = 0.7$	M1	117.5	0.5441	0.7124	0.1751	0.6381
	M2	124.1	0.4903	0.8231	0.2001	0.7478
$\varepsilon = 0.8$	M1	119.7	0.5391	0.7176	0.1549	0.6332
	M2	128.3	0.4943	0.8171	0.1714	0.7406

4.2.5 Çözüme Yeni Sensör Eklemenin Çözümün Kalitesine Etkisi:

İlk sensör ekleme yöntemi, popülasyonun %60'ına her 100 iterasyonda bir rasgele 3'er sensör eklemeyi içermektedir. Eklenecek sensörlerin tüm özellikleri limitler dahilinde rasgele belirlenmiştir. Bu yöntemin kullanıldığı her görev için toplam fayda değerleri Tablo 9'da verilmiştir. Tablo 7'deki testlerle aynı parametrelere (arazi şekli, görüntüleme modu ve diğer algoritmaya bağlı parametreler) sahip bu yöntemin sonuçları Tablo 7'deki sonuçlarla çok benzerlik göstermektedir.

Bu benzerliğe karşın görünürlük ve gizlilik baskın çözümlerde ortalama sensör sayısı Tablo 8'deki değerlerine göre %15 daha yüksek iken, maliyet baskın çözümlerde ortalama sensör sayısı 3 kat artmaktadır. Sensör sayısındaki artış üç görevdeki görünürlük fayda değerini de yükseltmektedir. Görünürlük baskın görevde maliyet fayda değeri azalma oranı görünürlüğün artma oranından yüksek olduğundan dolayı toplam fayda değerinde önemli bir değişiklik olmamıştır. Gizlilik ve maliyet baskın çözümlerde maliyet fayda değeri azalma oranı ile görünürlüğün artma oranı arasındaki fark görünürlük baskın çözümdeki değerlere göre çok daha yüksek olduğu için toplam fayda değerleri azalmıştır.

Tablo 9: Karma evrimsel algoritmada yeni sensör eklemenin çözüme katkısının karşılaştırılması

Güncelleme Yöntemleri	Misyon Tipi	Sensör Sayısı	Görünürlük	Gizlilik	Maliyet	Toplam Fayda
Sensör Ekleme	M1	77.0	0.5253	0.7506	0.3606	0.6694
	M2	74.6	0.4489	0.8999	0.4754	0.8304
	M3	38.9	0.2583	0.8570	0.7272	0.7626
Sensör Ekleme & Silme	M1	57.6	0.5016	0.7718	0.4556	0.6751
	M2	55.8	0.4218	0.9291	0.5748	0.8554
	M3	23.0	0.2013	0.9181	0.8281	0.8261

İkinci yöntemde, ilk yöntem ile bireylerdeki en kötü sensörlerin silinmesi yöntemi biraraya getirilmiştir. Sensör silme işlemi, görünürlük fayda değeri çözümdeki sensörlerin ortalama görünürlük fayda değerinin üçte birinden düşük olan sensörlerin çözümde çıkartılmasıyla gerçekleştirilir. Bu işlem her 100 iterasyonda bir tüm popülasyona uygulanır. Sensör silme işleminden sonra ilk yöntemde belirtilen sensör ekleme işlemi uygulanır. Deney sonuçları incelendiğinde görünürlük ve gizlilik ağırlıklı çözümlerde hem görünürlük hem de maliyet fayda değerlerindeki artıştan dolayı toplam fayda değerlerinde küçük miktarda bir artış gözlemlenmektedir. Maliyet ağırlıklı çözümlerde ise önceki değerlere göre bir azalma görülmektedir.

4.2.6 Sensör Görüntüleme Modunun Çözümün Kalitesine Etkisi:

Bu bölümde sensörlerin tespit, teşhis ve tanıma modunda olduğu, engebelilik derecesi az ve çok olan araziler üzerinde Karma Evrimsel Algoritmanın performansı değerlendirilecektir.

Sensörler kısıtlı görme mesafelerine sahip oldukları için görüş alanı içerisindeki herhangi bir yüksek eğimli ortam, sensörün bu bölgenin arkasını görmesini engelleyeceğinden sensörün görüş alanı daralacaktır. Bu nedenle, benzer özelliklere (sensör sayısı, maliyeti) sahip çözümlerde engebelilik derecesi yüksek (çok sayıda eğimli alan) arazilerin görünürlük fayda değeri, engebelilik derecesi düşük arazilere göre daha az olmuştur (bkz.Tablo 10-Tablo 11). Bu tablolarda görünürlük (M1) ve gizlilik (M2) ağırlıklı görevler dikkate alınmıştır.

Tablo 10: Düzlük bir arazide farklı sensör görüntüleme modlarının performans değerlendirmesi

Sensör Modu	Misyon Tipi	Algoritma	Sensör Sayısı	Görünürlük	Gizlilik	Maliyet	Toplam Fayda
Tespit	M1	RS	13.9	0.3578	0.6536	0.5161	0.5267
		HEA	14.7	0.5716	0.8000	0.4607	0.7419
	M2	RS	9.7	0.2337	0.7997	0.6785	0.6894
		HEA	14.7	0.5200	0.9615	0.5415	0.9216
Teşhis	M1	RS	80.7	0.4283	0.6108	0.4389	0.5524
		HEA	59.7	0.5561	0.7433	0.4840	0.7061
	M2	RS	54.9	0.3036	0.7020	0.6241	0.6465
		HEA	62.3	0.4743	0.9162	0.5811	0.8731
Tanıma	M1	RS	227.7	0.3053	0.7411	0.4673	0.5138
		HEA	178.2	0.4557	0.8339	0.3482	0.6474
	M2	RS	162.7	0.2289	0.7764	0.6117	0.6620
		HEA	170.3	0.3957	0.9439	0.5061	0.8409

Tablo 10 ve Tablo 11'de, farklı sensör görüntüleme modları ve farklı görev türleri için Karma Evrimsel Algoritma (HEA) ile rasgele yapılan bir aramanın (RS) performans karşılaştırılması yapılmıştır. Rasgele arama yöntemi, Karma Evrimsel Algoritma'da 50 bireyden oluşan bir popülasyonun başlangıç değerlerinden en iyi bireyin seçilmesi işlemidir. Her iki türdeki arazi içinde HEA yöntemi RS yöntemindeki değerleri çok yüksek oranda iyileştirmiştir. Bunlara ek olarak, 3-boyutlu bir araziyi görüntülemek için gerekli sensör sayısı her sensör görüntüleme modun için farklılık göstermektedir. "Tespit" modunda sensörlerin görüş mesafesi, yatay ve dikey görme açıları gibi davranışsal özellikleri diğer görüntüleme modlarına göre daha iyi olduğu için araziyi görüntülemek için daha az sayıda sensöre ihtiyaç duyulmaktadır.

Tablo 11: Engebeli bir arazide farklı sensör görüntüleme modlarının performans değerlendirmesi

Sensör Modu	Misyon Tipi	Algoritma	Sensör Sayısı	Görünürlük	Gizlilik	Maliyet	Toplam Fayda
Tespit	M1	RS HEA	11.3 16.3	0.2062 0.4710	0.7773 0.8022	0.5784 0.3980	0.4645 0.6545
	M2	RS HEA	8.0 16.3	0.1221 0.4239	0.8193 0.9555	0.6863 0.4670	0.6516 0.8561
Teşhis	M1	RS HEA	77.3 63.0	0.3321 0.4946	0.6697 0.7574	0.3990 0.4548	0.4975 0.6638
	M2	RS HEA	44.5 61.3	0.2075 0.4215	0.7393 0.9084	0.6483 0.5637	0.6321 0.8385
Tanıma	M1	RS HEA	214.3 180.6	0.2509 0.4022	0.7697 0.8430	0.4347 0.3658	0.4777 0.6121
	M2	RS HEA	141.9 166.5	0.1744 0.3406	0.8025 0.9415	0.6260 0.5220	0.6564 0.8148

Bu iki yöntemin, görüntülenen poligon yüzdesi ve düşman sensörleri tarafından gözlenen sensör yüzdesi cinsinden performans değerlendirmesi düzlük arazi için Tablo 12'de, engebeli arazi için Tablo 13'de listelenmiştir. Gözlenen sensör yüzdesi hesaplaması, düşman sensörlerin farklı konumlarda ve farklı açılarda yerleştirildiği beş farklı senaryonun ortalama gizlilik değerleriyle belirlenmiştir.

Tablolarda görüntülenen poligon yüzdeleri farklı görünürlük limitlerine (0.1, 0.5, 0.75) göre ayrılmıştır. Örneğin, " $V(S,P,p_i) \geq 0.1$ " sütunu en az %10 görünür olan poligonların yüzdesini vermektedir. Sonuçlardan da görüleceği üzere, Karma Evrimsel Algoritma (HEA) tüm sensör modlarında ve her iki arazi türünde de rasgele aramaya (RS) göre oldukça üstün çözümler vermektedir. Örneğin, görünürlük baskın bir görev ve tespit sensör görüntüleme modunda düzlük bir arazi için görüntülenen sensör yüzdesinde %80 ilerleme olurken engebeli bir arazi üzerinde %132 ilerleme olmuştur.

Tablo 12: Düzlük bir arazide farklı sensör görüntüleme modlarının performans değerlendirmesi

Sensör Modu	Misyon Tipi	Algoritma	Görüntülenen Poligonlar (%) eğer $V(S,P,p_i)$			Gözetlenen Sensörler (%) eğer $V(S,P, p_i)$		
			≥ 0.1	≥ 0.5	≥ 0.75	≥ 0.1	≥ 0.5	≥ 0.75
Tespit	M1	RS HEA	0.58 0.88	0.44 0.73	0.15 0.27	0.71 0.49	0.45 0.21	0.19 0.03
	M2	RS HEA	0.38 0.83	0.29 0.65	0.09 0.21	0.60 0.20	0.39 0.01	0.15 0.00
Teşhis	M1	RS HEA	0.71 0.87	0.52 0.70	0.17 0.24	0.80 0.63	0.36 0.17	0.07 0.01
	M2	RS HEA	0.53 0.79	0.36 0.58	0.11 0.17	0.73 0.34	0.31 0.00	0.05 0.00
Tanıma	M1	RS HEA	0.59 0.77	0.34 0.56	0.08 0.16	0.67 0.53	0.14 0.04	0.02 0.00
	M2	RS HEA	0.45 0.71	0.24 0.47	0.05 0.11	0.64 0.25	0.13 0.00	0.02 0.00

Tablo 12 ve Tablo 13'de düşman sensörler tarafından gözlenen sensör yüzdesi de farklı görünürlük limitlerine (0.1, 0.5, 0.75) göre ayrılmıştır. Örneğin, tablolardaki " $V_E(S,P,P_{si}) \geq 0.1$ " sütunu düşman sensörler tarafından gözlenen sensörlerden görünme değeri en az %10 olanların yüzdesini göstermektedir. Bu sütundaki yüzde oranının düşük olması düşman sensörler tarafından görünme oranının da düşük olduğunu ifade etmektedir. Önceki sonuçlarda olduğu gibi gözlenen sensör yüzdesi bakımından Karma Evrimsel Algoritma rasgele aramaya göre oldukça üstün sonuçlar vermektedir. Tablolardan görüleceği üzere düşman sensörler tarafından çözümdeki herhangi bir sensörün görünme değerinin %75'den büyük olmaması, çözümdeki sensör arazi üzerine gizlilik kriterine göre oldukça elverişli şekilde yerleştirildiğini göstermektedir. Ayrıca her iki yöntemi incelediğimizde, hem gözlenen sensörlerin yüzdeleri arasında oransal fark, hem de görüntülen poligonların yüzdeleri arasındaki oransal fark görünme alt limiti yükseldikçe önemli oranda artmaktadır.

Tablo 13: Engelibeli arazide farklı sensör görüntüleme modlarının performans değerlendirmesi

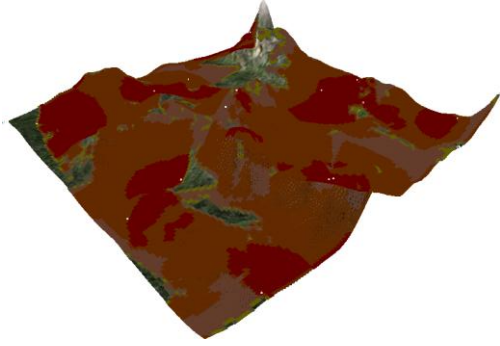
Sensör Modu	Misyona Tipi	Algoritma	Görüntülenen Poligonlar (%) eğer $V(S,P,p_i)$			Gözlenen Sensörler (%) eğer $V(S,P, p_i)$		
			≥ 0.1	≥ 0.5	≥ 0.75	≥ 0.1	≥ 0.5	≥ 0.75
Tespit	M1	RS	0.34	0.25	0.09	0.64	0.36	0.13
		HEA	0.76	0.58	0.20	0.49	0.17	0.05
	M2	RS	0.21	0.14	0.05	0.63	0.31	0.12
		HEA	0.70	0.52	0.16	0.24	0.00	0.00
Teşhis	M1	RS	0.58	0.38	0.13	0.77	0.30	0.05
		HEA	0.81	0.61	0.19	0.61	0.14	0.01
	M2	RS	0.38	0.24	0.07	0.70	0.24	0.03
		HEA	0.73	0.50	0.14	0.38	0.00	0.00
Tanıma	M1	RS	0.50	0.26	0.06	0.65	0.12	0.02
		HEA	0.72	0.48	0.13	0.52	0.03	0.00
	M2	RS	0.36	0.18	0.04	0.63	0.10	0.02
		HEA	0.63	0.39	0.09	0.25	0.00	0.00



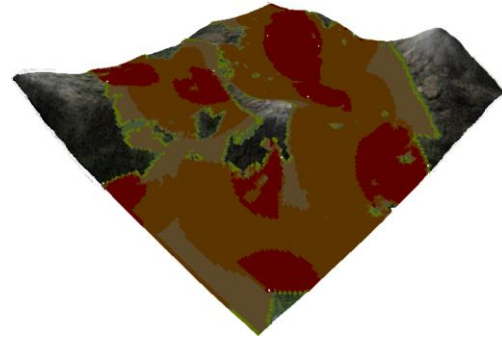
(a)



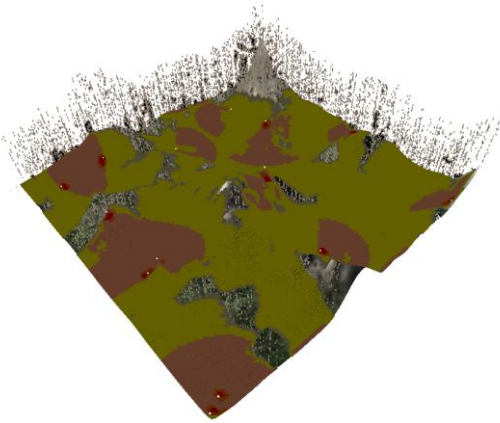
(b)



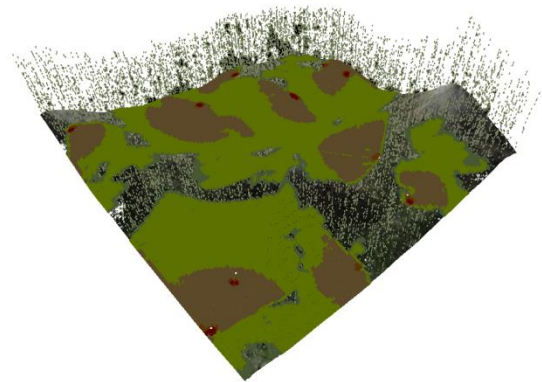
(c)



(d)



(e)

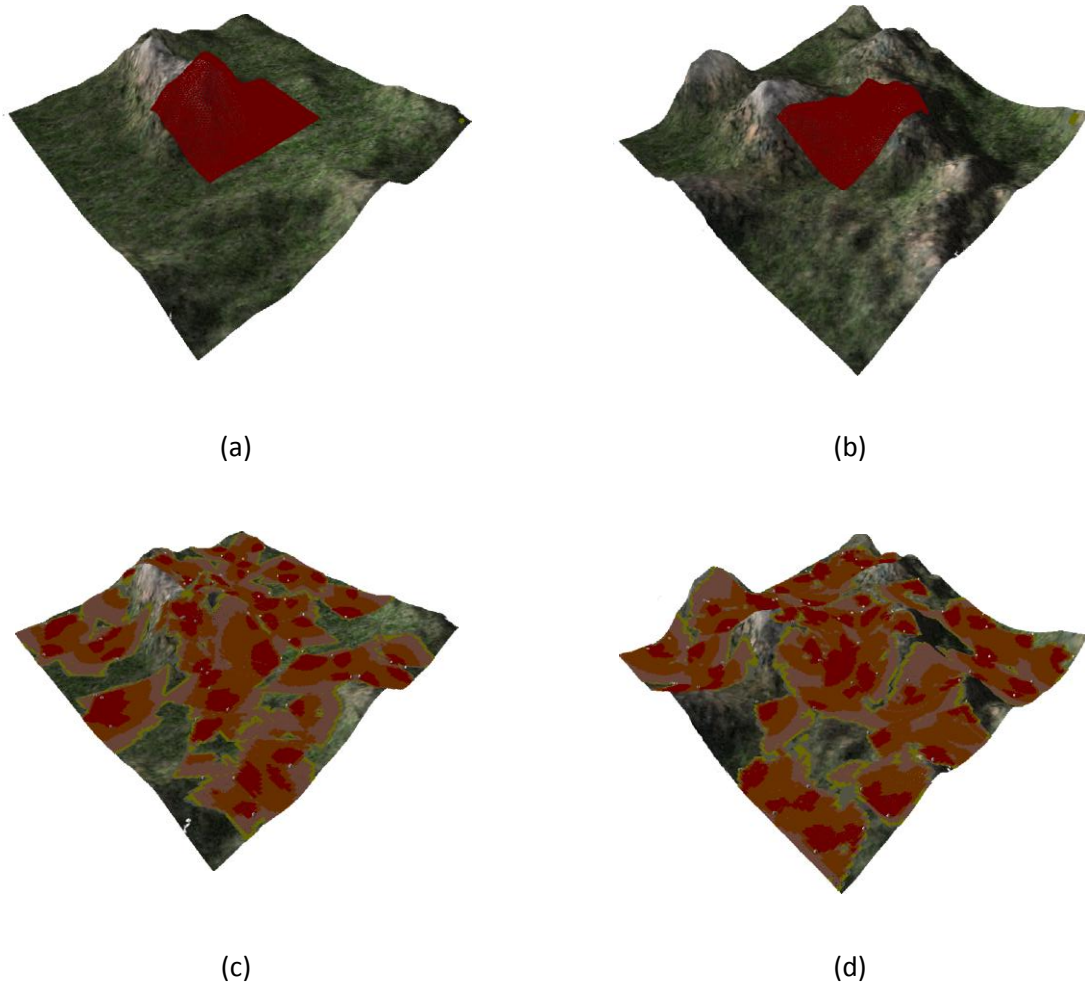


(f)

Şekil 31: (a) Engebeli ve (b) Düzlük araziler için: (c,d) engel olmadan elde edilen kapsama alanları, (e,f) yağışlı hava koşullarında elde edilen kapsama alanları.

Yukarıdaki şekillerde engebeli ve düzlük arazi yapıları ve onlara ait bulunan kapsama alanları verilmiştir (Şekil 31). Kapsama haritaları, “tesbit” sensör modunda w_{vis} , w_{ste} , w_{cost} değerleri sırasıyla 0.60, 0.30, 0.10 olan görünürlük ağırlıklı bir misyon kullanılarak yapılan örnek koşumların çıktılarıdır. Örnek koşumda kullanılan diğer parametreler algoritmaya özgü ve probleme özgü parametrelerle ilgili yapılan deneylerin sonuçlarına göre belirlenmiştir. Engebeli arazi yapısına ait kapsama sonuçlarında 14 sensörlük bir çözüm 0.7199 toplam fayda değerine sahiptir. Düzlük arazi yapısına ait kapsama sonuçlarında ise 9 sensörlük bir çözüm 0.7571 toplam fayda değerine sahiptir. Aynı parametreler ve arazi kullanılıp, yağmur-tabanlı hava engel koşulu göz önüne alınarak yapılan testlerde, sensör sayısında önemli bir değişiklik olmazken arazinin görünürlük değeri ile düşman sensörlerin görüntüleme değerleri büyük oranda düşmektedir (Şekil 31.e-f). Dolayısıyla elde edilen çözümlerin gizlilik değerleri yükselmektedir.

Yukarıdaki örnek koşumlardaki parametreler kullanılarak “teşhis” sensör modunda önem haritasının da dikkate alındığı yeni örnek koşumlar, aşağıdaki farklı birer engebeli ve düzlük arazi yapısında gerçekleştirilmiştir (Şekil 32). Bu örnek koşumlarda önem haritasının işaretlendiği bölgedeki sensörlerin yoğunluğu ve görüntülenme oranı aşağıdaki şekillerde kolayca görülmektedir.



Şekil 32: Önem haritasının yer aldığı (a) Engebeli ve (b) Düzlük araziler için elde edilen kapsama alanları (c,d)

4.2.7 Genel Güzergah Planlamaya Özgü Deneysel Çalışmalar

Bu bölüm, farklı karakteristiğe sahip araziler ve farklı askeri misyonlar için geliştirilen algorithmada genel güzergah planlama aşamasının performansını ölçmeye yönelik deneyler içermektedir.

Genel güzergah planlama (GGP) aşamasında kullanılan parametreler çalışmamızda sabit ve değişken olmak üzere iki gruba ayrılmıştır. Sabit değer alan parametreler Tablo 14’de, değişken değer alan parametreler ise Tablo 15’de verilmiştir.

Tablo 14: Genel güzergah planlama algoritmasındaki sabit parametreler

	Sabit Parametreler	Değerler
1	Sensörün bir bölgeye atanması	% 80 – merkeze % 20 – herhangi bir nokta
2	Kontrol noktası seçiminde diğer kontrol noktaları ile arasındaki minimum mesafe farkı	250
3	Kontrol noktası seçiminde sınıra minimum yakınlık mesafesi	90
4	Kontrol noktası seçiminde maksimum komşu nokta sayısı	5
5	Kontrol noktasının güzergaha doğrudan atanması için nokta ile güzergah arasındaki maksimum mesafe	40
6	Sensör sayısı	3

Tablo 15: Genel güzergah planlama algoritmasındaki değişken parametreler ve olası değerleri

Nokta Sayısı	Komşuluk Yarıçapı	VPD Sabiti	Kontrol Noktası Sayısı	Poligon Eğim Üst Limiti	Misyon Ağırlıkları Görünürlük (w_v)	Maliyet (w_d)	Düzgünlük (w_s)
81	200	2.0	7	35°	0.70	0.65	0.65
100	250	3.0	10	45°	0.50	0.35	0.35
121		4.0			0.30		

Tablo 15’de verilen değişken parametrelerin en uygun değerlerini belirlemek için olası değerler arasında bir ön deney gerçekleştirilmiştir. Bu deneyde 6 farklı misyon için toplam 432 farklı kombinasyonda parametreler test edilmiştir. Deneylerde algoritmanın farklı arazi karakteristiklerindeki performansı da ölçülmüştür. Engeli bir arazi yapısında yapılan deneylerle belirlenen farklı misyonlar için parametre değerleri Tablo 16’da verilirken, Tablo 17’de düzlük arazi yapısındaki sonuçlar yer almaktadır.

Tablo 16: Engelibeli arazide farklı askeri misyonlar için belirlenen genel güzergah algoritması parametreleri

Misyona Ağırlıkları			Poligon Eğim Üst Limiti	Nokta Sayısı	Komşuluk Yarıçapı	VPD Sabit	Kontrol Noktası Sayısı
Görünürlük (w_v)	Maliyet (w_d)	Düzgünlük (w_s)					
0.70	0.65	0.35	45°	100	250	2.0	10
			35°	121	250	4.0	7
0.70	0.35	0.65	45°	81	250	2.0	10
			35°	100	250	4.0	10
0.50	0.65	0.35	45°	100	250	3.0	10
			35°	121	250	4.0	7
0.50	0.35	0.65	45°	81	250	2.0	10
			35°	121	250	4.0	7
0.30	0.65	0.35	45°	100	250	3.0	10
			35°	121	250	4.0	7
0.30	0.35	0.65	45°	100	250	4.0	7
			35°	121	250	4.0	7

Tablo 17: Düzgünlük arazide yapısında farklı askeri misyonlar için belirlenen genel güzergah algoritması parametreleri

Misyona Ağırlıkları			Poligon Eğim Üst Limiti	Nokta Sayısı	Komşuluk Yarıçapı	VPD Sabit	Kontrol Noktası Sayısı
Görünürlük (w_v)	Maliyet (w_d)	Düzgünlük (w_s)					
0.70	0.65	0.35	45°	100	200	2.0	10
			35°	121	250	2.0	10
0.70	0.35	0.65	45°	81	200	3.0	10
			35°	121	250	2.0	10
0.50	0.65	0.35	45°	100	200	2.0	10
			35°	121	250	2.0	10
0.50	0.35	0.65	45°	81	200	3.0	10
			35°	121	250	2.0	10
0.30	0.65	0.35	45°	100	200	2.0	10
			35°	121	250	2.0	10
0.30	0.35	0.65	45°	100	200	2.0	10
			35°	100	250	2.0	7

Tablo 16 ve Tablo 17'deki sonuçlar incelendiğinde polygon eğim üst limitinin diğer parametrelere olan etkisi açıkça görülmektedir. Üst limit değeri azaldıkça PRM ile üretilen çizgedeki nokta ve kenar sayısı dolayısıyla çizgenin büyüklüğü azalır. Bu nedenle genel güzergah planlama adımından sağlanabilecek en yüksek katkı, nokta sayısı ve komşuluk yarıçapı parametrelerinin değerlerindeki artışla sağlanmıştır.

4.2.8 Yerel Güzergah Planlamaya Özgü Deneyisel Çalışmalar

Genel güzergah planlama aşamasında değişken parametrelerin farklı karakteristiğe sahip araziler ve farklı askeri misyonlar için belirlenmiş değerleri bu bölümde kullanılmıştır. Bu bölüm, önceki aşamada kullanılan askeri misyon ve arazi yapıları ile geliştirilen algoritmanın yerel güzergah planlama aşamasının performansını ölçmeye yönelik deneyler içermektedir.

Çalışmamızda yerel güzergah planlama aşamasında kullanılan parametreler sabit ve değişken olmak üzere iki gruba ayrılmıştır. Sabit değer alan parametreler Tablo 18’de, değişken değer alan parametreler ise Tablo 19’de verilmiştir.

Tablo 18: Yerel güzergah planlama algoritmasındaki sabit parametreler

	Sabit Parametreler	Değerler
1	Popülasyon büyüklüğü	50
2	Nesil sayısı	2000
3	Turnuva seçim büyüklüğü	5
4	Sensörün yön değiştirebildiği en düşük açı değeri	30°
5	Başlangıç popülasyonu oluştururken birincil ve ikincil ara noktaların bireye taşınma oranı	0.60
6	Birincil ara noktaların çözümdeki oranı	0.66
7	İkincil ara noktaların çözümdeki oranı	0.33
8	Tahmini güzergah uzunluğu sabiti	1.75
9	Maliyet hesaplamasında güzergahlar arasındaki mesafe farkı sabiti	0.50
10	Çarpazlama oranı	0.4
11	Erişilebilirlik sabiti (w_c)	5.0
12	Dinamik ortamda çarpışma kontrolü için gereken minimum mesafe	25
13	Sensör sayısı	3
14	Çarpışma kontrolünün tekrarlanma aralığı (iterasyon sayısı)	200

Tablo 19’de yer alan yerel güzergah planlama algoritmasındaki (YGP) değişken parametrelerin uygun değerlerini belirlemek için olası değerler arasında bir ön deney gerçekleştirilmiştir. Bu deneyde 6 farklı misyon için toplam 120 farklı kombinasyonda parametreler test edilmiştir. Sonuçların anlamlı olabilmesi için en az 30 replikasyonun yapılması gerektiğinden, her bir kombinasyon için program 30 defa çalıştırılmış ve elde edilen değerlerin ortalamaları bulunmuştur.

Tablo 19: Yerel güzergah planlama algoritmasındaki değişken parametreler ve olası değerleri

Çarpazlama	Mutasyon	Misyona Ağırlıkları		
		Görünürlük (w_v)	Maliyet (w_d)	Düzgünlük (w_s)
1 - Kontrol Noktaları – Rasgele	1 – Ara nokta ekleme	0.70	0.65	0.65
	2 – Ara nokta güncelleme	0.50	0.35	0.35
2 - Kontrol Noktaları – Görünürlük Tabanlı	3 –Ara nokta silme	0.30		
	4 – Kontrol Noktası Güncelleme			
	5 - Rasgele (1-4)			

Genel güzergah planlamada olduğu gibi deneylerde algoritmanın farklı arazi karakteristiklerindeki performansı ölçülmüştür. Engebeli bir arazi yapısında yapılan deneylerle belirlenen farklı misyonlar için parametre değerleri Tablo 20’e verilirken Tablo 21’de düzlük arazi yapısındaki sonuçlar yer almaktadır.

Tablo 20: Engelibeli arazi yapısında farklı askeri misyonlar için belirlenen yerel güzergah algoritması parametreleri

Misyona Ağırlıkları			Poligon Eğim Üst Limiti	Çaprazlama	Mutasyon
Görünürlük (w_v)	Maliyet (w_d)	Düzgünlük (w_s)			
0.70	0.65	0.35	45°	2	1
			35°	2	5
0.70	0.35	0.65	45°	2	1
			35°	2	1
0.50	0.65	0.35	45°	2	5
			35°	2	5
0.50	0.35	0.65	45°	2	5
			35°	2	5
0.30	0.65	0.35	45°	2	5
			35°	2	5
0.30	0.35	0.65	45°	2	5
			35°	2	5

Tablo 21: Düzgünlük arazi yapısında farklı askeri misyonlar için belirlenen yerel güzergah algoritması parametreleri

Misyona Ağırlıkları			Poligon Eğim Üst Limiti	Çaprazlama	Mutasyon
Görünürlük (w_v)	Maliyet (w_d)	Düzgünlük (w_s)			
0.70	0.65	0.35	45°	2	5
			35°	2	5
0.70	0.35	0.65	45°	2	1
			35°	2	1
0.50	0.65	0.35	45°	2	5
			35°	2	5
0.50	0.35	0.65	45°	2	5
			35°	2	5
0.30	0.65	0.35	45°	2	5
			35°	2	5
0.30	0.35	0.65	45°	2	5
			35°	2	5

Tablo 20 ve Tablo 21'deki sonuçlar incelendiğinde farklı misyon ağırlıklarına göre çaprazlama operatörü olarak görünürlük tabanlı çaprazlama, mutasyon operatörü olarak her nesilde mutasyon operatörünün rasgele seçildiği yöntem diğer operatörlere göre daha baskın olarak seçilmiştir.

Tablo 22: Engelibeli arazi yapısında genel güzergah algoritmasının performans değerlendirmesi

Misyon Ağırlıkları			Poligon Eğim Üst Limiti	Uygunluk Değerleri	
Görünürlük (w_v)	Maliyet (w_d)	Düzgünlük (w_s)		GGP	YGP
0.70	0.65	0.35	45°	0.5905	0.6954
			35°	0.5771	0.6913
0.70	0.35	0.65	45°	0.6311	0.7093
			35°	0.6271	0.7266
0.50	0.65	0.35	45°	0.6216	0.7385
			35°	0.5916	0.7411
0.50	0.35	0.65	45°	0.6651	0.7150
			35°	0.6699	0.7400
0.30	0.65	0.35	45°	0.6743	0.8386
			35°	0.6273	0.7967
0.30	0.35	0.65	45°	0.6960	0.8300
			35°	0.6728	0.7879

Tablo 23: Düzlük arazi yapısında genel güzergah algoritmasının performans değerlendirmesi

Misyon Ağırlıkları			Poligon Eğim Üst Limiti	Uygunluk Değerleri	
Görünürlük (w_v)	Maliyet (w_d)	Düzgünlük (w_s)		GGP	YGP
0.70	0.65	0.35	45°	0.5899	0.6901
			35°	0.5634	0.6971
0.70	0.35	0.65	45°	0.6229	0.7149
			35°	0.5958	0.7230
0.50	0.65	0.35	45°	0.6200	0.7762
			35°	0.5859	0.7512
0.50	0.35	0.65	45°	0.6674	0.7271
			35°	0.6312	0.7606
0.30	0.65	0.35	45°	0.6618	0.8435
			35°	0.6285	0.8803
0.30	0.35	0.65	45°	0.7010	0.8301
			35°	0.6798	0.8184

Tablo 20 ve Tablo 21’de verilen parametrelerle yapılan yerel güzergah planlama algoritması performans değerlendirmesinin sonuçları engelibeli ve düzlük arazi yapısı için sırasıyla Tablo 22 ve Tablo 23’de verilmiştir. Sonuçlar incelendiğinde her iki zorluktaki arazi yapısında GGP algoritmasının ürettiği güzergahlarda YGP ile tüm misyon ağırlıkları için iyileşme olduğu görülmektedir. Özellikle maliyet ve düzgünlük ağırlıklarının yüksek olduğu misyonlarda GGP algoritmasının ürettiği güzergahların uygunluk değerlerinde yüksek oranda artış görülmektedir. Bunun sebeplerinden biri yüksek eğimli arazilerde dar geçiş alanı olan bölgelerden seçilen noktalara erişimin YGP’deki sınırlı nokta sayısından dolayı çok dar açılarla dönüş gerektiren rotalar içermesidir. Bu dar açılar, YGP algoritmasındaki mutasyon operatörleri ile düzeltilerek daha gerçekçi güzergahlar elde edilmiştir. Uygunluk değerleri arasındaki fark da bunu göstermektedir. Örnek olarak görünürlük, maliyet, ve düzlük bileşenlerinin ağırlıklarının sırasıyla 0.30, 0.65, 0.35 alındığı testlerde YGP algoritması ile engelibeli arazide %24’lük iyileşme sağlanırken düzlük arazide %27’lik iyileşme sağlanmıştır.

5 Sonuçlar

Bu çalışmada, temsili olarak üretilen 3-boyutlu bir arazi üzerine sensörlerin konumlandırılması ve parametrelerinin tayini üzerinde durulmuştur. Çalışmamızın temel katkıları, sensör yerleştirme problemine yönelik olarak çok ölçütlü fayda teorisi tabanlı özgün bir model tasarlanması, bu problemin çözümüne ilişkin olarak karma evrimsel algoritma tabanlı yeni bir çözüm tekniğinin sunulması ve çoklu hareketli sensörlerin güzergah planlama problemi için yeni bir karma çözümün geliştirilmesidir.

Tasarlanan çok ölçütlü fayda teorisi tabanlı modeldeki temel varsayım; bir askeri harekâta sensörlerin algılama davranışının birbirleriyle çelişen üç ölçüt arasında rasyonel bir ödünleşmeye dayanması gerektiğidir. Bu üç ölçüt; arazide görünen alanın maksimize edilmesi, karşı güç tarafından tespit edilebilecek bilginin minimize edilmesi ve konumlandırılan sensörlerin toplam maliyetinin minimizasyonudur.

Sensör konumlandırılması problemi için geliştirdiğimiz karma evrimsel algoritma tabanlı çözümün, farklı özellikte arazi yapıları, hava koşul bileşenleri ve ortamdaki doğal ve yapay engelleri de göz önüne alınarak, çeşitli askeri keşif uygulamalarında farklı ölçütler için etkin bir şekilde kullanılabileceğini göstermektedir. Deneysel çalışmamız, görünürlük-ağırlıklı, gizlilik-ağırlıklı ve maliyet ağırlıklı ölçütlerin kendi aralarında en uygun dengenin sağlandığını göstermektedir.

Projedeki hedeflerimizi gerçekleştirmek için geliştirdiğimiz, 3-boyutlu arazi ve sensör yapıları üreten çok fonksiyonlu simülasyon ortamı, farklı alanlardaki benzer birçok problem üzerinde çalışma imkânı sağlamıştır. Bu kapsamda, çalışmamızda geliştirilen çok-fonksiyonlu üç-boyutlu arazi ve sensör simülatörü revize edilerek, kablosuz çoklu ortam sensör ağlarına yönelik bağlı kapsama problemi için, kapsanan alanın maximizasyonu ve sensör maliyetinin minimizasyonu hedefleyen evrimsel algoritma tabanlı özgün bir çözüm sunulmuştur.

Projemizde, hareketli sensörlerin güzergah planlama problemi için, çıkılabilecek en yüksek eğim ve en düşük dönüş açısı gibi farklı kısıtların dikkate alındığı yeni bir karma çözüm de geliştirilmiştir. Bu problemin karakteristikleri içerisinde yer alan çoklu güzergah planlama, ortamın temsili 3-boyutlu araziden oluşması, görünürlük, mesafe, erişilebilirlik gibi ölçütlere göre güzergahların oluşturulması, problemin karmaşıklık boyutunu göstermektedir. Farklı sayıda ve özellikte hareketli sensör platformları için istenilen başlangıç ve hedef noktaları arasında verilen ölçütlere göre arazi alanını maksimum kapsayan gerçekçi güzergahların oluşturulması, geliştirilen yöntemin ne kadar esnek ve sağlam olduğunu göstermektedir.

6 Kaynakça

- ACAR E.U., and CHOSET H., Robust sensor-based coverage of unstructured environments, in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, (2001).
- AKYILDIZ I.F., MELODIA T., and K.R. Chowdhury, A survey on wireless multimedia sensor networks, *Computer Networks*, 51 (4), (2007).
- ANDERSSON J., *Multiobjective Optimization in Engineering Design - Applications to Fluid Power Systems*. Dissertation, Linking Studies in Science and Technology, Dissertation No. 675, Linking University, Linking, Sweden, (2001).
- BARRAQUAND J., LATOMBE J.C., A Monte-Carlo algorithm for path planning with many degrees of freedom. In *Proceedings IEEE International Conference on Robotics & Automation*, 1712–1717, (1990).
- BHATTACHARYA P., GAVRILOVA M.L., Voronoi diagram in optimal path planning, *The 4th International Symposium on Voronoi Diagrams in Science and Engineering*, United Kingdom, (2007).
- CHEN S. Y., and LI Y.F., Optimal Viewpoint Planning for Model-Based Robot Vision, In *Proceedings of IEEE 2002 World Congress on Computational Intelligence (WCCI)/Congress on Evolutionary Computation*, 634–639, (2002).
- CHEN S. Y., and LI Y.F., Automatic Sensor Placement for Model-Based Robot Vision, *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol.34 no.1, 893–408, (2004).
- CLEMEN R.T., *Making Hard Decision*. PWS Kent Publishing, Boston, MA, (1991).
- COOK D.J., GMYTRASIEWICZ P., and LAWRENCE B., Decision-Theoretic Cooperative Sensor Planning, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.18 no.10, 1013–1023, (1996).
- DYER J.S., *Remarks on the Analytic Hierarchy Process* . Management Science, Vol. 36, No. 3, 249-259, (1990).
- EIBEN A.E. and SMITH J.E., *Introduction to Evolutionary Computing*, Springer Verlag, (2003).
- ERDEM U.M., and SCLAROFF S.. Optimal Placement of Cameras in Floorplans to Satisfy Task Requirements and Cost Constraints, In *Proceedings of OMNIVIS Workshop*, 111–120, (2004).
- FOURNIER A., FUSSEL D., and CARPENTER L., Computer Rendering of Stochastic Models, *Communications of the ACM*, vol. 25, no. 6, 371-384, (1982).
- FOX B.R. and MCMAHON M.B., Genetic Operators for Sequencing Problems. In Rawlins G. (Ed.), *Foundations of Genetic Algorithms, First Workshop on the Foundation of Genetic Algorithms and Classifier Systems*, Morgan Kaufmann, (1990).
- GHOSH A., and DAS S.K., Coverage and Connectivity Issues in Wireless Sensor Networks: A Survey, *Pervasive and Mobile Computing*, 4 (3), pp. 303-334, (2008).

GREFFENSTETTE J.J., Incorporating Problem Specific Knowledge into a Genetic Algorithm, in *Genetic Algorithms and Simulated Annealing*, ed: Davis L., Morgan Kaufmann Publishers, (1987).

HALL P., *Introduction to the theory of coverage processes*, John Wiley and Sons, (1988).

HANDERSON D., *ModTerrain: A Proposed Standart for Terrain Representation in Entity Level Simulation*, Master's Thesis, Naval Postgraduate School, California, (1999).

HERT S., TIWARI S., and LUMELSKY V., A terrain covering algorithm for an AUV, *Autonomous Robots*, vol. 3, 91-119, (1996).

HORSTER E., and LIENHART R., On the Optimal Placement of Multiple Visual Sensors, In *Proceedings of the 4th ACM International Workshop on Video Surveillance and Sensor Networks*, 111–120, (2006).

JIA J., CHEN J., CHANG G., WEN Y. and SONG J., Multi-objective Optimization for Coverage Control in Wireless Sensor Network with Adjustable Sensing Radius, In *Computers and Mathematics with Applications*, 1767-1775, (2009).

KAVRAKI L.E., SVETSKA P., LATOMBE J.C., and OVERMARS, M.H., Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces. *IEEE Transactions on Robotics and Automation*, 12, 566-580, (1996).

KEENEY R.L., and RAIFFA H., *Decision with Multiple Objectives: Preferences and Value Tradeoffs*. John Wiley and Sons, New York, (1993).

KRISTENSEN S., Sensor Planning with Bayesian Decision Theory, *Robotics and Autonomous Systems*, vol.19, 273–286, (1997).

LATOMBE J.C., *Robot Motion Planning*, Kluwer, Boston, MA,(1991).

LA VALLE S.M., *Planning Algorithms*, Cambridge University Press, (2006)

LA VALLE S.M., Rapidly exploring random trees: a new tool for path planning, Technical Report 98-11, Computer Science Department, Iowa State University, (1998).

LUO C., YANG S., and YUAN X., Real-time area covering operations with obstacle avoidance for cleaning robots, in *Proceedings of the 2002 IEEE International Conference on Intelligent Robots and Systems*, Switzerland, (2002).

MARENGONI M., DRAPER B.A., HANSON A. and SITAMARAN R., A System to Place Observers on a Polyhedral Terrain in Polynomial Time, *Image and Vision Computing*, vol.18, No.10, 773–780, (2000).

MICHALEWICS Z. and FOGEL D.B., *How to Solve It: Modern Heuristics*, Springer Verlag, (2000).

MIHAYLOVA L., LEFEBRE T., BRUYNINCKX H., GADEYNE K., and DE SCHUTTER J., Active Sensing for Robotics - a Survey, In *Proceedings of International Conference in Numerical Methods and Applications*, vol.19, 316–324, (2002).

MITTAL A. and DAVIS L.S., Visibility Analysis and Sensor Planning in Dynamic Environments, In *Proceedings of Computer Vision - ECCV 2004*, 175–189, (2004).

MITTAL A., Generalized Multi-sensor Planning. In *Proceedings of Computer Vision - ECCV 2006, Lecture Notes in Computer Science*, 522–535, (2006).

MITTAL A., and DAVIS L.S., A general method for sensor planning in multi-sensor systems: extension to random occlusion, *Int. J. Comput. Vision*, 76 (1), 31-52, (2008).

MOHAN M., SAWHNEY R., KRISHNA K.M., and SRIKANTH M.B., Covering hostile terrains with partial and complete visibilities: On minimum distance paths, in *IEEE International Conference on Intelligent Robots and Systems*, 2572-2577, (2008).

MONTGOMERY D.C., *Design and Analysis of Experiments*, Sixth Edition, John Wiley-Sons Inc., (2005).

MOSCATO P., On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms, Tech. Rep. Caltech Concurrent Computation Program Report 826, Caltech, (1989).

NAJJARAN H., and KIRCANSKI N., Path planning for a terrain scanner robot, in *Proceedings of 31st International Symposium of Robotics*, Canada, (2000).

NYSTROM B., Terrain Generation Tutorial: Hill Algorithm, (2000).

OLLIS M., and STENTZ A., First results in vision based crop line tracking, in *Proceedings of the IEEE International Conference on Robotics Automation*, Minneapolis, 951-956, (1996).

O'ROURKE J., *Art Gallery Theorems and Algorithms*. Oxford University Press, (1987).

O'ROURKE J., Computational geometry column, *International Journal of Computational Geometry and Applications*, 2 (2), 215-217, (1992).

PIRZADEH A., and SNYDER W., A unified solution to coverage and search in explored and unexplored terrains using indirect control, in *Proceedings of IEEE International Conference on Robotics Automation*, 2113-2119, (1990).

PROCTOR M.D., and GERBER W.J., Line-of-sight Attributes for a Generalized Application Program Interface, *The Journal of Defense Modeling and Simulation*, Vol. 1, No. 1, 43-57, (2004).

RADCLIFFE N.J. and SURRY P.D., Formal Memetic Algorithms, *Evolutionary Computing: AISB Workshop*, Springer-Verlag LNCS 865, 1-16, (1994).

RAIDL G., Hybrid Evolutionary Algorithms for Combinatorial Algorithms, Habilitation Thesis, Vienna University of Technology, (2002).

REED M. K. and ALLEN P.K., Constraint-Based Sensor Planning for Scene Modeling, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, 1460–1467, (2000).

RISEMAN E., SCHULTZ H., LIM W.J., DRAPER B., and HANSON A., Deadalus Battlefield Visualization System, In *Proceedings of ARPA Image Understanding*, (1996).

SPLETZER J. and TAYLOR C., Sensor Planning and Control in a Dynamic Environment, In *Proceedings of International Conference on Robotics and Automation*, (2002).

STAMOS I., and ALLEN P.K., Interactive Sensor Planning, In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, 489–494, (1998).

TARABANIS K.A., ALLEN P.K., and TSAI R.Y., A survey of sensor planning in computer vision, *IEEE Transactions on Robotics and Automation*, 11 (1), 86-104, (1995).

TEZCAN N., and WANG W., Self-Orienting Wireless Multimedia Sensor Networks for Occlusion Viewpoints. *Computer Networks*, vol.52, 2558–2567, (2008).

TIAN J., ZHENG Y., ZHU H., and SHEN L., A MPC and genetic algorithm based approach for multiple UAVs cooperative search, *International Conference on Computational Intelligence and Security*, China, 399-404, (2005).

VON NEUMANN J., and MORGENSTEIN O., *The Theory of Games and Economic Behaviour*. Princeton University Press, New Jersey, (1964).

WANG J., NIU C. and SHEN R., Priority-Based Target Coverage in Directional Sensor Networks Using Genetic Algorithm, *Computers and Mathematics with Applications*, (2008).

XAVIER P.G., Approximation algorithms for optimal kinodynamic robot plans, PhD Dissertation, Computer Science Department, Cornell University, Ithaca, NewYork, (1992).

XING G., WANG X., ZHANG Y., LU C., PLESS R., and GILL C., Integrated Coverage and Connectivity Configuration for Energy Conservation in Sensor Networks, *ACM Trans. on Sensor Networks*, 1 (1), 36-72, (2005).

YANG G., and QIAO D., Critical Conditions for Connected-k-Coverage in Sensor Networks, *IEEE Communications Letters*, 12 (9), (2008).

YANG S., and LUO C., A neural network approach to complete coverage path planning, *IEEE Transactions on Systems, Man and Cybernetics – Part B*, vol. 34, (2004).

YE F., ZHONG G., LU S., and ZHANG L., PEAS: A Robust Energy Conserving Protocol for Long-lived Sensor Networks, *The 23rd International Conference on Distributed Computing Systems (ICDCS'03)*, (2003).

ZHOU Z., DAS S.R., and GUPTA H., Connected K-Coverage Problem in Sensor Networks, in: *Proc. of the 13th Int. Conf. on Computer Communication and Networks (IC3N)*, Chicago, IL, (2004).

ZHU D., LATOMBE J., New heuristic algorithms for efficient hierarchical path planning, *IEEE Transaction on Robotics and Automation I*, 9-20, (1991).

7 Ekler

7.1 EK-1 Proje Kapsamında Yapılan Yayınlar

- “Positioning and Utilizing Sensors on a 3D Terrain: Part I – Theory and Modeling”, Haluk Rahmi Topcuoglu, Murat Ermis and Mesut Sifyan, IEEE Transactions on Systems, Man and Cybernetics: Part C: Applications and Review. *(kabul edildi)*
- “Positioning and Utilizing Sensors on a 3D Terrain: Part II – Solving with Hybrid Evolutionary Algorithm”, Haluk Rahmi Topcuoglu, Murat Ermis and Mesut Sifyan, IEEE Transactions on Systems, Man and Cybernetics: Part C: Applications and Review. *(kabul edildi)*
- “A New 3-D Wireless Multimedia Sensor Network Simulation Environment for Connected Coverage Problems”, Haluk Rahmi Topcuoglu, Murat Ermis, İlker Bekmezci and Mesut Sifyan, Simulation: Transactions of the Society for Modeling and Simulation International. *(düzeltme aşamasında)*
- “On the Optimal Deployment of Wireless Multimedia Sensor Networks Using Genetic Algorithms for Three-Dimensional Terrains”, Haluk Rahmi Topcuoglu, Murat Ermis, İlker Bekmezci and Mesut Sifyan, Computer Networks. *(değerlendirme aşamasında)*
- “Hybrid Evolutionary Algorithms for Sensor Placement on a 3D Terrain”, Haluk Rahmi Topcuoglu, Murat Ermis and Mesut Sifyan, International Conference on Intelligent Systems Design and Applications (ISDA), Pisa, İtalya, Kasım 2009.
- “Path Planning for Mobile Sensor Platforms Using Hybrid Evolutionary Algorithms on a 3D Terrain”, Mesut Sifyan, Haluk Rahmi Topcuoglu ve Murat Ermis, 2010 IEEE World Congress on Computational Intelligence (WCCI 2010), Barcelona, İspanya, Temmuz 2010.

TÜBİTAK

PROJE ÖZET BİLGİ FORMU

Proje No: 106E159
Proje Başlığı: Karma Evrimsel Algoritmalar Yardımı ile Temsili Olarak Üretilmiş bir Alan Üzerinde Sensörlerin Konumlandırılması ve Parametrelerinin Tayini
Proje Yürütücüsü ve Araştırmacılar: Prof. Dr. Haluk Topcuoglu (Proje Yürütücüsü) Yrd. Doç. Dr. Murat Ermiş (Araştırmacı) Mesut Sifyan (Araştırmacı)
Projenin Yürütüldüğü Kuruluş ve Adresi: Marmara Üniversitesi, Bilgisayar Mühendisliği Bölümü Göztepe Kampüsü, 34722 Kadıköy / ISTANBUL
Destekleyen Kuruluş(ların) Adı ve Adresi: TÜBİTAK, Atatürk Bulvarı, No:22 Kavaklıdere/ANKARA 06100
Projenin Başlangıç ve Bitiş Tarihleri: 01/02/2007- 30/10/2009
Öz (en çok 70 kelime): Projemizde, temsili olarak üretilen 3-boyutlu bir arazi için sensörlerin konumlandırılması ve parametrelerinin tayinini içeren çok ölçütlü fayda teorisi tabanlı yeni bir model tasarlanmış ve gerçekleştirilmiştir. Bu modele yönelik olarak, özgün bir karma evrimsel algoritma geliştirilmiştir. Yapılan deneysel çalışmalar, çözümümüzün çeşitli askeri uygulamalarda için etkin bir şekilde kullanılabileceğini göstermektedir. Projemiz kapsamında, hareketli sensörlerin güzergah planlama problemi için, çıkılabilecek en yüksek eğim ve en düşük dönüş açısı gibi kısıtların dikkate alındığı yeni bir çözüm yöntemi de geliştirilmiştir.
Anahtar Kelimeler: Karma Evrimsel Algoritmalar, Sensör Planlanması, Çok-Ölçütlü Eniyileme, Çok-Ölçütlü Fayda Teorisi, Yerel Arama, Olasılıksal Yol Haritası
Fikri Ürün Bildirim Formu Sunuldu mu? Evet ☐ Gerekli Değil ☒ Fikri Ürün Bildirim Formu'nun tesliminden sonra 3 ay içerisinde patent başvurusu yapılmalıdır.

Projeden Yapılan Yayınlar:

- “Positioning and Utilizing Sensors on a 3D Terrain: Part I – Theory and Modeling”, Haluk Rahmi Topcuoglu, Murat Ermis and Mesut Sifyan, IEEE Transactions on Systems, Man and Cybernetics: Part C: Applications and Review. *(kabul edildi)*
- “Positioning and Utilizing Sensors on a 3D Terrain: Part II – Solving with Hybrid Evolutionary Algorithm”, Haluk Rahmi Topcuoglu, Murat Ermis and Mesut Sifyan, IEEE Transactions on Systems, Man and Cybernetics: Part C: Applications and Review. *(kabul edildi)*
- “A New 3-D Wireless Multimedia Sensor Network Simulation Environment for Connected Coverage Problems”, Haluk Rahmi Topcuoglu, Murat Ermis, İlker Bekmezci and Mesut Sifyan, Simulation: Transactions of the Society for Modeling & Simulation International. *(düzeltme aşamasında)*
- “On the Optimal Deployment of Wireless Multimedia Sensor Networks Using Genetic Algorithms for Three-Dimensional Terrains”, Haluk Rahmi Topcuoglu, Murat Ermis, İlker Bekmezci and Mesut Sifyan, Computer Networks. *(değerlendirme aşamasında)*
- “Hybrid Evolutionary Algorithms for Sensor Placement on a 3D Terrain”, Haluk Rahmi Topcuoglu, Murat Ermis and Mesut Sifyan, International Conference on Intelligent Systems Design and Applications (ISDA), Pisa, İtalya, Kasım 2009.
- “Path Planning for Mobile Sensor Platforms Using Hybrid Evolutionary Algorithms on a 3D Terrain”, Mesut Sifyan, Haluk Rahmi Topcuoglu ve Murat Ermis, 2010 IEEE World Congress on Computational Intelligence (WCCI 2010), Barcelona, İspanya, Temmuz 2010.

On the Optimal Deployment of Wireless Multimedia
Sensor Networks Using Genetic Algorithms for
Three-Dimensional Terrains

Haluk Topcuoglu^a, Murat Ermis^b, Ilker Bekmezci^c, Mesut Sifyan^a

^aComputer Engineering Department, Marmara University, Kuyubasi, 34722, Istanbul, Turkey

^bIndustrial Engineering Department, Turkish Air Force Academy, Yesilyurt, 34149, Istanbul, Turkey

^cComputer Engineering Department, Turkish Air Force Academy, Yesilyurt, 34149, Istanbul, Turkey

Abstract

Wireless multimedia sensor network (WMSN) is a new type of network that can retrieve multimedia data like video, audio, via multimedia sensors and send the data to a sink through its wireless communication unit. The connected coverage and the cost are two of the most important design considerations for WMSN, and especially on three dimensional terrain, the sensor node deployment affects the coverage performance of the network directly. In this paper, we try to locate and utilize multimedia sensors not only to maximize coverage but also to minimize the cost while preserving the connected property of the network with genetic algorithm (GA) in a sythetically generated 3D terrain. In order to unify conflicting objectives, a global utilization function that includes the utility of the cost and the utility of the coverage is proposed by the help of Multi Attribute Utility (MAU) analysis, and an optimal solution of the global utility function is searched with GA under the connectivity constraint. Experimental analysis has shown that our approach can find a good balance between the cost and the coverage on 3D terrain.

Keywords: WMSN, connected coverage, cost, sensor deployment, optimization, genetic algorithms

Email addresses: topcuoglu@m.com.tr (Haluk Topcuoglu), m.ermis@hho.edu.tr (Murat Ermis), i.bekmezci@hho.edu.tr (Ilker Bekmezci), m.sifyan@m.com.tr (Mesut Sifyan)

1. Introduction

Wireless multimedia sensor networks (WMSN) are composed of sensor nodes that can retrieve video or audio information and relay the retrieved data to the sink [1]. Connected coverage concept is one of the most important issues for WMSN deployment. WMSN nodes should be deployed so that the covered area is maximized, and the nodes must be able to form a connected network. This paper tries to locate and utilize wireless multimedia sensor networks to maximize coverage and to minimize cost under the connectivity constraint on three-dimensional (3D) synthetically generated terrain with Genetic Algorithms (GA).

The notion of the area coverage can be considered as a measure of the quality of service (QoS) for WSN and WMSN [2]. Basically, there are three types of coverage which are the blanket coverage that maximizes the detection rate of targets, the barrier coverage that maximizes intrusion detection probability through the barrier, and the sweep coverage that maximizes the detection rate and minimizes the number of missed detections [3]. In the context of this paper, only the blanket coverage problem is considered.

The coverage problem has been studied extensively in the literature for many different types of applications [4]. The coverage problem for video sensors is closely related to the traditional Art Gallery Problem which aims to determine the number of observers and their locations, such that every point in the art gallery is seen by at least one observer [5]. Although Art Gallery problem can be solved optimally in a 2D plane, it is shown to be NP-hard when being extended to a 3D space [6]. There are only approximate solutions for 3D Art Gallery Problem [7].

Furthermore, video sensor planning efforts has concentrated on incorporating different kinds of constraints like resolution, focus, field of view, visibility, view angle, and prohibited regions [8]. Mittal has introduced a general approach to integrate these static and dynamic constraints [9]. Horster and Lienhart have proposed a binary integer programming model that includes a cost constraint [10]. The target coverage issue in multimedia sensor networks is discussed and approximated with using genetic algorithm approach by Wang et al. [11]. Erdem and Sclaroff have optimized the video sensor locations for a given floor plan with binary optimization [12].

In recent years, with the availability of low-cost and tiny CMOS cameras,

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

microphones, along with wireless communication unit, WMSN, in which the nodes have multimedia sensors, are realized to get visual or audio information from the environment. The problem of self-orientation (i.e. the direction of the video sensors) in WMSN is also studied to maximize multimedia coverage [13]. Genetic Algorithm technique is used to locate and utilize video sensors for maximization of coverage, maximization of stealth, and minimization of cost on a synthetically generated 3D terrain [14].

Although the coverage is an important issue for WMSN, the connectivity is also crucial for operating a network. The connectivity is very closely related to sensor node deployment like the coverage. This is why coverage is discussed together with connectivity [15]. Firstly, the connected coverage concept is considered for WSN. Probing Environment and Adaptive Sleeping Protocol, PEAS, is one of the first WSN designs to address both coverage and connectivity in a configurable fashion [16]. However, PEAS does not provide analytical guarantees on the degree of coverage and connectivity. A geometric analysis of the fundamental relationship between the coverage and the connectivity is presented by Xing et al. [17]. It also introduces the design and analysis of novel protocols that can dynamically configure a network to achieve guaranteed degrees of coverage and connectivity. In [18], the problem of selecting connected k-cover problem which is to select a set of sensors such that each point in the sensor network is covered by at least k different sensors is considered. Yang and Quiao are discussed critical conditions for connected-k-coverage in WSN [19].

The connected coverage property is an important issue not only for WSN but also for other types of sensor networks. Deployment strategies for 2D and 3D communication architectures for underwater acoustic sensor networks are proposed to maximize the connected coverage [20]. Han et al. has investigated the problems of deploying a minimum number of directional sensors to form a connected WMSN network to cover a certain terrain [21].

The connected coverage performance can be affected by the terrain structure. Majority of the researches about WSN or WMSN models the terrain as 2D. In 2D space, while the scalar sensor coverage is modeled as circle, multimedia sensor coverage is modeled as sector. The 2D terrain modeling is accepted as sufficient for especially indoor applications. However, there are many applications which operates on 3D terrain, like outdoor military surveillance. The difference between 2D and 3D video sensor coverage area is illustrated in Figure 1. In 2D-based model, the video sensor coverage is a sector, whereas on 3D-based model, the coverage area is a cone.

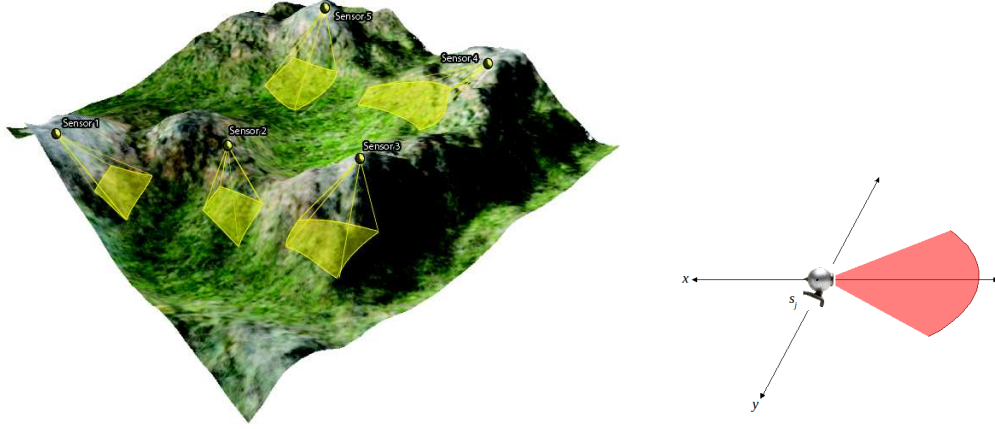


Figure 1: Coverage areas of video sensors on (a) 3D terrain (b) 2D terrain

In this paper, the connected coverage problem is investigated for WM-SNs on 3D occluded terrain. The main aim is to locate different multimedia sensors and to set their orientational parameters (i.e., tilt, heading angle) on a synthetically generated 3D terrain with conflicting objectives: the maximization of the coverage, and the minimization of the cost, while preserving the connected property of the wireless multimedia sensor network. In order to formulate the problem, a unified utility function that includes the utility of the coverage and the utility of the cost is developed. We try to find the optimal solution of the global utility function by using a novel GA approach.

The remainder of the paper is organized as follows. In Section 2, the problem definition is presented. In Section 3, GA-based formulation to solve the sensor deployment problem is explained. Experimental study and performance results are discussed in Section 4. Section 5 concludes the paper.

2. The Connected Coverage Problem Formulation

In this section, a formal definition and mathematical formulation of the connected coverage problem is provided. For the formal definition of this deployment problem, terrain, sensor, visibility and connectivity terms firstly is defined.

2.1. Terrain

Let a terrain T be a set of polygons, $T = \{p_1, \dots, p_n\}$, such that each polygon p_i has a set of attributes given below:

- x, y, z coordinates; x_i, y_i, z_i
- Normal vector, n_i
- cm_i , the center of mass point of p_i
- Weather density, w_i , where $0 \leq w_i \leq 1$, which is used to represent the weather conditions of p_i
- Object density, o_i , where $0 \leq o_i \leq 1$, which is used to represent the occlusions of p_i

In this paper, a previously developed terrain generator, TerGen, is used [14]. TerGen supports two different common terrain generation algorithms: Hills Algorithm [22] and Diamond-Square Algorithm [23]. These algorithms require a number of parameters in order to set the roughness or smoothness of the terrain. After the generation of the terrain, it is also possible to create different occlusions and weather conditions that can affect the visibility conditions. Occlusions can be natural objects like forest, bush and etc., or artificial objects like buildings. The weather conditions, like fog, rain or snow can be set either for the whole terrain or for a user-selected region of the terrain.

2.2. Sensors

Let S be a set of multimedia sensors, $S=\{s_1,...,s_m\}$. Each multimedia sensor has a set of technological and deployment attributes. Technological attributes of sensor s_j are as follows:

- Depth of view, d_j
- Horizontal viewing angle, h_j
- Vertical viewing angle, v_j
- Acquisition cost, c_j
- Communication range, r_j

Deployment attributes of a multimedia sensor node are as follows:

- Position, p_{s_j} . The sensors are assumed to be located on the center of mass point of a polygon. The position of a sensor is represented as the polygon number on which the sensor is located.
- Heading angle, θ_j
- Tilt angle, β_j

Figure 2 illustrates the attributes of a multimedia sensor.

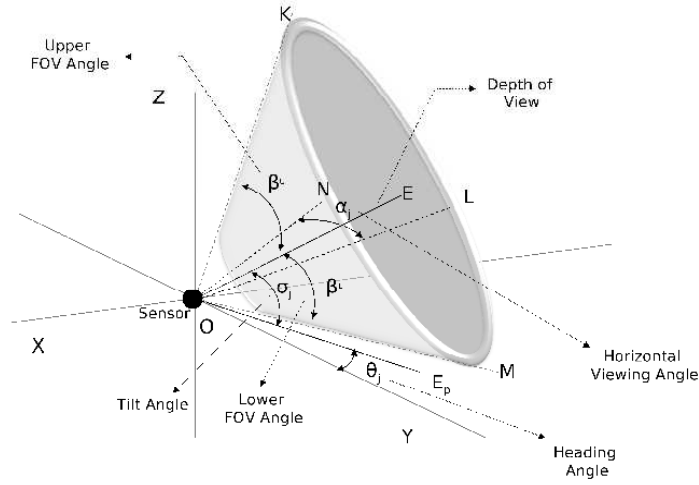


Figure 2: Sensor attributes of sensor s_j

Technological attributes: $|OE| = d_j$, $\widehat{BOC} = v_j$, $\widehat{AOC} = h_j$. Deployment attributes: O is the center of mass of p_j , $\widehat{ZOE} = \beta_j$, $\widehat{YOE_p} = \theta_j$, where E_p is the projection of E on xy plane.

2.3. Coverage Analysis

The coverage analysis used in this paper is based on the coverage modeling proposed in [14]. The coverage performed by a set of sensors S on a given terrain T is computed by adding the coverage of the polygons on the terrain T . The total coverage formulation on T is expressed in Equation 1.

$$TotalCov(S, T) = \sum_{p_i \in T} C(S, p_i) \quad (1)$$

$C(S, p_i)$ is the coverage value of a polygon p_i and it is computed as the average visibility of selected polygon points. Since calculating the visibility

values for all points in a polygon is computationally expensive, three corners and center of mass point of the polygon p_i are selected as representative points for visibility calculations, (i.e., x_i, y_i, z_i, cm_i). The formulation of the coverage of a polygon is presented in Equation 2.

$$C(S, p_i) = \frac{V(S, x_i) + V(S, y_i) + V(S, z_i) + V(S, cm_i)}{4} \quad (2)$$

Since the point b can be recognized by multiple sensors (with different visibility values), maximum visibility of b is considered as in Equation 3.

$$V(S, b) = \max \{VS(s_j, b)\} \quad (3)$$

where b is a point on a polygon and $s_j \in S$. The term $VS(s_j, b)$, which is the visibility of the point b from the sensor s_j , is formally defined in Equation 4.

$$VS(s_j, b) = LOS(s_j, b) \times VC(s_j, b) \times O(s_j, b) \quad (4)$$

where $LOS(s_j, b)$ is the line-of-sight (LOS) constraint, $VC(s_j, b)$ is the view cone constraint, and $O(s_j, b)$ is the aggregated occlusion effect between s_j and b .

The formal expressions of $LOS(s_j, b)$, $VC(s_j, b)$, and $O(s_j, b)$ are presented in the following subsections.

2.3.1. Line-of-Sight Constraint

Let b be a point on p_i , and $\vec{n}_{p_i}$ is the normal vector of p_i . If s_j is a multimedia sensor located on p_k , then $\vec{V}_{s_j, b}$ is the vector from s_j to b . It is assumed that s_j is deployed at cm_k (Figure 3).

Sensor s_j can have visual contact with point b , if the angle between the normal vector of p_i (i.e., $\vec{n}_{p_i}$) where b is located and the vector $\vec{V}_{s_j, b}$ from s_j to b is between 90° and 180° , and the angle between the normal vector of p_k (i.e., $\vec{n}_{p_k}$) and the vector $\vec{V}_{s_j, b}$ is between 0° and 90° .

$$LOS(s_j, b) = \begin{cases} 1, & \text{if } \vec{n}_{p_i} \cdot \vec{V}_{s_j, b} < 0 \wedge \vec{n}_{p_k} \cdot \vec{V}_{s_j, b} > 0 \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

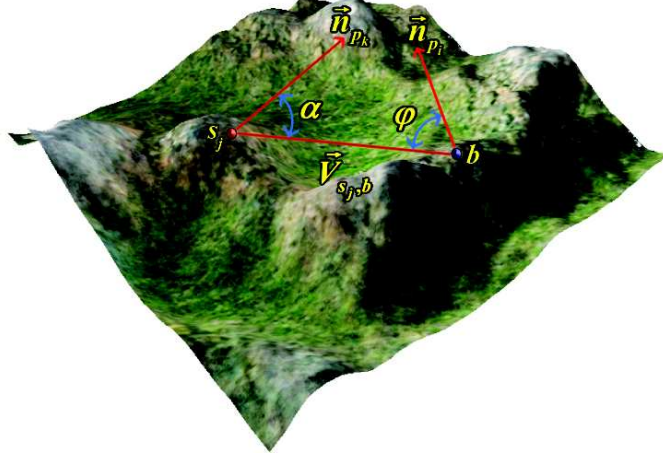


Figure 3: Visual contact calculation between a sensor and a point on terrain

2.3.2. View Cone

Point b must be in the view-cone of s_j which is depicted in Figure 2. The mathematical expression of this constraint is as follows:

$$VC(s_j, b) = \begin{cases} 1, & \text{if } \vec{n}_{p_k} \cdot \vec{V}_{s_j b} - \|\vec{n}_{p_k}\| \times \|\vec{V}_{s_j b}\| \cos(\frac{\alpha}{2}) \geq 0 \wedge d(s_j, b) \leq d_j \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where $d(s_j, b)$ is the euclidean distance from s_j to b .

2.3.3. Occlusions

The total occlusion density between the sensor s_j and the point b is the product of the densities of occlusions on the line between s_j and b .

$$O(s_j, b) = \prod_{o_k \in L_{s_j, b}} o_k \quad (7)$$

where $L_{s_j, b}$ is the line between s_j and b .

2.4. Connectivity Analysis

Connectivity is an important constraint for our problem. A network is connected if a node can communicate with all the other nodes. In order to test the network against connectivity, a connection assumption is needed. In

many studies, two sensor node can communicate with each other, if they are in a certain communication range. However, due to weather conditions or occlusions, they may not be able to communicate even if they are located in the range. In this paper, it is assumed that two sensor nodes s_i and s_j are connected, if the visibility between the sensors is greater than a predefined threshold (κ) value and the distance between them is less than the connection range of the sensors, r_i and r_j .

$$Cn(s_i, s_j) = \begin{cases} 1, & \text{if } LOS(s_i, s_j) = 1 \wedge d(s_i, s_j) \leq r_i, r_j \wedge V(s_i, s_j) \geq \kappa \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

where $d(s_i, s_j)$ is the euclidean distance from s_i to s_j .

Communication on the network is performed by transmitting data using the connections on intermediate nodes between sending and receiving nodes. On testing the connectivity of a network, a graph representation is designed where the nodes represent sensors, and the edges represent connections between two sensors. The connectivity testing of the network is performed by depth-first search (DFS) algorithm which uses the network graph.

3. Proposed Genetic Algorithm Based Solution Approach for Sensor Deployment

Although simple genetic algorithms have been widely used for many optimization problems in the literature, they are not generally efficient for the complex combinatorial problems [24]. In order to improve performance of evolutionary algorithms, specialized and/or problem-specific operators [25, 26, 27, 28] are combined with evolutionary algorithms. These complex hybrid systems are called with different names in related work, including hybrid genetic algorithms, hybrid evolutionary algorithms, genetic local search algorithms, and memetic algorithms [29, 30].

For optimization problem of locating multimedia sensor network, we consider several methods in our algorithm. In generation of initial population, problem-specific heuristics are considered to get feasible solutions. For variation operator, Connectivity-Based Crossover operator, which incorporate problem specific knowledge, is designed. And a repair function is implemented

to increase the number of feasible solutions.

In our GA-based approach, each solution contains the type and the locational attributes (position, heading angle, tilt angle) of sensors (see Figure 4). It should be noted that there is no restriction on sensor quantity; therefore string representation supports variable-length chromosomes. Additionally, this work has no restriction on the order of sensors in a given solution, such as determining the sensor order based on locations of sensors.

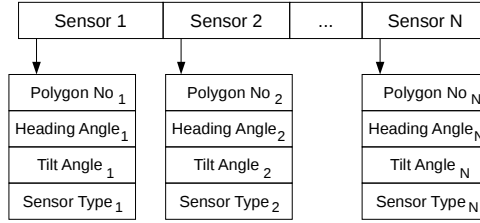


Figure 4: String Representation in Genetic Algorithm

In this study, a steady-state Genetic Algorithm is applied which generates one individual at each iteration. We consider tournament selection mechanism and the tournament size is varied in the experiments. In each iteration, firstly a crossover operator is applied on two parents that are selected by the tournament selection method. Crossover operator considers problem-specific information in order to carry the sensors from parents to offspring. Then a mutation operator is applied on the output of the crossover operator. Finally, the repair function is applied to ensure the connectivity of the solution.

3.1. Objective Function

In this paper, locating multimedia sensors is expressed as a multiobjective optimization problem and it is solved by the help of priori articulation method [31]. In this method, decision-maker decides the objective preferences, before the optimization is performed. Weighted sum and multi-attribute utility analysis (MAUA) are two examples of priori articulation. The basic principle of weighted sum and MAUA is to produce one single function, total utility function, in terms of fundamental objective functions. Firstly, every objective function must be mapped into a utility function that produces dimensionless scalar quantity expressing the goodness of a particular design [32]. In most

of the cases, the utility values changes between 0 and 1. In weighted sum, total utility function always forms a strictly convex combination of objectives. Although the easiest to understand and implement method is weighted sum approach, it can not represent the interactions between different objectives [33]. Multi attribute utility (MAU) method overcomes the interaction limitations of the weighted sum approach.

Although utility function theory has a well-known history in optimization [33], there are only a few utility theory studies in the sensor network context. Byers and Nasser has proposed a utility-based sensor selection method in [34]. Isler and Bajscy has developed a utility function based model to locate sensor nodes for target localization [35]. Bian et.al. has proposed a utility-based sensor selection framework [36]. In this paper, a utility function is used to locate multimedia sensor nodes for maximum coverage and minimum cost on a 3D terrain. To the best of our knowledge, this is the first paper that uses utility function theory for WMSN coverage and cost for 3D terrains.

The key points of MAUA are the designs and the weights of utility functions. According to the most common risk attitudes, there are three main types of utility functions: risk-seeking, risk-averse and risk-neutral [33].

In our study, the coverage maximization and the cost minimization are the two objectives or attributes that are conflicting each other. To test the capability of the algorithm on various scenarios, we present two alternative objective functions:

- Risk Neutral
- Risk Averse

3.1.1. Risk Neutral Objective Function

A novel multi-attribute utility model, which is presented in [37], is adapted to our problem (see Equation 9).

$$f_1 = w_v * U_{cov} + w_c * U_{cost} + w_v * w_c * U_{cov} * U_{cost} \quad (9)$$

In this equation, w_v and w_c are the weights of coverage and cost utility functions, respectively, where $0 < w_v, w_c < 1$ and $w_v + w_c = 1$. These weights are set based on the experimentation on a given terrain by considering various missions. While the first and the second terms are very similar to weighted sum method, the third term represent the objective interactions.

3.1.2. Risk Averse Objective Function

In Equation 10, risk averse objective function aims to raise the coverage rate of a terrain by increasing the number of sensors in the network. The cost function in Equation 11 lowers the weight of cost attribute than the cost function in risk neutral optimization function. By Equation 12, coverage rate dominates the optimization function (especially where $U_{cov} < 0.60$). So, the risk averse objective function can be more effective for coverage weighted missions.

$$f_2 = w_v * f_v + w_c * f_c + w_v * w_c * f_v * f_c \quad (10)$$

$$f_c = 1 - e^{-U_{cost}} \quad (11)$$

$$f_v = \ln(U_{cov}) \quad (12)$$

3.2. Initial Population Generation

The first phase to generate a solution for initial population is to determine the number of sensors considered in the solution, which is set randomly between the upper and the lower limit of the sensor quantity.

To determine the limits of the sensor quantity, the terrain is separated into a number of regions whose length of each side is equal to the maximum or minimum depth of view of the sensors. The number of regions generated determines the sensor quantity limits. In calculation of the upper limited of the sensor quantity, minimum depth of view is considered. Likewise, maximum depth of view is considered to calculate the lower limited of the sensor quantity. These limits can change with respect to the complexity of the terrain by multiplying the number of regions with constants μ_1 and μ_2 , where $0.5 \leq \mu_1 \leq 1.25$ and $1.0 \leq \mu_2 \leq 1.75$ for lower and upper limits respectively.

After the sensor quantity of a solution is determined, the type of each sensor is set randomly by preserving the inverse proportionality to sensor capacities (i.e., the view ranges with respect to given sensor mode). Therefore, a given solution will have low capacity (so low cost) sensors with high probability and it will have high capacity (so high cost) sensors low probability. Then, locations of sensors in each solution are determined by a heuristic method (explained below) for a predefined percentage of the initial population; and the locations are set randomly for the remaining part of the initial population.

Our heuristic approach aims to distribute the selected sensors randomly with keeping the communication distance upper limit for at least one sensor in the solution. First, one sensor is located on the terrain randomly. Then the second sensor is located randomly on the communication range of the first sensor. For k^{th} sensor, the location is determined randomly on the range of at least one sensor which is already located in the solution. This procedure is applied for all sensors in the solution. It should be noted that each sensor is located on the center of mass of each selected polygon. Finally, the heading angle (θ) and the tilt angle (σ) of each sensor given in the solution of initial population are determined randomly by considering a feasible range of angles.

3.3. Connectivity Based Crossover Operator

In our study we consider one crossover operator, which is Connectivity-Based Crossover. It should be noted that the crossover operator considers problem-specific knowledge and generates a single offspring as the output.

This operator is similar to the original single-point crossover operator proposed in the literature [25]. A crossover point is selected randomly from the range $[1..m - 1]$ where m is the minimum of number of sensors exist in two parents. Then, both parents are splitted at the given crossover point, and two children are created by exchanging the tails. The better one of these two children is selected as offspring. After that, test for connectivity is applied on the selected offspring. If the offspring is not feasible, one sensor is attempted to insert into the solution to make the the solution connected. In insertion, all sensors in the eliminated child solution are tested in order to make the solution feasible until achieving and only one of the sensors is inserted if it makes the solution feasible. In Figure 5, sensor s_m establishes the connection among three connected sensor groups after crossover operation. In this operator, there is no upper bound for number of sensors considered; although this operator generates the output of the crossover operator.

3.4. Mutation Operators

Mutation operators in our study can be considered in two categories, which are the locational mutation operators, and the angular mutation operators. In our experimental study, we can either select one of the two categories randomly at the first mutation and do not change the category during the execution, or select the category randomly at each mutation case.

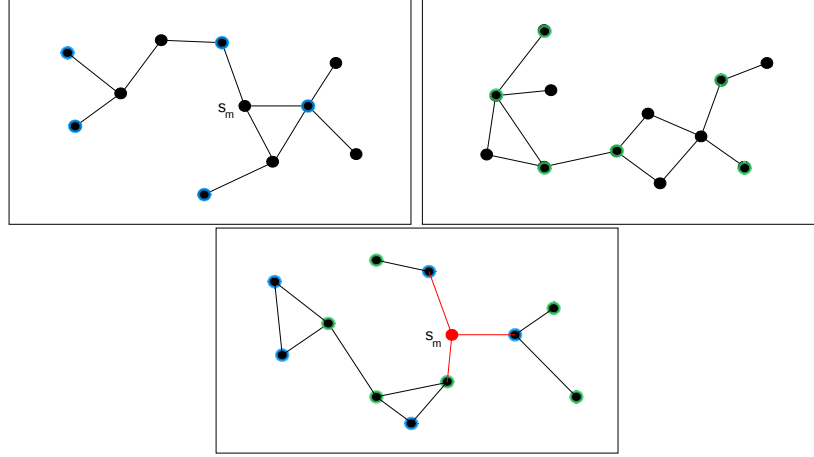


Figure 5: Connectivity Based Crossover

3.4.1. Polygon Based Mutation Operators

This group deals with only the locations of sensors (i.e. the polygon numbers on which they are located). There are three different mutation operators in this group and one of them is selected randomly if this group is selected for mutation. The mutation operators are:

- *Move Sensor.* Updating the location of a sensor is performed by selecting the sensor and selecting the new location of the selected sensor. Sensor selection procedure includes two alternatives: selecting randomly or selecting the one which has minimum coverage value. The new location of selected sensor is determined by either selecting randomly from the whole terrain or selecting randomly from the neighbourhood of the selected sensor.
- *Remove Sensor.* This operator removes one of the sensors, which can be either a randomly selected sensor or the one which has the minimum coverage value.
- *Add Sensor.* A new sensor is inserted to the solution on a polygon that is randomly selected either from the whole terrain or from the neighbourhood the sensor with the minimum coverage value. It should be noted that sensor type and heading and tilt angles of the new sensor is set randomly by considering the feasibility ranges of angles.

3.4.2. Angle Based Mutation Operators

This group deals with updating the angular attributes of a sensor, which is either selected randomly or selecting the sensor that has minimum coverage value. In this group, there are two operators and one of them is selected randomly if this group is selected for mutation. The mutation operators are:

- *Heading Angle Mutation.* It updates the heading angle of the selected sensor randomly.
- *Tilt Angle Mutation.* It updates the tilt angle of the sensor randomly in feasible range.

One extension is to select one of the two mutation operators from this class non-uniformly. Specifically, mutation on the heading angle can be selected with a higher probability than mutation on the tilt angle.

3.5. Repairing Phase

Repair function affects the feasibility level of solutions and tries to make infeasible solutions become feasible by inserting new sensor between two disconnected sensor groups. This method is used only for infeasible solutions.

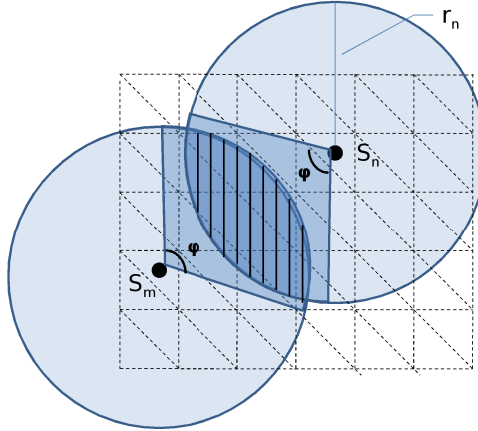


Figure 6: Repair Function

Firstly, sensors which are connected to at least one sensor are grouped as connected. Then, the distance between any two sensors from both connected sensor group and disconnected sensors is computed. Two sensors whose

distance between each other is minimum are selected. Then all polygons seen by both sensors which guarantee the connectivity threshold (κ) are listed by searching on possible viewing area of φ angle (see Figure 6). And one of these polygons is selected randomly as the location of new sensor. Finally, the heading angle and tilt angle of the new sensor is determined randomly by considering angle limits. If there is not any polygon seen by selected two sensors, the repair function is not applied on that iteration.

4. Experimental Study

In this section, we present the results of computational experiments in order to evaluate the effectiveness of our genetic algorithm (GA).

4.1. General Settings of the Experimentation

The genetic algorithm of our framework is coded in C programming language and the computational experiments are conducted on a cluster of machines, each of which has an Intel Xeon 2.33 Ghz. processor running the Linux operating system. The default values of general parameters are as follows: Coefficients of limiting the number of sensors in initial population are set to 1.25 (μ_1) and 1.75 (μ_2), respectively. The percentage of sensors allocated on a terrain region (center loc. vs. random loc.) is set to (20%, 80%). The tournament and population sizes are determined as 5 and 50. Also, the communication range and threshold value of a sensor are set to 400 and 0.05, respectively.

4.2. Performance Evaluation and Discussion

The set of experimental design is for identifying both the values of some algorithm-specific parameters (i.e., strategy for selecting the initial population, the type of mutation operator, the sensor selection criteria in mutation, and the type of risk function) and the problem-specific parameters (i.e., terrain type, utility weights for determining a specific mission). Table 1 presents the set of values considered for the parameters.

The Effects of Parameter Groups. In this part, we present the effectiveness of our algorithm for two distinct parameter groups (i.e., parameters for risk neutral objective function (Type I) and parameters for risk averse objective function (Type II)) on both smooth and rough terrains. Parameters used in

Initial Population Selection Strategy	Mutation Operator	Source Sensor in Mutation	Utility Weights	Terrain	Risk Function
(50% Heuristic, 50% Random)	Location-Based	Random Sensor	(0.5,0.5)	T1 (smooth)	Risk-Neutral
100% Random	Angular-Based	Sensor with min. Visibility	(0.6,0.4)	T2 (smooth)	Risk-Averse
	Random		(0.7,0.3)	T3 (rough)	
			(0.8,0.2)	T4 (rough)	

experiments for both types of experimentation are given before in Subsection 4.1.

Since the sensors have limited ranges, a steep slope on the view cone of a sensor decreases the visibility range of this sensor. Therefore, when comparable number of sensors are utilized on both terrains, the total utility value observed on a smooth terrain is higher than the total utility value on a rough terrain (see Tables 2 & 3).

Parameter Group	Occlusion	Algorithm	Number of Sensors	Total Utility	Connectivity Ratio
Type I	Occluded	RS	34.4	0.4715	0.3353
		GA	24.5	0.5774	0.9767
	Non-Occluded	RS	35.0	0.4915	0.3333
		GA	26.1	0.5916	0.9960
Type II	Occluded	RS	84.2	0.3561	0.3460
		GA	85.1	0.4333	1.0000
	Non-Occluded	RS	83.0	0.3852	0.3440
		GA	85.0	0.4633	1.0000

The performance comparison of the GA with a random search method (RS) is presented for all parameter groups and occlusion option pairs in Tables 2 and 3. The best individual in a randomly generated initial population of the GA with 50 individuals is the output of the RS method for each test. The total utility of the solutions generated by the GA are significantly outperforms those of the RS method for both smooth and rough terrains.

In addition to the comparisons based on the total utility function, we also provide the performance comparison of the algorithms with respect to percentage of the polygons recognized by sensors and they are presented in Table 4 for a smooth terrain (related with the tests of Table 2) and in Table 5

Table 3: Performance Results on a Rough Terrain with Different Parameter Groups

Parameter Group	Occlusion	Algorithm	Number of Sensors	Total Utility	Connectivity Ratio
Type I	Occluded	RS	39.4	0.4515	0.2852
		GA	23.3	0.5702	0.7786
	Non-Occluded	RS	37.0	0.4679	0.3000
		GA	24.7	0.5817	0.9297
Type II	Occluded	RS	83.5	0.3352	0.2753
		GA	86.6	0.4175	1.0000
	Non-Occluded	RS	83.0	0.3554	0.3133
		GA	85.9	0.4422	1.0000

for a rough terrain (related with the tests of Table 3). It should be noted that each terrain is represented by a set of triangular polygons and every terrain is generated with fixed number of polygons, which is 32768.

Table 4: Performance Comparison on a Smooth Terrain with respect to Coverage

Parameter Group	Occlusion	Algorithm	Covered Polygons (%) when $C(S, p_i)$		
			≥ 0.1	≥ 0.5	≥ 0.75
Type I	Occluded	RS	0.32	0.17	0.05
		GA	0.47	0.28	0.07
	Non-Occluded	RS	0.36	0.23	0.07
		GA	0.51	0.34	0.08
Type II	Occluded	RS	0.65	0.40	0.14
		GA	0.81	0.53	0.19
	Non-Occluded	RS	0.65	0.46	0.17
		GA	0.81	0.62	0.22

The percentage of the number of polygons covered is given for three different minimum visibility ratios (0.1, 0.5, 0.75) in the tables. Specifically, the column for $C(S, p_i) \geq 0.1$ gives the percentage of the polygons that are at least 10% visible by the sensors deployed. Our algorithm significantly surpasses the RS method for all different parameter groups studied. As a specific example, when detection mode with Type I parameter group and occlusion is considered, our algorithm observes up to 65% (i.e., $(0.28 - 0.17)/0.17$) more number of polygons on a smooth terrain and up to 47% (i.e., $(0.25 - 0.17)/0.17$) more number of polygons on a rough terrain. As an example, the coverage map of the sensors on a rough terrain is depicted in Figure 7.

Table 5: Performance Comparison on a Rough Terrain with respect to Coverage

Parameter Group	Occlusion	Algorithm	Covered Polygons (%) when $C(S, p_i)$		
			≥ 0.1	≥ 0.5	≥ 0.75
Type I	Occluded	RS	0.34	0.17	0.05
		GA	0.45	0.25	0.06
	Non-Occluded	RS	0.35	0.21	0.06
		GA	0.48	0.30	0.07
Type II	Occluded	RS	0.62	0.36	0.12
		GA	0.78	0.51	0.17
	Non-Occluded	RS	0.62	0.41	0.13
		GA	0.80	0.58	0.19

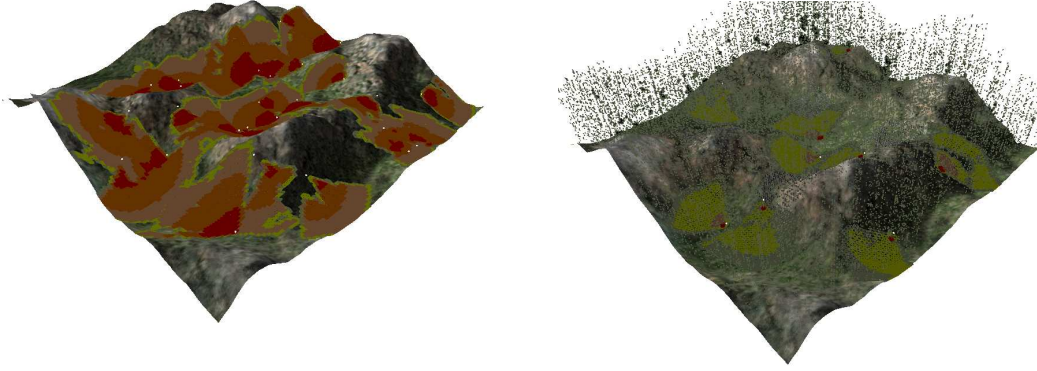


Figure 7: Sensor Coverage Examples: (a) Without Occlusion, and (b) Weather and Object Occlusions Included

Performance comparison of the algorithms with respect to different communication thresholds (i.e. 0.10, 0.25, 0.50) is presented in Table 6 for a smooth terrain and in Table 7 for a rough terrain. Increase of the communication thresholds can cause to reduce the number of communications links among the sensors in a network. Depending on this, the network can become unconnected. However, the results on Table 6 and Table 7 show that the performance of the algorithm is not affected very much despite the increase communication thresholds. So, the robustness of our algorithm is proved for different communication thresholds.

5. Conclusions

WMSN, which can retrieve multimedia data from the environment and send the data to a sink, has its own unique design considerations, like con-

Table 6: Performance Results on a Smooth Terrain with Different Communication Thresholds

Parameter Group	Occlusion	Communication Threshold	Number of Sensors	Total Utility	Connectivity Ratio
Type I	Occluded	0.10	24.2	0.5785	0.9920
		0.25	24.8	0.5808	0.9873
		0.50	24.7	0.5808	0.9938
	Non-Occluded	0.10	25.9	0.5896	0.9693
		0.25	26.3	0.5910	0.9947
		0.50	25.6	0.5897	0.9953
Type II	Occluded	0.10	86.6	0.4336	1.0000
		0.25	84.5	0.4307	1.0000
		0.50	85.6	0.4302	1.0000
	Non-Occluded	0.10	85.1	0.4685	1.0000
		0.25	84.2	0.4671	1.0000
		0.50	86.3	0.4639	1.0000

Table 7: Performance Results on a Rough Terrain with Different Communication Thresholds

Parameter Group	Occlusion	Communication Threshold	Number of Sensors	Total Utility	Connectivity Ratio
Type I	Occluded	0.10	25.4	0.5737	0.9784
		0.25	25.8	0.5743	0.9600
		0.50	24.8	0.5712	0.8990
	Non-Occluded	0.10	26.0	0.5779	0.9569
		0.25	26.4	0.5819	0.9564
		0.50	26.5	0.5802	0.9500
Type II	Occluded	0.10	84.9	0.4204	1.0000
		0.25	85.9	0.4186	1.0000
		0.50	86.5	0.4174	1.0000
	Non-Occluded	0.10	85.6	0.4374	1.0000
		0.25	85.4	0.4408	1.0000
		0.50	86.4	0.4377	1.0000

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

nected coverage. Sensor node deployment affects the connected coverage performance of WMSN, especially for 3D terrain structure. Sensor nodes should be located and utilized so that the coverage of the sensors is maximized and the connectivity must be preserved. In this paper, multimedia sensor node location and utilization problem is investigated to maximize coverage and to minimize cost while preserving the connectivity on 3D occluded terrain.

In order to formulate the multiobjective optimization (maximize coverage and minimize cost), a novel global utility function is proposed to unify the utility of the coverage and the utility of the cost with MAU analysis. According to risk attitudes of the system, two different global utility functions are defined. These are risk neutral and risk averse functions. While risk neutral function represents equivalent worth of cost and coverage, risk averse function represent the dominance of coverage over cost especially for lower coverage rates.

The global utility function is optimized by the help of newly proposed GA methods under the connectivity constraint. Our GA algorithm includes new methods for initial population, crossover (ie., connectivity based) and mutation (i.e., polygon based and angle based) operators. In generation of initial population, problem-specific heuristics are considered to get feasible solutions. A repair phase is also designed to increase feasible solutions subject to connectivity constraint. In this phase, the repair function injects new nodes to the solution graph in order to satisfy the connectivity constraint, if the solution is infeasible.

Experimental studies show that 3D structure and the occlusions affect the performance of WMSN directly. In addition to this, the new MAU based utility function and the novel GA methods can ensure a good balance between cost and coverage. Another performance comparison is presented based on the percentage of the covered polygons. Performance result comparisons show that new GA algorithm can find better solution not only for the global utility function, but also for the percentage of the covered poygon. The last experiment set is performed to illustrate the effects of the communication threshold. It is also shown that the performance of the algorithm is not affected very much despite the increase communication thresholds.

References

- [1] I.F. Akyildiz, T. Melodia, and K.R. Chowdhury, A survey on wireless multimedia sensor networks, *Computer Networks*, 51 (4) (2007) pp.

921-960.

- [2] D.Chen and P.K. Varshney, QoS support in wireless sensor networks: A survey, in: Proceedings of the 2004 International Conference on Wireless Networks (ICWN 2004), Las Vegas, Nevada, USA, June 2004.
- [3] D.W. Gage, Command control for many-robot systems, in: Nineteenth Annual AUVS Technical Symposium, Reprinted In Unmanned Systems Magazine, 10 (4) (1992) pp. 28-34.
- [4] P. Hall, Introduction to the theory of coverage processes, John Wiley and Sons, 1988.
- [5] J. O'Rourke, Art gallery theorems and algorithms, Oxford University Press, New York, 1987.
- [6] J. O'Rourke, Computational geometry column, International Journal of Computational Geometry and Applications, 2 (2) (1992), pp. 215-217.
- [7] M. Marengoni, B.A. Draper, A. Hanson, R. Sitaraman, A system to place observers on a polyhedral terrain in polynomial time, Image and Vision Computing, 18 (1996), pp. 773-780.
- [8] K.A. Tarabanis, P.K. Allen, and R.Y. Tsai, A survey of sensor planning in computer vision, IEEE Transactions on Robotics and Automation, 11 (1) (1995), pp.86-104.
- [9] A. Mittal, and L.S. Davis, A general method for sensor planning in multi-sensor systems: extension to random occlusion, Int. J. Comput. Vision, 76 (1) (2008), pp. 31-52.
- [10] E. Hrster, and R. Lienhart, On the optimal placement of multiple visual sensors, in: Proceedings of the 4th ACM international Workshop on Video Surveillance and Sensor Networks (VSSN '06) Santa Barbara, California, USA, October 2006, pp. 111-120.
- [11] Jian Wang, Changyong Niu, Ruimin Shen, Priority-based target coverage in directional sensor networks using a genetic algorithm, to appear in Computers and Mathematics with Applications, 2008.

- [12] U.M. Erdem, S. Sclaroff, Optimal Placement of Cameras in Floorplans to Satisfy Task Requirements and Cost Constraints, in: Proceedings of The fifth Workshop on Omnidirectional Vision, Camera Networks and Non-classical cameras (Omnivis2004), May 2004.
- [13] N. Tezcan, and W. Wang, Self-orienting wireless multimedia sensor networks for occlusion-free viewpoints, Computer Networks, 52 (13) (2008), pp. 2558-2567.
- [14] H.R. Topcuoglu, M. Ermiş, and M. Sifyan, Positioning and Utilizing Sensors on a 3D Terrain - Part II: Solving with an Hybrid Evolutionary Algorithm, submitted to IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Review.
- [15] A. Ghosh, and S.K. Das, Coverage and Connectivity Issues in Wireless Sensor Networks: A Survey, Pervasive and Mobile Computing, 4 (3) (2008), pp. 303-334.
- [16] F. Ye, G. Zhong, S. Lu, and L. Zhang, PEAS: A Robust Energy Conserving Protocol for Long-lived Sensor Networks, The 23rd International Conference on Distributed Computing Systems (ICDCS'03), May 2003.
- [17] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill, Integrated Coverage and Connectivity Configuration for Energy Conservation in Sensor Networks, ACM Trans. on Sensor Networks, 1 (1) (2005), pp. 36-72.
- [18] Z. Zhou, S.R. Das, and H. Gupta, Connected K-Coverage Problem in Sensor Networks, in: Proc. of the 13th Int. Conf. on Computer Communication and Networks (IC3N), Chicago, IL, Oct 2004.
- [19] G. Yang, and D. Qiao, Critical Conditions for Connected-k-Coverage in Sensor Networks, IEEE Communications Letters, 12 (9) (2008).
- [20] D. Pompili, T. Melodia, and I. Akyildiz, Three-Dimensional and Two-Dimensional Deployment Analysis of Underwater Acoustic Sensor Networks, Ad Hoc Networks, 3 (3) (2005), pp. 257-279.
- [21] Xiaofeng Han, Xiang Cao, Errol L. Lloyd and Chien-Chung Shen. Deploying Directional Sensor Networks with Guaranteed Connectivity and Coverage, IEEE SECON 2008, San Francisco, CA, June 16-20, 2008.

- [22] Bob Nystrom, Terrain Generation Tutorial: Hill Algorithm, 2000.
- [23] A. Fournier, D. Fussel, and L. Carpenter, Computer Rendering of Stochastic Models, Communications of the ACM, 25 (6) (1982), pp. 371-384.
- [24] Z. Michalewicz, and D. B. Fogel, How to Solve It: Modern Heuristics, Springer Verlag, 2000.
- [25] A. E. Eiben and J. E. Smith, Introduction to Evolutionary Computing, Springer Verlag, 2003.
- [26] B. R. Fox and M. B. McMahon, Genetic Operators for Sequencing Problems, In G. Rawlins (Ed.), Foundations of Genetic Algorithms, First Workshop on the Foundation of Genetic Algorithms and Classifier Systems, Morgan Kaufmann, 1990.
- [27] J. J. Greffentette, Incorporating Problem Specific Knowledge into a Genetic Algorithm, in Genetic Algorithms and Simulated Annealing, L. Davis (Ed.), Morgan Kaufmann Publishers, 1987.
- [28] G. Raidl, Hybrid Evolutionary Algorithms for Combinatorial Algorithms, Habilitation Thesis, Vienna University of Technology, 2002.
- [29] P. Moscato, On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms, Tech. Rep. Caltech Concurrent Computation Program Report 826, Caltech, 1989.
- [30] N. J. Radcliffe and P. D. Surry, Formal Memetic Algorithms, Evolutionary Computing: AISB Workshop, Springer-Verlag LNCS 865, pp. 1-16, 1994.
- [31] J. Andersson, Multiobjective Optimization in Engineering Design - Applications to Fluid Power Systems, Dissertation, Linkping studies in science and Technology, Dissertation No. 675, Linkping University, Linkping, Sweden, 2001.
- [32] O.L. De Weck, Multiobjective Optimization: History and Promise, in: Third China-Japan-Korea Joint Symposium on Optimization of Structural and Mechanical Systems (CJK-OSM3) Kanazawa, Japan, October-November 2004.

- 1
2
3
4
5
6
7
8
9 [33] R.L. Keeney, and H. Raiffa, Decisions with Multiple Objectives: Pref-
10 erences and Value Tradeoffs, Cambridge University Press, 1993.
11
12 [34] J. Byers, and G. Nasser, Utility-based decision-making in wireless sensor
13 networks, in: Proceedings of the 1st ACM international Symposium on
14 Mobile Ad Hoc Networking and Computing, Boston, Massachusetts,
15 2004, pp. 143-144.
16
17 [35] V. Isler, and R. Bajcsy, The sensor selection problem for bounded
18 uncertainty sensing models, in: Proceedings of the 4th international
19 Symposium on information Processing in Sensor Networks, Los Ange-
20 les, California, April 2005, pp. 24-27.
21
22 [36] F. Bian, D. Kempe, and R. Govindan, Utility-based sensor selection,
23 in: Proceedings of The Fifth International Conference on Information
24 Processing in Sensor Networks (IPSN 2006), Nashville, TN, USA, April
25 2006, pp. 11-18.
26
27 [37] H. Topcuoglu, M. Ermiş and M. Sifyan, Hybrid Evolutionary Algorithm
28 for Sensor Placement on a 3-D Terrain, in: Proceedings of IEEE Interna-
29 tional Conference on Intelligent Systems Design and Applications, Italy,
30 2009.
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

Haluk Rahmi Topcuoglu received the B.Sc. and M.Sc. degrees in computer engineering from Bogazici University, Bogazici, Istanbul, in 1991 and 1993, respectively, and the Ph.D. degree in computer science from Syracuse University, Syracuse, NY, in 1999. He is an Associate Professor in the Computer Engineering Department, Marmara University, Istanbul, Turkey. His main research interests include task scheduling and mapping for parallel and distributed systems, parallel computing, evolutionary algorithms, and evolutionary computing for dynamic environments. Dr. Topcuoglu is a member of the IEEE Computer Society and the Association for Computing Machinery (ACM).

Murat Ermis received the B.Sc. and M.Sc. degrees in industrial engineering from Bogazici University, Istanbul, and Middle East Technical University, Ankara, respectively, and the Ph.D. degree in industrial engineering from Istanbul Technical University, Istanbul, in 2005. He is an Assistant Professor in the Industrial Engineering Department, Turkish Air Force Academy, Istanbul, Turkey. His main research interests include mathematical modeling, optimization, metaheuristics and scheduling.

Ilker Bekmezci is an instructor in the Dept. of Computer Eng., Turkish Air Force Academy, Istanbul. He received B.Sc., M.Sc., and Ph.D. degrees in Computer Engineering from Bogazici University, Istanbul, in 1994, 1998, 2002 respectively. His interesting areas are computer networks, wireless sensor networks, and wireless multimedia sensor networks.

Haluk Topcuoglu photo
[Click here to download high resolution image](#)



Murat Ermis photo
[Click here to download high resolution image](#)





Positioning and Utilizing Sensors on a 3-D Terrain

Part I - Theory and Modeling

Haluk Topcuoglu, *Member, IEEE*, Murat Ermis, and Mesut Sifyan

Abstract—Positioning multiple sensors for acquisition of a given environment is one of the fundamental research areas in various fields, such as military scouting, computer vision, and robotics. In this paper, we propose a new model for the problem of sensor deployment. Deploying and configuring a set of given sensors on a synthetically generated 3D terrain have multiple objectives on conflicting attributes: maximizing the visibility of the given terrain, maximizing the stealth of the sensors, and minimizing the cost of the sensors used. Since they are utility-independent, these complementary and conflicting objectives are modelled by a multiplicative total utility function, based on multi-attribute utility theory. The total utility function proposed in this paper can also be adapted for various military scouting missions with different characteristics.

Keywords—Sensor planning, multi-attribute utility theory, multiobjective optimization, viewpoints.

I. INTRODUCTION

ALTHOUGH remote sensing technologies have become widespread in various domains due to decrease in the cost of manufacturing more sophisticated sensing devices, gathering more accurate and comprehensive information is still a challenging problem. Since there are different types of sensors and sensing technologies which are used to search and detect targets with different costs, deployment of multiple sensors for the covering of a given environment, finding the most favorable orientation is an important and demanding combinatorial optimization problem [1], [2].

In military operations, reconnaissance, surveillance, and target acquisition can include a plurality of sensor platforms that are used to collect information about an area under surveillance and play a vital role [3]. In order to detect the position of foes, some sensors should be placed in certain terrains to provide maximum visibility while maintaining sensors' stealth.

In addition to military applications, sensor placement problem is a fundamental research area in various other domains. Determining the number of sensors necessary to cover a room or a region was originally posed in the computational geometry literature as the Art Gallery Problem [4]. In the traditional formulation, two points (observer and target points) are called visible if the straight line segment between these points lies entirely inside the view cone. Even with such simple definition

of the visibility, the problem is NP-complete. There are a number of studies in the literature on 2-D coverage problem.

The general camera placement problem is an old and well-known problem and it is defined with assumptions that are more consistent with the capabilities of real world cameras [5]. Given a floor plan to be observed, the problem is to efficiently compute a camera layout such that certain task-specific constraints are met with minimal camera setup cost. A solution to this problem is obtained via binary optimization over a discrete problem space. Han et al. propose the problems of deploying a minimum number of directional sensors to form a connected network to cover and offer two approximation solutions [6].

There have been various works on vision planning. An investigation of the coverage and orientation problem for video sensor networks is addressed [7]. They propose a new algorithm to find the most beneficial orientations for the sensors in 2-D case, while minimizing the effects of occlusion in the environment and improving the aggregate quality of the information sensed from the region of interest. The terrain structure highly affects the coverage performance of the sensor network. Most of the studies deal with 2-D terrain representation which is sufficiently accurate for indoor applications [8]. Nevertheless, a 2-D terrain representation does not satisfy the requirements of outdoor applications such as military scouting, border surveillance, etc. [9]. The 3-D video sensor coverage area is illustrated in Fig. 1. In this model, the coverage area is a cone.

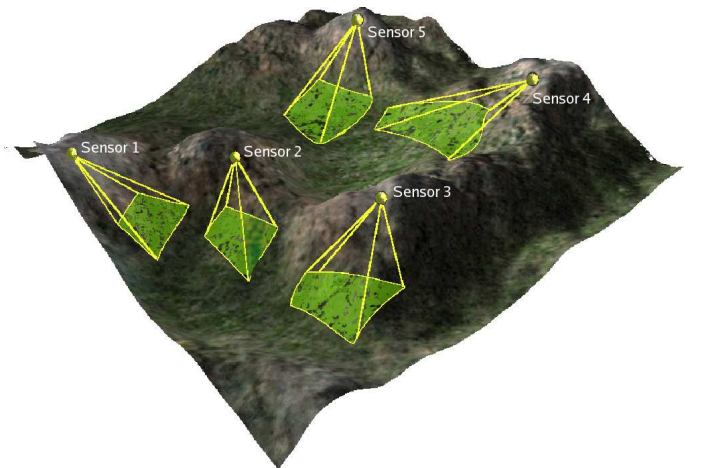


Fig. 1. Coverage area of sensors on a 3-D terrain

In another study, Zhao et al. propose a novel 3-D visibility model [10]. However, their approach is specifically designed for visual tagging problems and it can not be directly used for

Haluk Topcuoglu is with the Computer Engineering Department, Marmara University, Goztepe Kampusu, Kuyubasi, Istanbul, 34722, Turkey. email: haluk@marmara.edu.tr

Murat Ermis is with the Industrial Engineering Department, Turkish Air Force Academy, Yesilyurt, Istanbul, 34149, Turkey. email: m.ermis@hho.edu.tr

Mesut Sifyan is with the Computer Engineering Department, Marmara University, Goztepe Kampusu, Kuyubasi, Istanbul, 34722, Turkey. email: mesutsifyan@marun.edu.tr

coverage problems. Bakhtari et al. present an implementation of an active-vision-based surveillance system for target localization in online applications [11]. Mittal studied determining optimal sensor configurations by enhancing the analysis of scenarios where there are dynamically occurring objects capable of occluding each other [12], [13]. He proposed a generic framework for sensor planning with the integration of visibility and static constraints, which can be customized for a particular task involving randomly occurring objects. Mittal also studied the presence of random object occlusion in 3-D based visibility models [14]. But, this model does not take into account of other occlusion types, like weather occlusions or natural occlusions. Becker et al. introduce a discrete 3-D space modeling in [15]. In this model, it is assumed that the target space to be observed is the volume where a human being might navigate. But, it is not sufficient to model large scale outdoor applications for coverage issues. The problem of maintaining sensing coverage by a small number of sensors and low energy consumption in a wireless sensor network is also studied in the literature [1], [16].

Automatic selection of sensor viewpoints to view multiple objects and regions in a given 3-D scene is an important research area in computer vision [17]–[20]. In such tasks, a set of viewpoints and sensor placement must be planned to sample all features of interest. Similarly, for autonomous mobile robot navigation, a robot is equipped with a set of sensors which cooperate to detect the obstacles and the free space. Sensor planning in robotics, which is the process of deciding the types, configurations and tasks of the sensors, has been studied with various techniques [21]–[23]. Tharmarasa et al. propose an algorithm for the problem of dynamic assignments of sensors to local fusion centers (LFCs) in a distributed architecture by maximizing tracking performance of multiple targets [24]. They decompose the sensor management problem into a set of subproblems including selecting subsets of sensors, assign them to LFCs, and assigning the frequency and transmission power to each active sensor. A cooperative sensor planning system based on multi-attribute utility theory was recently presented, which unifies research from vision, sensor planning and multi-agent planning [25].

Although utility function theory has a well-known history in optimization [26], there are only a few utility theory studies in the sensor network context. Byers and Nasser proposed a utility-based sensor selection method in [27]. Isler and Bajscy developed a utility function-based model to locate sensor nodes for target localization [28]. Bian et al. proposed a utility-based sensor selection framework [29].

In this paper, we develop a novel formulation for determining the optimal number of sensors, locating and setting their orientational sensor-specific parameters in a synthetically generated 3-D terrain with multiple objectives. Our solution approach relies on the rational trade-off between three conflicting objectives which are: maximizing the coverage area while maintaining the maximum stealth, and minimizing the total acquisition cost of deploying the sensors. These objectives are utility-independent; and when multi-objective utility theory is considered, a multiplicative function over these attributes fits for the total utility function in our study. Therefore, we aim

at maximizing the total or the unified utility function that includes the utility of visibility, the utility of stealth and the utility of the cost.

The remainder of the paper is organized as follows: In Section 2, we present perception methods for sensor systems which are used to compute the cumulative visibility and stealth values of the region. We present our novel multi-attribute utility function and its sub-objective formulations in Section 3, and Section 4 concludes the paper.

II. PERCEPTION METHODS DEVELOPED FOR SENSOR SYSTEMS

In our study, sensor characteristics are represented in two sets of attributes, *locational attributes* and *behavioral attributes*. The locational attributes of a sensor are the positional characteristics that can change throughout the run. Specifically, the locational attributes of the sensor s_i are:

- the position on the terrain in terms of the polygon number, p_{s_i} , on which this sensor is located. Each sensor can be located only on the center of mass of the selected polygon.
- the heading angle, θ_i . This is the angle between the looking direction of the sensor carrier and X axis, which is between 0 and 359 degrees.
- the tilt angle, σ_i . This is the angle between sensor and its carrier which is in the range of $-90 + \beta^L \leq \sigma_i \leq 90 - \beta^U$, where β^L , β^U are the (vertical) lower and upper viewing angles of the sensor s_i .

The locational attributes of sensors are determined based on the selected mode of the sensor, which can be detection, identification, or recognition. The behavioral attributes specify the visibility capability of each sensor; and do not change throughout the run. The sensor s_i has the following behavioral attributes:

- depth of view of the sensor, Δ_i . This specifies the range of visibility for the given sensor.
- horizontal viewing angle, α_i . The viewing angle indicates the wideness of viewing area of the sensor.
- (vertical) lower and upper viewing angles, which are β^L and β^U , respectively. These angles bound the vertical viewing area of the given sensor.
- financial cost of the sensor, $Cost_F(i)$. This depends on the sensing ability of a sensor. Specifically, a sensor with a high depth of view and a large viewing angle is more expensive.
- type of the sensor, T_i . This is a number used for referencing a sensor with specific viewing parameters.

The visibility value of a polygon is set by computing the average visibility of the four points (the three corner points and the center of mass point) of the polygon. Checking each polygon one by one and computing the visibility of the four points on the selected polygon by considering all sensors located in the system is a computationally intensive operation. We present a cost-effective approach in our framework by considering only polygons within the view range of sensors. After all polygon points within the view cone of a selected sensor are determined, the perception procedure of the sensor

is handled in the three phases (given in the following three subsections). Then, the total visibility of the terrain is derived by unifying the results (i.e., the visibility of polygon points) of all sensors considered.

A. Phase 1 - Horizontal Field-of-View (FOV) Test

This phase is applied to those polygons that lie in the broadest field of view; and a region on the terrain (in terms of a set comprising polygons) that passes the horizontal FOV test is determined. The main purpose of this test is to provide a proper filtering for the XZ axis (without considering the Y axis). A subset of locational attributes (the polygon number to represent its location and the heading angle), a subset of the behavioral attributes (the depth of view and the horizontal viewing angle) and the type of the sensor are the required inputs for the filtering function of the horizontal FOV test (see Fig. 2).

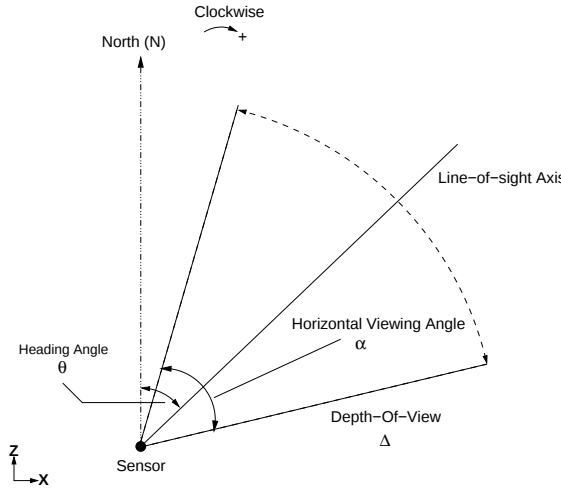


Fig. 2. Attributes Considered for Horizontal FOV Test

Figure 3 is the XZ plane of a sample terrain which is visualized as polygons. In this figure, sensor is placed on point O; the heading angle is $\theta = \angle ZOB$, and the horizontal viewing angle is $\alpha = \angle AOC$. The steps of FOV procedure (by considering Fig. 3) follow:

- *Calculating angles required in the tests.* The $\angle XOB$ angle is calculated through the equation $\angle XOB = (450 - \theta) \bmod 360$. Then, $\angle XOA$ and $\angle XOC$ angles are calculated similarly through the equations given below:

$$\begin{aligned}\angle XOA &= (\angle XOB + \frac{\alpha}{2}) \bmod 360 \\ \angle XOC &= (\angle XOB - \frac{\alpha}{2}) \bmod 360\end{aligned}\quad (1)$$

- *Calculating the X and Z Coordinates of A, B and C points.* By taking left bottom point of the Fig. 3 (marked with R where R is the reference point) as the (0,0) point, coordinate calculations of A, B, and C points are conducted. Here, $|OB|$ is the depth of view of the sensor, $\Delta = |OB|$. It is also known on which polygon the sensor

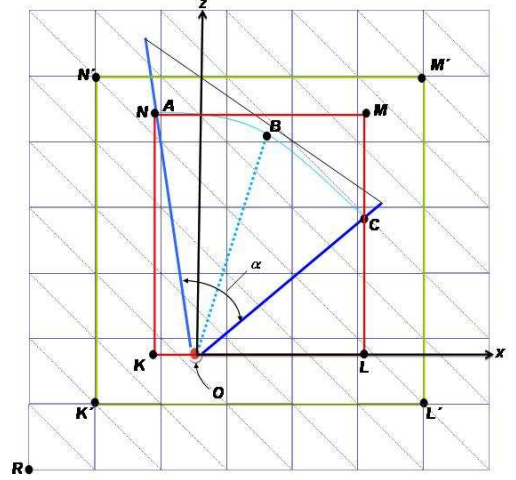


Fig. 3. A Sample Terrain for the Horizontal FOV Test

is located; and the X and Z coordinates of the point O are basically the coordinates of center of mass of the given polygon. The coordinates of point B are computed by:

$$\begin{aligned}B.x &= O.x + (|OB| \times \cos(\angle XOB)) \\ B.z &= O.z + (|OB| \times \sin(\angle XOB))\end{aligned}\quad (2)$$

Since the line-of-sight axis (OB line) is bisector of the viewing angle, $|OB| = |OA| = |OC|$, the X and Z coordinates of A and C points are computed with the similar equations.

- *Forming a Bounding Rectangle.* The minimum and the maximum values for X and Z coordinates of O, A, B, and C points are calculated; the combination of these points form a rectangle (say KLMN rectangle). It should be noted that one or more of these points can not be corner points (of polygons). Then, these points are extended to the nearest polygon points so that they will form a new rectangle ($K'L'M'N'$) so that the new rectangle will bound and include the KLMN rectangle. In this step, it is provided that the corners of the new rectangle should be within the terrain.
- *Applying Horizontal FOV Test to the Points in the New Rectangle:* This step is for determining whether the corner points and center of mass point of all polygons (that are in the rectangle area constructed in the previous step) are in the horizontal view-cone of the sensor or not. For every point considered, it is checked whether the point is in either the AOB triangle or BOC triangle. It should be noted that if we consider the AOC triangle instead of the AOB and BOC triangles, we lose some points from the horizontal FOV area.

B. Phase 2 - Vertical Field-Of-View (FOV) Test

The list of terrain points that are perceived from the horizontal FOV test will be input for the vertical FOV test. The upper viewing angle, the lower viewing angle and the tilt angle of the given sensor are the main attributes that will be considered

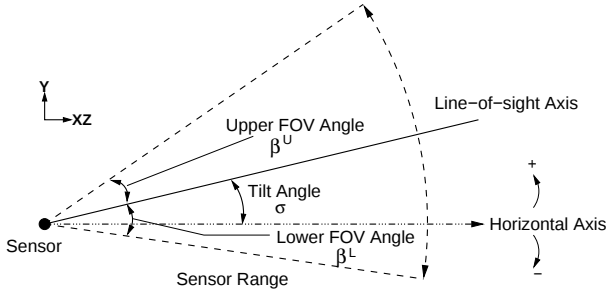


Fig. 4. Attributes Considered for Vertical FOV Test

in vertical FOV test (see Fig. 4). In this test, the first step is to calculate the slope of the sensor. By using σ , β^U , and β^L values, the slope of upper FOV (given in ρ^U) and slope of lower FOV (given in ρ^L) are calculated through the following equation:

$$\rho^U = \tan(\sigma + \beta^U) \quad , \quad \rho^L = \tan(\sigma - \beta^L) \quad (3)$$

For the case of $\tan(90)$ and $\tan(-90)$, an infinitesimal value (0.05) is added to the angle value. Then, for each point to which the vertical FOV test is applied, the slope between the sensor and the selected point is calculated. If this slope (α) satisfies the following inequality, the point passes the vertical FOV test:

$$\min(\rho^U, \rho^L) < \tan(\alpha) < \max(\rho^U, \rho^L) \quad (4)$$

The output of this test (in terms of polygon points) is the view-cone of the given sensor.

C. Phase 3 - Line-of-Sight (LOS) Algorithm

In this phase, a LOS algorithm is applied to compute the visibility value of each polygon within the view-cone from the previous phase. A LOS algorithm finds out whether any point on the terrain blocks one or more rays going from a source sensor to a target point [30]. In order to determine whether the target point is occluded or not, an *evaluation set* is chosen among the different points of the 3-D ray going from the sensor towards to the target. For each (x, z) point in this set (without taking into account the height dimension in a XZ plane), the height of the terrain at that point is compared with the height of the ray at the same point. If the terrain is higher at the given point, it is concluded that the ray is blocked and other points in the evaluation set are not considered; otherwise, the next point from the evaluation set is considered for the same test.

In our study, we implement two different LOS algorithms from the literature, the Janus Algorithm [31] and Bresenham Algorithm [30]. The first algorithm considers the interpolation strategy in order to determine the evaluation set of points, and the second one is based on the nearest-post strategy. Since the Bresenham Algorithm does not require computations for interpolation, and it may require less number of control points, it is faster but less accurate than the Janus Algorithm.

III. A NOVEL MULTI-ATTRIBUTE UTILITY MODEL

In this section, we present our novel multi-attribute utility function formulation and its sub-objective formulations for the sensor optimization problem. Multi-attribute utility analysis is a widely-accepted scalarization-based multiobjective optimization method especially in military applications due to the testing capability of various scenarios (such as risk averse, risk prone, etc.) under uncertainty or limited data cases [26], [32], [33]. In our study, there are three objectives or attributes that are both conflicting and, at the same time, supporting of each other, as a multiplicative function over the given three attributes. In the first step, the single-criterion utility function value, whose range is between 0 and 1, is calculated for each attribute. Then, the multi-attribute utility (MAU) function calculates the effect of each measurement on the total performance according to their determined weights.

In our study, the total function to maximize, $U(A, S, P)$, of scanning an area A using a set S of sensors which are located on a set P of polygons (i.e. the sensor s_i located on the polygon p_k), is represented with the following formula:

$$\begin{aligned} U(A, S, P) = & w_{vis} U_{vis}(A, S, P) + w_{st} U_{st}(A, S, P) + \\ & w_{cost} U_{cost}(S, P) + \\ & w_{vis} w_{st} U_{vis}(A, S, P) U_{st}(A, S, P) + \\ & w_{vis} w_{cost} U_{vis}(A, S, P) U_{cost}(S, P) + \\ & w_{st} w_{cost} U_{st}(A, S, P) U_{cost}(S, P) + \\ & w_{vis} w_{st} w_{cost} U_{vis}(A, S, P) U_{st}(A, S, P) U_{cost}(S, P) \end{aligned} \quad (5)$$

where $U_{vis}(A, S, P)$ is the utility of visibility of area A by the set of sensors S located on the set of polygons P , $U_{st}(A, S, P)$ is the utility of stealth of the set of sensors S located on set of polygons P , and $U_{cost}(S, P)$ is the utility of the cost of the sensors S located on set of polygons P . In this equation, w_{vis} , w_{st} , w_{cost} are the weights (coefficients) of visibility, stealth and cost utility functions, respectively, where $0 < w_{vis}, w_{st}, w_{cost} < 1$ and $w_{vis} + w_{st} + w_{cost} = 1$. These weights are set based on experimentation on a given terrain by considering various military scouting missions. The following subsections briefly describe how the utility functions are defined.

Additionally, we also consider the total utility of each sensor s_j located on polygon p_k in our computations, which is represented by $U^j(A, s_j, p_k)$. By using (5), this term requires $U_{vis}^j(A, s_j, p_k)$, $U_{st}^j(A, s_j, p_k)$ and $U_{cost}^j(s_j, p_k)$ terms, which are the sensors-specific utility of visibility, stealth, and cost, respectively.

A. Computing the Utility of Visibility

The value of utility of visibility is derived by using the amount of visibility of the given terrain, which is computed by adding the visibility of all polygons on the given terrain. Formally, the utility of visibility of area A by the set of sensors S (located at set of polygons P), $U_{vis}(A, S, P)$ is computed using (6):

$$U_{vis}(A, S, P) = \frac{\sum_{p_i \in A} V(S, P, p_i) \times W_{p_i}}{\sum_{p_i \in A} W_{p_i}} \quad (6)$$

where, W_{p_i} is the weight of the polygon p_i and $V(S, P, p_i)$ is the visibility value of polygon p_i by using the set of sensors S located on the set of polygons P , which is computed by (7):

$$V(S, P, p_i) = \frac{\sum_{b \in p_i} \max_{s_j \in S} \{V_{s_j}(s_j, p_k, b)\}}{n} \quad (7)$$

The term $V_{s_j}(s_j, p_k, b)$ is the visibility of a single point b (which is part of polygon p_i) from the sensor s_j which is already located on the center of mass of polygon p_k . Since point b can be recognized by multiple sensors (with different visibility values), the maximum visibility of b is considered in (7). In this equation, the visibility of a polygon p_i is computed by the average of visibility of selected polygon points. In our study, three corner points and center of mass are considered as the selected polygon points, therefore, $n = 4$ in (7) and (9). Although the $V_{s_j}(s_j, p_k, b)$ term is also used in (9), this equation computes the visibility value of polygon p_i by using sensor s_j .

The weight of each polygon p_i , W_{p_i} , can be set with two different techniques in our experimental study:

- During terrain generation, one or more regions of the terrain (or even the entire terrain) can be selected; and the same weight (or importance) value is assigned to all polygons in a selected region, where the values are from a discrete set that consists of 5 values: $\{0, 0.25, 0.5, 0.75, 1.0\}$. Note that if a polygon is not selected, it has the default weight of 0.5.
- In the second approach, weights are set by considering the terrain characteristics. A function is already defined in order to set the weights of all polygons in the range (0..1) by considering the heights and slopes of the polygons. Specifically, a polygon that is among the highest polygons of the terrain and whose slope is higher than a threshold value is set with a very high weight value (i.e., which is very close to 1).

Since sensor-specific utility values are also required in our study, utility of visibility of each sensor is computed in addition to overall utility of visibility. The utility of visibility of a specific sensor s_j is computed by using (8), where $V_{s_j}(s_j, p_k, p_i)$ is the visibility of p_i by using sensor s_j that is already located on polygon p_k . It should be noted the utility of visibility value is not equal to the summation of sensor-specific utility of visibility values.

$$U_{vis}^j(A, s_j, p_k) = \frac{\sum_{p_i \in A} V_{s_j}(s_j, p_k, p_i) \times W_{p_i}}{\sum_{p_i \in A} W_{p_i}} \quad (8)$$

$$V_{s_j}(s_j, p_k, p_i) = \frac{\sum_{b \in p_i} V_{s_j}(s_j, p_k, b)}{n} \quad (9)$$

Additionally, a new term called the *visibility effectiveness* of a sensor is considered in our study. It is computed similar to (8), where the nominator of the new term is equal to the nominator of (8). On the other hand, the denominator of the new term is equal to the summation of the weights of the polygons that are only in the best sensor's horizontal field-of-view region by locating the best sensor on polygon p_k . Therefore, it differentiates from (8) where the denominator is

computed by summing the weights of all polygons within the whole terrain. Since, horizontal field-of-view of the best sensor basically shows the maximum visibility of a sensor, visibility effectiveness computation generates normalized results due to its denominator.

The term $V_{s_j}(s_j, p_k, b)$, the visibility of the destination point b from the sensor s_j located at source point a (which is the center of mass of polygon p_k), is formally defined by (10):

$$V_{s_j}(s_j, p_k, b) = (1 - \eta_{s_j} \times \frac{D(a, b)}{\Delta_{s_j}}) \times (1 - \max_{p_c \in a \rightarrow b} \psi_c^W) \times (1 - \max_{p_c \in a \rightarrow b} \psi_c^O) \quad (10)$$

where $D(a, b)$ is the distance between point a and point b ; Δ_{s_j} is the depth of view and η_{s_j} is the range effect coefficient of sensor s_j . The range values varies with respect to different types of the problem addressed; i.e., there will be three different range values (for detection, recognition, and identification) of each sensor.

The second term is the *weather permeability value*, which is derived by the density values of weather conditions. The term ψ_c^W is the weather density over a polygon p_c where p_c is a polygon that is in between point a and point b . As explained in Section 2, snow, rain and fog are the weather conditions considered in our study, and a polygon on the terrain can be marked with one of these weather conditions with a density value of the condition, where a high density value causes a low visibility value. The density values of weather conditions over all polygons between points a and b are considered a part of LOS algorithms and the maximum value is returned if no intermediate point (between a and b) is obstructed by the terrain.

The last term in (10) is the *object permeability value*, which is set by using the density value of objects. As explained in Section 2, there can be two types of objects located on synthetically generated terrain, which are natural objects such as trees and artificial objects such as buildings. When an object is located on a polygon, the height and the density value of the object is added to the polygon.

B. Computing the Utility of Stealth

The low observable sensors hold great promise in increasing the effectiveness of military operations. Although it may seem that stealth is imperative for nearly all tactical missions, its use needs close examination, particularly in strategic implications. Stealth reduces the risk of many military operations including conducting area searches without revealing the presence of the sensor due to decreased risk of detection.

The utility of stealth value for a set of sensors that are already located on the terrain is derived by subtracting the cost of the total visibility of the located sensors (by using enemy or opponent objects) from one. For this purpose, a predefined number of opponent objects of different types are scattered across the terrain randomly (by utilizing angle and distance constraints), as part of a given scenario m . These objects are the vehicles carrying opponent sensors. In our experimental study, the angle-based locational attributes (such as viewing

angle, depth of view, etc.) of opponent sensors can be set with the values of either the best or the worst sensor in the system.

The utility of stealth of a set of S sensors that are positioned on a set of polygons P by considering a single scenario is computed with the following equation:

$$U_{s,m}(A, S, P) = 1 - \sum_{s_i \in S} V_E(E, P, p_{s_i}) \times R_U(A, s_i, p_{s_i}) \quad (11)$$

where $V_E(E, P, p_{s_i})$ is the maximum visibility of the sensor s_i (located on polygon p_{s_i}) from the set of opponent vehicles E which are located on set of positions P . As in sensor visibility, it is computed by:

$$V_E(E, P, p_{s_i}) = \max_{e_j \in E} \{V_E(e_j, p_k, p_{s_i})\} \quad (12)$$

The $V_E(e_j, p_k, p_{s_i})$ term in this equation is the visibility of a single point on the terrain (where the sensor s_i is located) from an opponent sensor located on polygon p_k . Here, this term is the dual of the term $V_S^P(s_j, p_k, b)$; therefore it is also computed with (10). Further, the utility of stealth of a single sensor s_j for a given scenario is computed by removing the summation term in (11) and updating the denominator of the R_U term with visibility effectiveness term (explained in Section III-A).

The $R_U(A, s_i, p_{s_i})$ term is the ratio of the utility of visibility of the given sensor s_i to the cumulative utility of visibility of all sensors, as shown in the following equation:

$$R_U(A, s_i, p_{s_i}) = \frac{U_{vis}(A, s_i, p_{s_i})}{\sum_{s_j \in S} U_{vis}(A, s_j, p_{s_j})} \quad (13)$$

It should be stated that the denominator in the previous equation is not the overall utility of visibility, but the cumulative utility of visibility. This is due to the fact that the overall utility of visibility may be equal to utility of visibility of a single sensor, which may generate a negative value in (11). If a sensor with a high utility of visibility value (and therefore a high utility ratio) is completely seen by an opponent object, this will significantly decrease the utility of stealth. Another alternative for the second term in (11) (i.e., utility of visibility ratio) is to consider the percentage of polygons that are visible by the given sensor. However, this alternative is not considered, since the difference in the visibility of polygons and weights of the polygons are not considered in this approach.

The process of scattering and computation phases is repeated for a predefined number of scenarios, where the mean value of utilities for the given scenarios determines the overall utility of stealth. If r is the number of different scenarios (for setting the opponent objects), the overall utility of maintaining stealth is computed by:

$$U_{st}(A, S, P) = \frac{\sum_{x=1}^r U_{s,m}(A, S, P)}{r} \quad (14)$$

The locational attributes (polygon number, heading and tilt angles, sensor types) of $r * |E|$ opponent sensors are set at the beginning of the program execution, where $|E|$ is the set of opponents considered at each scenario. The attributes of all opponents do not change during program execution. In this

study, we also propose a dynamic version, which reset all locational attributes of $r * |E|$ sensors at every g generation, where g is set with different values in our experiments.

C. Computing the Utility of Cost

In our study, the term *cost* of a given sensor s_i includes two separate meanings a) the (normalized) financial cost of the sensor, $NCost_F(s_i)$, and b) the (normalized) placement cost of the sensor to its current location (i.e., polygon p_k), which is represented by $NCost_L(s_i, p_k)$. Based on these terms, the utility of cost for a set of sensors S placed on a set of locations P is formally defined by:

$$U_{cost}(S, P) = 1 - \left\{ \frac{\sum_{s_i \in S} (\omega_F \times NCost_F(s_i))}{|S|} + \frac{\sum_{s_i \in S} (\omega_L \times NCost_L(s_i, p_k))}{|S|} \right\} \times N|S| \quad (15)$$

Here, the financial and the placement (locational) costs are independent, and ω_F and ω_L are the weights (from the range [0..1]) of the two cost terms, respectively. In our experiments, $\omega_F = 0.7$ and $\omega_L = 0.3$, unless otherwise specified. The placement cost term in the right side of this equation is computed by $NCost_L(s_i, p_k) = P_L(s_i, p_k) \times \varphi_{s_i}$ where $P_L(s_i, p_k)$ is the locating probability of sensor s_i on polygon p_k ; and φ_{s_i} is a sensor specific constant that is set to 1, unless otherwise specified. The locating probability values of all terrain polygons are set by considering the heights and slopes of polygons and characteristics of the sensors (i.e., sensor carriers). In order to simplify the model, this term can be set with the weights of polygons based on the second method explained in Section III-A. In (15), $N|S|$ is the normalized value of sensors usage, which is computed by:

$$N|S| = \frac{|S|}{E(S)} \quad (16)$$

where $|S|$ is the number of sensors used in the solution, and $E(S)$ is the expected value of the number of sensors for the given terrain.

IV. CONCLUSIONS

Positioning and utilizing multiple directed sensors for acquisition of a given environment is one of the fundamental research topics in various domains including military operations, computer vision, and robotics. Sensor positions and orientations highly affects coverage and stealth performance, especially on 3D terrains. Sensors should be positioned and utilized so that the overall visibility of the given terrain is maximized, while maintaining maximum stealth, and minimizing total acquisition costs of deploying the sensors. Since these objectives are conflicting and utility-independent, the unified utility function is computed as a multiplicative function over the objectives considered as one of the basic results of multi-attribute utility theory.

In this paper, we introduced a new multi-attribute utility function formulation in order to find the optimal number of

sensors, locations and their orientational sensor-specific parameters in a synthetically generated 3-D terrain with multiple objectives. We also formulated its subobjective formulations of the sensor optimization problem. In our following paper [34], we propose a new hybrid evolutionary algorithm to solve the sensor placement problem by considering the multi-attribute utility-based formulation given in this paper.

ACKNOWLEDGMENTS

This research was supported by The Scientific and Technological Research Council of Turkey (TUBITAK) with a research grant (No. 106E159). Additionally, parts of the computations have been carried out by using UYBHM at ITU through a grant (No. 20432008). The authors would like to thank to anonymous referees for their helpful comments and suggestions to improve this manuscript.

REFERENCES

- [1] J. Wang, C. Niu, and R. Shen, "Priority-based target coverage in directional sensor networks using genetic algorithm," *Computers and Mathematics with Applications*, Article in Press.
- [2] M. Marengoni, B. A. Draper, A. Hanson, and R. Sitaraman, "A system to place observers on a polyhedral terrain in polynomial time," *Image and Vision Computing*, vol. 18, no. 10, pp. 773–780, 2000.
- [3] E. Riseman, H. Schultz, W. Lim, B. Draper, and A. Hanson, "Deadalus battlefield visualization system," in *Proc. ARPA Image Understanding*, 1996.
- [4] J. O'Rourke, *Art Gallery Theorems and Algorithms*. Oxford University Press, 1987.
- [5] U. M. Erdem and S. Sclaroff, "Optimal placement of cameras in floor-plans to satisfy task requirements and cost constraints," in *Proceedings of OMNIVIS Workshop*, 2004, pp. 111–120.
- [6] X. Han, E. Lloyd, and C. Shen, "Deploying directional sensor networks with guaranteed connectivity and coverage," in *In Proceedings of IEEE SECON 2008*, 2008, pp. 153–160.
- [7] N. Tezcan and W. Wang, "Self-orienting wireless multimedia sensor networks for occlusion viewpoints," *Computer Networks*, vol. 52, pp. 2558–2567, 2008.
- [8] E. Horster and R. Lienhart, "On the optimal placement of multiple visual sensors," in *Proceedings of the 4th ACM International Workshop on Video Surveillance and Sensor Networks*, 2006, pp. 111–120.
- [9] H. Ma, X. Zhang, and A. Ming, "A coverage-enhancing method for 3d directional sensor networks," in *Proceedings of the 28th Conference on Computer Communications - INFOCOM 2009*, 2009, pp. 2791–2795.
- [10] J. Zhao, S. Cheung, and T. Nguyen, "Optimal camera network configurations for visual tagging," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2(2), pp. 464–479, 2008.
- [11] A. Bakhtari, M. D. Naish, M. Ekandri, E. A. Croft, and B. Benhabib, "Active-vision-based multisensor surveillance - an implementation," *IEEE Transactions on Systems, Man and Cybernetics- Part C: Applications and Reviews*, vol. 36, no. 5, pp. 668–680, 2006.
- [12] A. Mittal and L. Davis, "Visibility analysis and sensor planning in dynamic environments," in *Proceedings of Computer Vision - ECCV 2004*, 2004, pp. 175–189.
- [13] A. Mittal, "Genetic operators for sequencing problems," in *Proceedings of Computer Vision - ECCV 2006, Lecture Notes in Computer Science*, 2006, pp. 522–535.
- [14] A. Mittal and L. Davis, "A general method for sensor planning in multi-sensor systems: Extension to random occlusion," *International Journal of Computer Vision*, vol. 76(1), pp. 31–52, 2008.
- [15] E. Becker, G. Guerra-Filho, and F. Makedon, "Automatic sensor placement in a 3d volume," in *Proceedings of the 2nd International Conference on Pervasive Technologies Related To Assistive Environments*, 2009, pp. 1–8.
- [16] J. Jia, J. Chen, G. Chang, Y. Wen, and J. Song, "Multi-objective optimization for coverage control in wireless sensor network with adjustable sensing radius," *Computers and Mathematics with Applications*, Article in Press.
- [17] I. Stamos and P. Allen, "Interactive sensor planning," in *Proceedings of International Conference on Computer Vision and Pattern Recognition*, 1998, pp. 489–494.
- [18] M. K. Reed and P. K. Allen, "Constraint-based sensor planning for scene modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1460–1467, 2000.
- [19] S. Y. Chen and Y. F. Li, "Optimal viewpoint planning for model-based robot vision," in *Proceedings of IEEE 2002 World Congress on Computational Intelligence (WCCI)/Congress on Evolutionary Computation*, 2002, pp. 634–639.
- [20] —, "Constraint-based sensor planning for scene modeling," *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 34, pp. 393–408, 2004.
- [21] S. Kristensen, "Sensor planning with bayesian decision theory," *Robotics and Autonomous Systems*, vol. 19, pp. 273–286, 1997.
- [22] L. Mihaylova, T. Lefebvre, H. Bruyninckx, K. Gadeyne, and J. D. Schutter, "Active sensing for robotics - a survey," in *Proceedings of International Conference in Numerical Methods and Applications*, 2002, pp. 316–324.
- [23] J. Spletzer and C. Taylor, "Sensor planning and control in a dynamic environment," in *Proceedings of International Conference on Robotics and Automation*, 2002.
- [24] R. Tharmarasa, T. Kirubarajan, and J. Peng, "Optimization-based dynamic sensor management for distributed multitarget tracking," *IEEE Transactions on Systems, Man and Cybernetics- Part C: Applications and Reviews*, vol. 39, no. 5, pp. 534–546, 2009.
- [25] D. Cook, P. Gmytrasiewicz, and L. B. Holder, "Decision-theoretic cooperative sensor planning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 1013–1023, 1996.
- [26] R. Keeney and H. Raiffa, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Cambridge University Press, 1993.
- [27] J. Byers and G. Nasser, "Utility-based decision-making in wireless sensor networks," in *Proceedings of the 1st ACM international Symposium on Mobile Ad Hoc Networking and Computing*, 2004, pp. 143–144.
- [28] V. Isler and R. Bajcsy, "The sensor selection problem for bounded uncertainty sensing models," in *Proceedings of the 4th international Symposium on information Processing in Sensor Networks*, 2005, pp. 24–27.
- [29] D. K. F. Bian and R. Govindan, "Utility-based sensor selection," in *Proceedings of The Fifth International Conference on Information Processing in Sensor Networks (IPSN 2006)*, 2006, pp. 11–18.
- [30] M. D. Proctor and W.J. Gerber, "Line-of-sight attributes for a generalized application program interface," *the Journal of Defense Modeling and Simulation*, vol. 1, no. 1, pp. 43–57, 2004.
- [31] D. Handerson, "Modterrain: A proposed standart for terrain representation in entity level simulation," Master's Thesis, Naval Postgraduate School, California, 1999.
- [32] R. Clemen, *Making Hard Decision*. PWS Kent Publishing, 1991.
- [33] J. Dyer, "Remarks on the analytic hierarchy process," *Management Science*, vol. 36, no. 3, pp. 249–259, 1990.
- [34] H. Topcuoglu, M. Ermiş, and M. Sifyan, "Positioning and utilizing sensors on a 3d terrain: Part 2 - solving with a hybrid evolutionary algorithm," *IEEE Transactions on Systems, Man and Cybernetics- Part C: Applications and Reviews*, Accepted for publication.

Positioning and Utilizing Sensors on a 3-D Terrain

Part II - Solving with a Hybrid Evolutionary Algorithm

Haluk Topcuoglu, *Member, IEEE*, Murat Ermis, and Mesut Sifyan

Abstract—In this paper, we explore using a hybrid evolutionary algorithm for deploying and configuring a set of given sensors on a synthetically generated 3-D terrain. In our EA-based solution, various methods are considered in order to incorporate specialized operators for hybridization, including problem-specific heuristics for initial population generation, intelligent variation operators (Contribution-Based Crossover operator and Proximity-Based Crossover operator) which comprise problem specific knowledge, and a local search phase. The experimental study validates finding the optimal balance among visibility-oriented, stealth-oriented, and cost-oriented objectives. The obtained results also indicate the effectiveness and robustness of our hybrid evolutionary algorithm-based solution for various practical scenarios with different objectives.

Keywords—Sensor planning, local search, hybrid genetic algorithms.

I. INTRODUCTION

COLLECTING comprehensive and accurate data from sensors is still a demanding problem; even though remote sensing technologies have become better and more widespread in various domains due to the decrease in cost of producing more sophisticated sensing devices. Presently, there are different types of sensors and sensing technologies with different costs that are used to search and detect targets in different domains. Specifically, reconnaissance, surveillance, and target acquisition are example military operations where deploying and utilizing sensors play a vital role [1]. Therefore, deployment of multiple sensors on a given environment and finding the most suitable orientation for coverage is an important and challenging combinatorial optimization problem [2]–[4].

We develop a framework that has a novel approach for determining the optimal number of sensors, locating and setting their orientational sensor-specific parameters on a synthetically generated 3-D terrain with multiple objectives [5]. Our solution approach relies on the rational trade-off between three conflicting objectives which are maximizing the coverage area while maintaining the maximum stealth, and minimizing the total acquisition cost of deploying the sensors. These objectives are utility-independent; and when multi-objective

utility theory is considered, a multiplicative function over these attributes fits for the total utility function in our study.

Motivated by our constructed framework, this paper explores the use of a hybrid evolutionary algorithm for sensor placement and orientation problems. Evolutionary Algorithms [6]–[8] (EAs), which are inspired by the Darwinian evolutionary process present in nature, operate by iteratively generating a population of chromosomes that are encoded forms of candidate solutions. In order to reach an optimum solution, EA typically uses various genetic operators (including selection, crossover, and mutation) by applying the “survival of the fittest” principle. EAs have been efficiently used in a wide variety of applications in the engineering, science and business fields [6], [9].

However, simple evolutionary algorithms are generally poor for solving complex combinatorial problems [10]. EAs are usually strengthened with problem-specific characteristics or operators [8], [11]–[13]; and may be combined with specialized heuristics to produce hybrid systems, which are called different names including hybrid evolutionary algorithms and memetic algorithms [14]–[16].

In this paper, we propose two specialized crossover operators called the Contribution-Based Crossover (CBX) and the Proximity-Based Crossover (PBX) that comprise the domain specific information on the sensor placement problem. The CBX operator carries a sensor to the offspring from one of its parents based on their contributions, which is expressed in terms of the utility of the related sensor. Further, the PBX operator considers the acceptable proximity distances and angular differences in order to carry a sensor to the corresponding sensor. Experiments on synthetic 3-D terrains with various characteristics are conducted in order to present the effectiveness of our EA-based framework. The results of the experimental study clearly show that our proposed approach is very successful in deploying and utilizing sensors by considering the multiple objectives.

The remainder of the paper is organized as follows: In Section 2, we briefly introduce our sensor optimization framework. Section 3 gives our proposed hybrid EA-based formulation for solving the sensor optimization problem. Section 4 presents performance evaluation and experimental study of our work; and finally, Section 5 concludes the paper.

II. SENSOR OPTIMIZATION FRAMEWORK

In this section, we present the details of our sensor optimization framework which consists of three modules (given

Haluk Topcuoglu is with the Computer Engineering Department, Marmara University, Goztepe Kampusu, Kuyubasi, Istanbul, 34722, Turkey. email: haluk@marmara.edu.tr

Murat Ermis is with the Industrial Engineering Department, Turkish Air Force Academy, Yesilyurt, Istanbul, 34149, Turkey. email: m.ermis@hho.edu.tr

Mesut Sifyan is with the Computer Engineering Department, Marmara University, Goztepe Kampusu, Kuyubasi, Istanbul, 34722, Turkey. email: mesutsifyan@marun.edu.tr

in Fig. 1): the Terrain Generation Module (*the TerGen Module*), the Scenario Editor Module (*the SenEd Module*), and the Sensor Optimization Module (*the SenOpt Module*). The *TerGen Module* is used for generating synthetic terrains and covering these terrains with natural and/or artificial objects and weather conditions. It executes the Hill Algorithm [17] and the Diamond-Square Algorithm [18] (see Fig. 2) for terrain generation. The Hill Algorithm considers random circular displacements on the initially flat terrain, where the radius of each displacement causes a rough and rocky or smooth and rolling terrain. On the other hand, the Diamond-Square Algorithm starts with a 2-D grid and applies the diamond and the square steps in a grid of points repeatedly in order to cover the entire plane in squares.

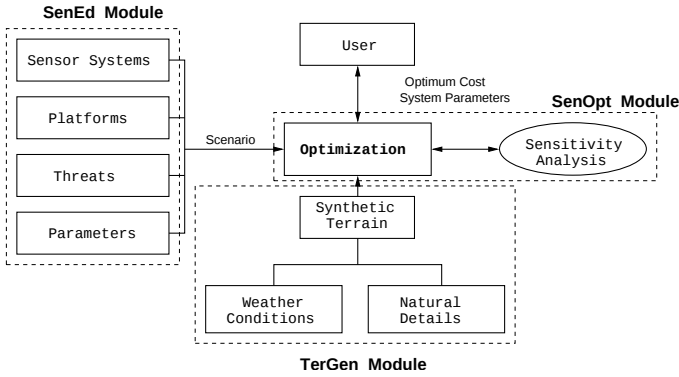


Fig. 1. Software Architecture of our Framework

After a terrain is generated, it is covered natural objects (forest, bush, lakes, etc.) or artificial objects (buildings, bridges, etc.). The *TerGen Module* also sets various weather conditions on the terrain including rain, snow, fog, and temperature value. The weather conditions are taken into account during the visibility calculations of sensors. The *TerGen Module* provides an interface to set different importance levels and constraints for regions of the generated terrain. Importance levels are intended to differentiate the importance of regions of a given terrain. A user may put constraints on one or more area(s) on the terrain to restrict locating sensors on these regions.

In our study, each generated terrain is represented by a set of triangular polygons, where each polygon p_i has the following attributes stored in the database.

- Three vertices (each of which given in x,y,z coordinates) of the polygon p_i .
- Normal vector of p_i .
- Center of mass of the polygon (given in x,y,z coordinates), which is computed and stored during the terrain generation phase.
- Weather density (ψ_i^W) and object density (ψ_i^O) of polygon p_i , where the values are in the range of $0 \leq \psi_i^W, \psi_i^O \leq 1$. The density values are used to compute the permeability value of a ray that traverses from an origin to a predefined destination through the given polygon p_i by considering object and weather conditions over the polygon.
- Weight of the polygon, W_i , which indicates the importance of the given polygon. The weights of each polygon

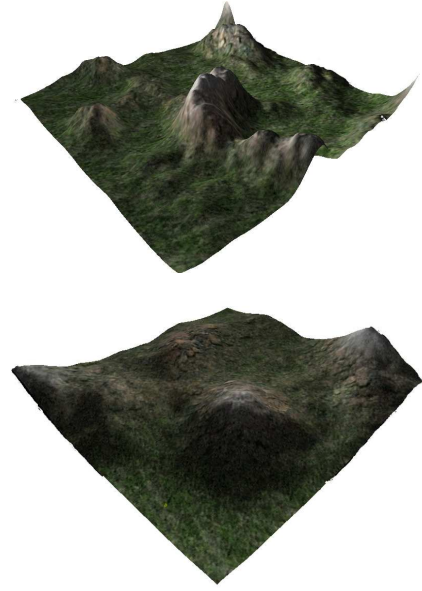


Fig. 2. (a) A Rough Terrain generated by the Hill Algorithm, and (b) A Smooth Terrain generated by the Diamond Square Algorithm

can be set either by the user during the terrain generation process, or by the topographic analysis of the terrain.

The *SenEd Module* of our framework defines the parameters and attributes of various sensor systems. As explained in [5], sensors attributes are classified in two categories: behavioral attributes and locational attributes. The attributes in the former set are determined by the *SenEd Module*, which includes the depth of view (Δ_i), horizontal viewing angle (α_i), vertical lower viewing angle (β^L), vertical upper viewing angle (β^U) and the financial cost ($Cost_F(i)$) of each sensor s_i . We consider a multi-mode approach (detection, recognition, and identification modes) for characteristics of sensors in order to provide more accurate and meaningful results for perception process of located sensors. Therefore behavioral attributes of each sensor may vary according to the mode considered. Moreover, the latter group includes attributes determined by the *SenOpt Module*, which are the position (i.e., the polygon number p_{s_i}), the heading angle (θ_i) and the tilt angle σ_i of each sensor s_i .

The *SenOpt Module* optimizes the deployment of sensors for a given scenario with our novel hybrid evolutionary algorithm, which positions sensors and utilizes sensors' parameters/attributes. Specifically, the algorithm determines the most suitable type, number, position, and parameter values of the sensors that will maximize the provided objective function (given a multi-attribute function as explained in [5]). Since the *SenOpt Module* performs the sensitivity analysis of the optimization results according to various success criteria, experimental designs can be prepared accordingly. Specifically, the tests are repeated in various synthetically generated terrains in order to improve the measurement of the efficiency of our EA-based approach.

III. PROPOSED HYBRID EVOLUTIONARY ALGORITHM FOR SENSOR OPTIMIZATION PROBLEM

Although simple evolutionary algorithms have been widely used for many optimization problems in the literature, they are not generally efficient for the complex combinatorial problems [10]. Specialized and/or problem-specific operators [8], [11], [12] are combined with evolutionary algorithms improving their performance, where these hybrid systems are called hybrid evolutionary algorithms or memetic algorithms [14], [16] in the literature. In order to build our hybrid evolutionary algorithm, problem-specific heuristics are considered for initial population generation. Two novel variation operators that incorporate problem specific knowledge are presented in our study, which are the Contribution-Based Crossover and the Proximity-Based Crossover operators. Additionally, a local search phase is applied on the output of the variation operators.

In our EA-based approach, each solution contains the type and the locational attributes (position, heading angle, tilt angle) of sensors. Possible solutions of a given problem may have different number of sensors deployed; therefore string representation in our EA-based approach supports variable-length chromosomes. Additionally, there is no order of sensors in a given solution. In this study, a steady-state Genetic Algorithm is applied which generates one individual at each iteration. We consider tournament selection mechanism and the tournament size is varied in the experiments. The value of the fitness function is set with the value of the total utility function (based on the equation derived in [5]).

A. Initial Population Generation

The first phase to generate a solution for initial population is to determine the number of sensors considered in the solution, which is set randomly between the upper and the lower limit of the sensor quantity. The sensor mode (which can be detection, recognition, or identification) and the horizontal depth of view (Δ) on the given sensor mode are considered in order to calculate the limits. Assume that our problem is the recognition of the terrain and the maximum depth of view of sensors for recognition mode is equal to Δ_{max} and the minimum depth of view is equal to Δ_{min} . The number of sensors $|S|$ in the given solution is determined randomly by using (1):

$$\left(\frac{Terrain_Dim}{\Delta_{max}}\right)^2 * \gamma_1 \leq |S| \leq \left(\frac{Terrain_Dim}{\Delta_{min}}\right)^2 * \gamma_2 \quad (1)$$

In this equation, $Terrain_Dim$ is the length of one dimension of a polygon on XZ plane of the generated terrain. It should be noted that the length, with respect to the X and Z axis, is the same, since terrain generation algorithms used in this study consider grids. The terms γ_1 and γ_2 are constants, which are assigned from the ranges $0.5 \leq \gamma_1 \leq 1.25$ and $1.0 \leq \gamma_2 \leq 1.75$, with respect to the complexity of the terrain. After the sensor quantity of a solution is determined, the type of each sensor is randomly set by preserving the inverse proportionality to sensor capacities (i.e., the view ranges with respect to given sensor mode). Therefore, a given solution

will have low capacity (thus low cost) sensors with high probability and it will have high capacity (thus high cost) sensors with low probability. Then, locations of sensors in each solution are determined by a heuristic method (explained below) for a predefined percentage of the initial population and the locations are set randomly for the remaining part of the initial population.

Our heuristic approach aims to distribute the selected sensors evenly on the given terrain. If there are k sensors (same or different type) to be placed, firstly the terrain is partitioned to $\lfloor \sqrt{k} \rfloor$ rows and columns, which will generate a total $\lfloor \sqrt{k} \rfloor * \lfloor \sqrt{k} \rfloor$ regions after partitioning. First, one sensor (of previously determined type) is assigned to each region; then, the remaining $k - \lfloor \sqrt{k} \rfloor * \lfloor \sqrt{k} \rfloor$ number of sensors are located on the terrain randomly. After a region is selected, the sensor is located on the center polygon of the region with a probability of 20% and on any other polygon from the region with a probability of 80%. Finally, the heading angle (θ) and the tilt angle (σ) of each sensor given in the solution of initial population are determined randomly by considering feasible ranges of angles.

B. Crossover Operators

Our EA-based approach includes a set of variation operators (crossover and mutation operators) and a local search phase. First, a crossover operator is applied on two parents that are selected by the tournament selection method. Crossover operators consider problem-specific information in order to carry the sensors from parents to offspring. Then, a local search is applied on the output from the selected crossover operator, which aims to improve the quality of the offspring by modifying heading and/or tilt angles. Finally, a mutation operator is applied on the output of the local search.

In our study we consider three different crossover operators, which are the Contribution-Based Crossover (CBX), the Proximity-Based Crossover (PBX), and the Cut and Splice Crossover (CSX). The CBX and CSX operators are the ones that consider problem-specific knowledge. All three crossover operators generate a single offspring as the output.

1) *Contribution Based Crossover (CBX) Operator*: The main idea behind this operator is to carry a sensor to the offspring from one of the parents based on its contribution, which is expressed in terms of utility of the sensor. The sensors in both parents are examined in this operator. One sensor is selected randomly from each parent and the better one, which has higher total utility value than the other sensor, is moved into the offspring as the first sensor of the offspring. At each step, one of the remaining sensors from each parent is selected and the best one of selected sensors is moved to the offspring if the sensor keeps acceptable proximity distance with all sensors that are already in the offspring. The steps of calculating the proximity distance is given below:

Assume that the sensor s_m is the sensor that has the highest total utility value (highest contribution value) among the two sensors that are selected from the remaining sensors of each parent solution; and assume that s_m is already located on polygon m with the heading angle θ_m as given in Figure 3.

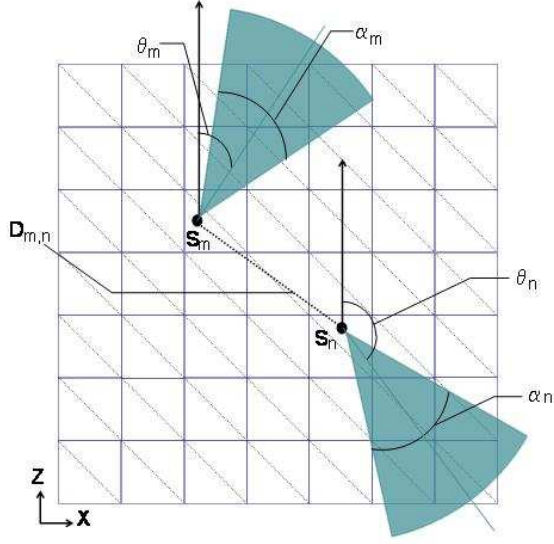


Fig. 3. Contribution Based Crossover

Assume that s_n is one of the sensors that is already copied to offspring; and s_n is located on polygon n with the heading angle θ_n . Assume that (horizontal) viewing angles of s_m and s_n are α_m and α_n , respectively. In order to decide whether sensor s_m located on polygon m will be accepted for the offspring, the following two inequalities are considered:

$$D(n, m) > (\Delta_n + \Delta_m) \times \mu \quad (2)$$

$$|\theta_m - \theta_n| > \frac{\alpha_m + \alpha_n}{2} \times \kappa \quad (3)$$

The right side of the first inequality is the acceptable proximity distance. The distance between sensors (in other words, the distance between the center of masses of polygons m and n , $D(m, n)$) should be equal or greater than the acceptable proximity distance, which is computed by multiplying the summation of depth of views with a problem specific constant, μ , where $0 < \mu < 1$. The depth of view value of each sensor is based on the sensor mode considered in the experiments.

If s_m validates the first inequality, it is selected for the offspring the second inequality is no longer considered. Similarly, if the sensor fails in the first inequality but it validates the second one, it is still moved to the offspring. If the angular distance between heading angles of sensors (the left side in the second inequality) is greater than the summation of half of the horizontal viewing angles of two sensors, the candidate sensor is copied since they are looking at different directions even if the distance between them is less than the acceptable proximity distance. The term κ , an angular constant related with overlapping ratio of horizontal viewing angles of sensors, is set to 0.8 unless otherwise specified.

If it does not validate both of the inequalities, the candidate sensor is dropped and sensor selection is performed from the remaining sensors of parents. This process is repeated as long as the number of sensors ($|S|$) moved to the offspring is less than the upper limit, which is computed by using the following

inequality:

$$|S| \leq (|S_1^P| + |S_2^P|) * \epsilon \quad (4)$$

In this inequality, $|S_1^P|$ and $|S_2^P|$ are the number of sensors allocated in the first and the second parents of the offspring, respectively; and ϵ is a constant which is less than 1. Here, there is no lower limit for sensor quantity in the offspring. The number of sensors in the offspring can be less than both two parent solutions due to the restrictions on both proximity distance and heading angle.

2) *Proximity Based Crossover (PBX) Operator*: In this operator, firstly, a crossover point k is selected randomly by considering $1 \leq k \leq m - 1$ where m is the minimum of the number of sensors used in two parent solutions. Starting from the first sensor, k sensors of the first parent are copied to the offspring. The next phase is to copy sensors from the second parent. Starting from the first element of the second parent, sensors are copied to the offspring in order by considering the inequalities given in (3). The first equation is for testing the proximity distance and if the current sensor of the second parent passes this test for all sensors already in the offspring it is copied to the offspring. Otherwise the second test related with viewing angle is applied. The upper bound for the number of sensors in the offspring (given in (4)) is applied in the PBX operator; we do not consider a lower bound.

3) *Cut and Splice Crossover (CSX) Operator*: This operator is similar to the original single-point crossover operator as proposed in the literature. As in the previous operator, a crossover point is selected randomly from the range $[1..m - 1]$ where m is the minimum of number of sensors existing in two parents. Then, both parents are separated at the given crossover point and two children are created by exchanging tails and the better one is selected as the offspring. The tests of proximity distance and viewing angle are not applied in this operator, and there is no upper bound on the number of sensors considered.

C. Local Search Phase

After a solution is generated using the crossover operator, the local search phase targets on improving the quality of the solution by modifying angular attributes of sensors. The steps of the local search phase is presented in Fig. 4. At each iteration of the local search, the sensor with the minimum utility of visibility value is selected. Here, we consider sensor's individual utility of visibility values.

The heading angle is updated by adding a predefined increment amount (δ_H) repeatedly until the first improvement on utility of visibility for the given sensor (steps 8-10). Here, the value of δ_H is a predefined constant which is set based on the terrain size (i.e. the number of polygons on the terrain) and parameters of terrain generation algorithms that are for specifying the terrain complexity. If there are large number of polygons on the terrain, δ_H can be low; otherwise a higher value can be set. In our experiment, $\delta_H = 25^\circ$, unless otherwise specified.

The next phase (given in steps 10-13) is to modify the tilt angle of the sensor in order to improve sensor's visibility. Similar to the heading angle, an increment amount (δ_T) is added to the tilt angle of the given sensor repeatedly. This

process is repeated until all possible alternatives covered and the alternatives are selected from the range given below:

$$-90 + \beta^L \leq \sigma \leq 90 - \beta^U \quad (5)$$

where β^L , β^U are the lower and upper (vertical) Field-of-View (FOV) angles of the given sensor and σ is the tilt angle which is already updated with the given increment amount.

This process is repeated until one of the following two conditions occurs: a) the cumulative improvement ratio on the utility of visibility of the modified sensors (I_R) is greater than or equal to 20% or b) where at least a 10% of sensors in the original solution are considered as part of the local search process. The percentage values considered for the stopping condition is updated with different values as part of our experimental study. The improvements are not directly related with the total utility value of the solution, because different sensors which view common polygons are not tested. Only local improvements of sensors are considered and number of sensors to be examined is limited in order to bound the running time of this phase.

-
1. Sensors of the input solution is kept in the set S .
 2. $I_R \leftarrow 0$ // total improvement ratio.
 3. $q \leftarrow 0$ // number of sensors considered.
 4. $\delta_H \leftarrow$ increment amount for the heading angle.
 5. $\delta_T \leftarrow$ increment amount for the tilt angle.
 6. Select sensor s_m that has $\min_{s_i \in S} U_v(A, s_i, p_k)$.
 7. **while** ($I_R < 0.20$ **or** $q < 0.10 \times |S|$) **do**
 8. **while** no increase in $U_v(A, s_m, p_k)$ **and** $\theta < 360$ **do**
 9. $\theta_m = \theta_m + \delta_H$
 10. If no increase in utility of visibility
 11. Store the original value of θ_m .
 12. **for** ($\sigma = -90 + \beta^L$; $\sigma \leq 90 - \beta^U$; $\sigma = \sigma + \delta_T$) **do**
 13. Compute $U_v(A, s_m, p_k)$ with new σ value
 14. Keep σ value which maximizes $U_v(A, s_m, p_k)$.
 14. If no increase in utility of visibility
 15. Store the original value of σ_m .
 16. $q \leftarrow q + 1$
 17. Select another s_m that has $\min_{s_i \in S} U_v(A, s_i, p_k)$.
 18. **endwhile**
-

Fig. 4. Steps of the Local Search Phase

D. Mutation Operators

Mutation operators in our study can be considered in two categories, which are the locational mutation operators, and the angular mutation operators. In our experimental study, we can either select one of the two categories randomly at the first mutation and do not change the category during the execution, or select the category randomly at each mutation case.

1) *Locational Mutation Operators*: This category deals with only the locations of sensors (i.e. the polygon numbers on which they are located) and the heading and tilt angles are not modified. If this class is selected, one of three different operators (called update, delete, and insert operators) is selected randomly.

- *Update*. This operator updates the location of a sensor in two steps, which include selecting the sensor, and selecting the new location of the selected sensor. We consider two alternatives for each of these two steps. The sensor for the update operator is selected either randomly or is the one that has the minimum utility of visibility value. The target location is either selected randomly from the whole terrain or is selected randomly from the same region where the original sensor is located. The region concept is explained in the subsection for generating the initial population.
- *Delete*. This operator deletes one of the sensors, which can be either a randomly selected sensor or the one which has the minimum utility of visibility value.
- *Insert*. A new sensor is added to the solution on a polygon that is randomly selected either from the whole terrain or from the region which already includes the sensor with the minimum utility of visibility value. The sensor type and the heading angle and the tilt angle of the new sensor is set randomly by considering the feasibility ranges of angles.

2) *Angular Mutation Operator*: This category is for updating the angular properties of a sensor, which is either selected randomly or is the sensor of the given solution that minimizes the utility of visibility value. This class has two operators, one of which is selected randomly. The mutation operators are:

- *Mutation on Heading Angle*. This updates the heading angle of the selected sensor randomly.
- *Mutation on Tilt Angle*. This updates the tilt angle of the sensor randomly from a predefined feasible range.

One extension is to select one of the two mutation operators from this class non-uniformly. Specifically, the mutation on the heading angle can be selected with a higher probability than the mutation on the tilt angle.

In addition to these two classes of mutations, updating the sensor type is also considered as a mutation operator. In our experimental study, type of a randomly selected sensor is updated randomly at every 100 generations; if the update causes an improvement, it is accepted; otherwise it is rejected. The update on the sensors type is applied for both of the mutation classes given above.

IV. EXPERIMENTAL STUDY

In this section, we present the results of computational experiments in order to evaluate the effectiveness of our hybrid evolutionary algorithm. This is coded in C programming language and the computational experiments are conducted on a cluster of machines, each of which has an Intel Xeon 2.33 Ghz. processor running the Linux operating system. The default values of general parameters listed in Table I are considered in our study, unless otherwise stated.

Although the performance of our genetic algorithm is examined for longer durations as part of the experimental study, it is not worthwhile to run more than 1,000 generations. The total utility value of each generation of a sample run with a visibility-domain mission is given in Fig. 5.a. This figure indicates that the amount of fitness increased before

TABLE I
DEFAULT SETTINGS OF SELECTED PARAMETERS OF EXPERIMENTATION

Parameter	Value
1-Coefficients of Sensors Quantity Limits in Initial Population	$\gamma_1 = 1.00 \gamma_2 = 1.75$
2-Coefficient for Depth of View in CBX and PBX	$\mu = 0.4$
3-Coefficient for Sensor Quantity Inherited from Parents to Offspring	$\epsilon = 0.5$
4-Improvement Ratio in Local Search Phase	$I_R = 20\%$
5-Sensor Ratio in Local Search Phase	$SR = 10\%$
6-Number of Enemies (detection, recognition and identification modes)	(10, 40, 100)
7-Tournament Size	5
8-Number of Generations for Termination	1000

the 1,000th generation is significantly greater than the amount of fitness increase after the 1000th generation. Based on the cross-correlation figure (given in Fig. 5.b) for the same run, the utility of visibility and the utility of stealth increase while the utility of cost decreases with an increase in number of generations.

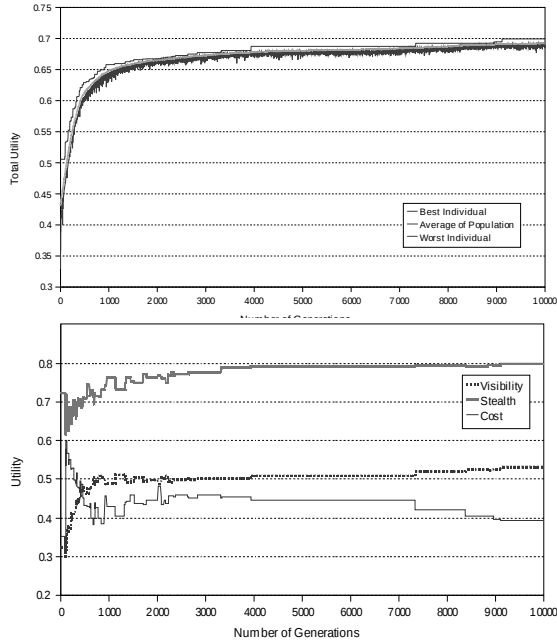


Fig. 5. Performance of Hybrid Evolutionary Algorithm with respect to Number of Generations for (a) Total Utility Value, and (b) Three Individual Utility Values

The experimental study presented in this section is classified in two categories. In the first category, the experiments are performed by considering the algorithm-specific parameters of our evolutionary algorithm. The second category focuses on the problem-specific parameters of the experiments.

A. Experiments on Algorithm-Specific Parameters

The four algorithm-specific parameters which are population size, type of the crossover operator, type of the mutation operator, and sensor selection criteria in mutation along with their sets of values are presented in Table II. The values in the table causes 54 different combinations, each of which is run with 30 replications. Moreover, the remaining parameters

of our experimentation is set with the default values of the 1,620 tests considered. Specifically, the initial population of our algorithm is set randomly without any heuristic. The increment amount of heading angle (δ_H) is set to 25° , which is one-fourth of the minimum horizontal viewing angle of sensors considered. The increment amount of the tilt angle is set to $\delta_T = 3^\circ$. A visibility-dominant, that occurs mission by selecting recognition mode of sensors with a rough terrain is considered.

In each complete replication of the experiments, all possible combinations of the levels of factors given in this table are examined; therefore, we involve full factorial design in this study [19]. Based on the analysis of variance (ANOVA) on the total utility value for a visibility-dominant mission, four main effects are observed: population size, crossover operator, selection strategy of the source sensor in mutation, and interaction between the population size and the crossover operator. The “Pred. R-Squared” of 0.8733 is in reasonable agreement with the “Adj. R-Squared” of 0.8742. The optimization results are acquired by setting the goals for the total utility value and generating the optimal conditions. Specifically, the population size is set to 50, the contribution-based crossover is selected, and the sensors with the minimum visibility value is considered for the mutation operator. However, the mutation operator does not have a significant effect on total utility for a visibility-dominant mission.

B. Experiments on Problem-Specific Parameters

The second category aims at mainly identifying the problem-specific parameters including terrain type and the three utility weights for determining a specific mission. Additionally, it includes three algorithm-specific parameters that are not considered in the previous subsection, which are the strategy for selecting the initial population, the increment value in heading angle and the increment value in tilt angle (see Table III). The best values of the algorithm-specific parameters observed in the previous subsection are taken into account in our experiments given in this subsection.

Based on the analysis of variance, the first three parameters (given in the first three columns of Table III) do not significantly affect the total utility values. The terrain type, whether a smooth or a rough terrain, and the mission type, given with different weights of three utilities, affect the quality of solutions as expected. Therefore, the experiments of the previous subsection are extended for the other alternative missions, which are the stealth-dominant mission and the cost-dominant mission.

For a stealth-dominant mission, it is observed that the population size, crossover type, mutation type and interactions between them affect the total utility value significantly. This relation is strongly proven with the “Pred. R-Squared” of 0.9479 and “Adj. R-Squared” of 0.9488. The parameters given in Table II for the stealth-dominant mission are set to those values observed for the visibility-dominant mission. Similarly, for a cost-dominant mission, all four parameters and interactions between them affect the total utility value significantly. This relation is reasonably agreed with “R-Squared” values.

TABLE II
ALGORITHM-SPECIFIC PARAMETERS

Population Size (A)	Crossover Operator (B)	Mutation Operator (C)	Source Sensor in Mutation (D)
1. 30	1. Contribution-Based (CBX)	1. Location-Based	1. Random Sensor
2. 50	2. Proximity-Based (PBX)	2. Angular-Based	2. Sensor with Minimum Visibility
3. 100	3. Cut and Splice (CSX)	3. Random	

TABLE III
PROBLEM-SPECIFIC PARAMETERS

Initial Population Selection Strategy	Increment in Heading Angle	Increment in Tilt Angle	Utility Weights (w_{vis} , w_{st} , w_{cost})	Terrain
(75% Heuristic, 25% Random)	25	3	(0.6,0.3,0.1)	Terrain 1 (smooth)
(50% Heuristic, 50% Random)	50	5	(0.3,0.6,0.1)	Terrain 2 (smooth)
100% Random	75	7	(0.25,0.25,0.50)	Terrain 3 (rough)
		10		Terrain 4 (rough)

The population size is set to 30, and a single-point crossover and location-based mutation operator is selected for a cost-dominant mission.

C. Performance Evaluation and Discussion

In this section, the effects of various factors on the quality of solutions are measured and related discussions are provided.

a) The Effects of Termination Condition in Local Search:

Since the sensor configurations are generated randomly in the initial population, they may not have the most suitable angular behaviors. Specifically, the utility of visibility for most of the sensors in initial population may be less than 0.01. The local search phase given in Fig. 4 aims at improving the utility of visibility of sensors by updating the heading angle and the tilt angle of a set of selected sensors of the given individual. In our experiments, the local search phase is terminated either when a total of 20% improvement is achieved or the number of sensors examined reaches to 10% of sensors in the given solution.

Based on the experimental study on number of sensors examined in local search phase at each generation, the amount of improvement on the worst sensor is large enough to provide the predefined improvement ratio (i.e. 20% improvement on the utility of visibility) during the first 200 generations of execution for the visibility-dominant mission (see Fig. 6). The number of sensors examined increases slightly with an increase in generation number. A total of six sensors are examined before leaving the local search phase after the 300th generation. The number of sensors in individuals is around 60 for the given terrain when the recognition sensor mode is considered; therefore, six is the upper limit when 10% sensor ratio is considered. As a result, the local search phase is terminated due to the limits on improvement ratio in the earlier generations. Moreover, it is terminated due to the sensor ratio for the later part of a given execution. A similar observation is done for the other missions considered.

b) The Effects of Alternative Strategies for Local Search:

As mentioned above, the local search phase of our algorithm improves the quality of solutions by modifying angular attributes of a set of selected sensors from the individuals. Table IV presents the performance of three alternatives on the

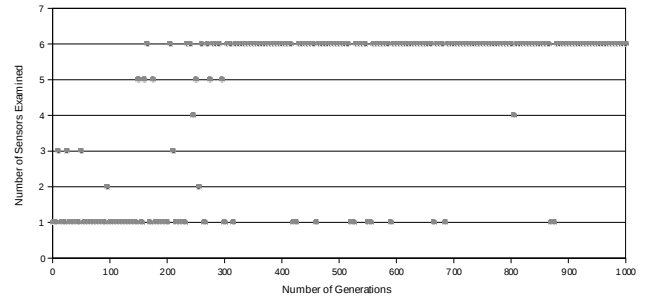


Fig. 6. Local Search Termination Condition for a Visibility-Dominant Mission

local search phase. The first alternative is denoted as “Heading $\rightarrow$ Tilt” which updates the heading angle of the selected sensor on its first improvement; then all possible alternatives of tilt angles are examined to get the maximum utility of visibility. The second one, “Tilt $\rightarrow$ Heading”, changes order of updates on heading angle and tilt angle of the first alternative. Further, the third alternative removes the local search phase completely by increasing the generation size to 2,000. It should be stated that the generation size is set to 1,000 for the first and the second alternatives.

The second column is for mission type, where M1, M2, and M3 represent the visibility-dominant, stealth-dominant, and cost-dominant missions, respectively. The three cases for each mission type give almost equal total fitness values. Since, the number of heading angle alternatives is less than the number of tilt angle alternatives, the running time of the “Heading $\rightarrow$ Tilt” case is longer than the running time of the “Tilt $\rightarrow$ Heading” case. The algorithm, without considering the local search phase, reaches equal total utility value with the other two alternatives by doubling the number of generations. Therefore, the “Tilt $\rightarrow$ Heading” case is selected in our experiments due to its shorter running time.

c) The Effects of Injecting Individuals to Population:

The first extension is to inject three sensors randomly into 60% of the population at every 100th generation, where the attributes of the inserted sensors are randomly generated. The total utility value of this injection method for each mission given in Table V are very close to that of Table IV, where tests of the both tables are done with the same terrain, the

TABLE IV
PERFORMANCE EVALUATION OF ALTERNATIVE TECHNIQUES FOR LOCAL SEARCH PHASE

Update on Local Search	Mission Type	Number of Sensors	Utility of Visibility	Utility of Stealth	Utility of Cost	Total Utility	Exec. Time (min.)
Heading→Tilt	M1	64.2	0.4953	0.7549	0.4476	0.6621	26
	M2	63.8	0.4192	0.9183	0.5491	0.8418	25
	M3	12.6	0.1186	0.9683	0.9117	0.8618	9
Tilt→Heading	M1	64.2	0.4992	0.7479	0.4425	0.6614	18
	M2	62.8	0.4263	0.9112	0.5516	0.8408	17
	M3	12.0	0.1191	0.9660	0.9114	0.8610	8
Local Search Excluded	M1	72.4	0.5041	0.7645	0.3770	0.6615	24
	M2	67.0	0.4176	0.9224	0.5049	0.8363	21
	M3	7.1	0.1123	0.9949	0.9447	0.8901	8

TABLE V
COMPARATIVE RESULTS OF INJECTION METHODS FOR THE HYBRID EVOLUTIONARY ALGORITHM

Updates Considered	Mission Type	Number of Sensors	Utility of Visibility	Utility of Stealth	Utility of Cost	Total Utility
Sensor Injection	M1	77.0	0.5253	0.7506	0.3606	0.6694
	M2	74.6	0.4489	0.8999	0.4754	0.8304
	M3	38.9	0.2583	0.8570	0.7272	0.7626
Sensor Injection & Deletion	M1	57.6	0.5016	0.7718	0.4556	0.6751
	M2	55.8	0.4218	0.9291	0.5748	0.8554
	M3	23.0	0.2013	0.9181	0.8281	0.8261

TABLE VI
PERFORMANCE RESULTS ON A SMOOTH TERRAIN WITH DIFFERENT SENSOR MODES

Sensor Mode	Mission Type	Algorithm	Number of Sensors	Utility of Visibility	Utility of Stealth	Utility of Cost	Total Utility
Detection	M1	RS	13.9	0.3578	0.6536	0.5161	0.5267
		HEA	14.7	0.5716	0.8000	0.4607	0.7419
	M2	RS	9.7	0.2337	0.7997	0.6785	0.6894
		HEA	14.7	0.5200	0.9615	0.5415	0.9216
Recognition	M1	RS	80.7	0.4283	0.6108	0.4389	0.5524
		HEA	59.7	0.5561	0.7433	0.4840	0.7061
	M2	RS	54.9	0.3036	0.7020	0.6241	0.6465
		HEA	62.3	0.4743	0.9162	0.5811	0.8731
Identification	M1	RS	227.7	0.3053	0.7411	0.4673	0.5138
		HEA	178.2	0.4557	0.8339	0.3482	0.6474
	M2	RS	162.7	0.2289	0.7764	0.6117	0.6620
		HEA	170.3	0.3957	0.9439	0.5061	0.8409

same sensor mode, and the same values of parameters.

Moreover, the average number of sensors in the solutions for both the visibility-dominant and stealth-dominant missions are 15% higher than those values given in Table IV, and the cost-dominant mission triples the sensor usage. Therefore, the utility of visibility of the solution is improved due to increases in sensor quantity for all three missions. In the visibility-dominant mission, the total utility value is not affected so much due to the strong reduce on the utility of cost. The total utility value decreases for the stealth-dominant and the cost-dominant missions, since the increase rate of the utility of visibility is less than the decrease rate of the utility of cost.

In our second extension, the first injection is combined with deletion of the worst sensors. Specifically, the sensors whose utility of visibility is less than one third of the average utility of visibility of the solution are removed from the solution at every 100th generation. After the deletion operation, the sensor insertion procedure is applied as mentioned in the first technique. There is a slight increase in the total utility due to the increase on both the utility of visibility and the utility of cost values, for both the visibility-dominant and the stealth-dominant missions. However, the total utility of the

cost-dominant mission is less than the previous results.

d) The Effects of Sensor Modes: In this part, we present the effectiveness of our algorithm for detection, recognition, and identification modes of sensors on both smooth and rough terrains. Since sensors have limited range for all modes, a steep slope on the view cone of a sensor decreases the visibility range of the sensor. Therefore, the total utility value observed on a smooth terrain is higher than the total utility value on a rough terrain, when comparable number of sensors are utilized on both terrains (see Tables VI & VII). Three different sensor modes on both the visibility-dominant (M1) and the stealth-dominant (M2) missions are considered in these tables.

The performance comparison of the hybrid-EA (HEA) with a random search method (RS) is presented for all sensor modes and mission type' pairs in Tables VI and VII. The best individual in a randomly generated initial population of the HEA with 50 individuals is the output of the RS method for each test. The individual utility values and the total utility of the solutions generated by the HEA significantly outperforms those of the RS method for both smooth and rough terrains. Additionally, the required number of sensors to cover a 3-D terrain varies according to the selected sensor mode. Since,

TABLE VII
PERFORMANCE RESULTS ON A ROUGH TERRAIN WITH DIFFERENT SENSOR MODES

Sensor Mode	Mission Type	Algorithm	Number of Sensors	Utility of Visibility	Utility of Stealth	Utility of Cost	Total Utility
Detection	M1	RS	11.3	0.2062	0.7773	0.5784	0.4645
		HEA	16.3	0.4710	0.8022	0.3980	0.6545
	M2	RS	8.0	0.1221	0.8193	0.6863	0.6516
		HEA	16.3	0.4239	0.9555	0.4670	0.8561
Recognition	M1	RS	77.3	0.3321	0.6697	0.3990	0.4975
		HEA	63.0	0.4946	0.7574	0.4548	0.6638
	M2	RS	44.5	0.2075	0.7393	0.6483	0.6321
		HEA	61.3	0.4215	0.9084	0.5637	0.8385
Identification	M1	RS	214.3	0.2509	0.7697	0.4347	0.4777
		HEA	180.6	0.4022	0.8430	0.3658	0.6121
	M2	RS	141.9	0.1744	0.8025	0.6260	0.6564
		HEA	166.5	0.3406	0.9415	0.5220	0.8148

TABLE VIII
PERFORMANCE COMPARISON ON A SMOOTH TERRAIN WITH RESPECT TO COVERAGE AND STEALTH

Sensor Mode	Mission Type	Algorithm	Recognized Polygons (%) when $V(S,P,p_i)$			Observable Sensors (%) when $V_E(S,P,p_{S_i})$		
			≥ 0.1	≥ 0.5	≥ 0.75	≥ 0.1	≥ 0.5	≥ 0.75
Detection	M1	RS	0.58	0.44	0.15	0.71	0.45	0.19
		HEA	0.88	0.73	0.27	0.49	0.21	0.03
	M2	RS	0.38	0.29	0.09	0.60	0.39	0.15
		HEA	0.83	0.65	0.21	0.20	0.01	0.00
Recognition	M1	RS	0.71	0.52	0.17	0.80	0.36	0.07
		HEA	0.87	0.70	0.24	0.63	0.17	0.01
	M2	RS	0.53	0.36	0.11	0.73	0.31	0.05
		HEA	0.79	0.58	0.17	0.34	0.00	0.00
Identification	M1	RS	0.59	0.34	0.08	0.67	0.14	0.02
		HEA	0.77	0.56	0.16	0.53	0.04	0.00
	M2	RS	0.45	0.24	0.05	0.64	0.13	0.02
		HEA	0.71	0.47	0.11	0.25	0.00	0.00

the behavioral attributes of sensors including depth of view, horizontal and vertical angles for the detection mode are better than those values for the other modes, fewer sensors are required to cover in the detection mode for both smooth and rough terrains.

Performance comparison of the algorithms with respect to percentage of the polygons recognized by sensors and percentage of sensors observed by enemies are presented in Table VIII for a smooth terrain (related with the tests of Table VI) and in Table IX for a rough terrain (related with the tests of Table VII). Here, each terrain is represented by a set of triangular polygons and every terrain is generated with fixed number of polygons, which is 32768. The percentage values of the observable sensors are the averages of five different scenarios with different locational and angular behaviors of enemy sensors.

The percentage of the number of polygons observed is given for three different minimum visibility ratios (0.1, 0.5, 0.75) in the tables. Specifically, the column for $V(S, P, p_i) \geq 0.1$ gives the percentage of the polygons that are at least 10% visible by the sensors deployed. Our algorithm significantly surpasses the RS method for all sensor modes with all missions studied. As a specific example, when detection mode with the visibility-dominant mission is considered, our algorithm observes up to 80% more number of polygons on smooth terrain and up to 132% more number of polygons on rough terrain.

Additionally, the percentage of observable sensors are also given for three different visibility ratios (0.1, 0.5, 0.75) in Ta-

bles VIII and IX. Specifically, the column for $V_E(S, P, p_{S_i}) \geq 0.1$ gives the percentage of the sensors that are at least 10% visible by enemy sensors. It should be stated that the lower the percentage is the better the result. As in previous results, our method significantly outperforms the RS method with respect to the percentage of sensors observed. Our algorithm provides efficient deployments of sensors so that none of the sensors can be at least 75% visible by enemy sensors for some of the test cases given in the tables. The difference between the percentage of the observable sensors, and the difference between the numbers of recognized polygons for both of the algorithms widen when the visibility ratio increases.

Fig. 7.a and Fig. 7.b represent the coverage map of the rough and the smooth terrains given in Fig. 2.a and Fig. 2.b. The coverage maps are the output of sample runs by considering a visibility-dominant mission on detection sensor mode where $w_{vis}, w_{st}, w_{cost}$ are equal to 0.60, 0.30, and 0.10 respectively. The remaining parameters of the sample run were determined by considering the results of experiments on algorithm-specific and problem-specific parameters. Fig. 7.a gives the coverage of 14 sensors located with a total utility value of 0.7199. The utility of visibility, the utility of stealth and the utility of cost values are equal to 0.5439, 0.7925, and 0.4476, respectively. Fig. 7.b gives the coverage of 9 sensors with the total utility value 0.7571.

Weather occlusion constraint is added to the terrains given in Fig. 2 by setting the weather permeability value to 0.5. It represents a global heavy rain condition over the given terrains.

TABLE IX
PERFORMANCE COMPARISON ON A ROUGH TERRAIN WITH RESPECT TO COVERAGE AND STEALTH

Sensor Mode	Mission Type	Algorithm	Recognized Polygons (%) when $V(S, P, p_i)$			Observable Sensors (%) when $V_E(S, P, p_{S_i})$		
			≥ 0.1	≥ 0.5	≥ 0.75	≥ 0.1	≥ 0.5	≥ 0.75
Detection	M1	RS	0.34	0.25	0.09	0.64	0.36	0.13
		HEA	0.76	0.58	0.20	0.49	0.17	0.05
	M2	RS	0.21	0.14	0.05	0.63	0.31	0.12
		HEA	0.70	0.52	0.16	0.24	0.00	0.00
Recognition	M1	RS	0.58	0.38	0.13	0.77	0.30	0.05
		HEA	0.81	0.61	0.19	0.61	0.14	0.01
	M2	RS	0.38	0.24	0.07	0.70	0.24	0.03
		HEA	0.73	0.50	0.14	0.38	0.00	0.00
Identification	M1	RS	0.50	0.26	0.06	0.65	0.12	0.02
		HEA	0.72	0.48	0.13	0.52	0.03	0.00
	M2	RS	0.36	0.18	0.04	0.63	0.10	0.02
		HEA	0.63	0.39	0.09	0.25	0.00	0.00

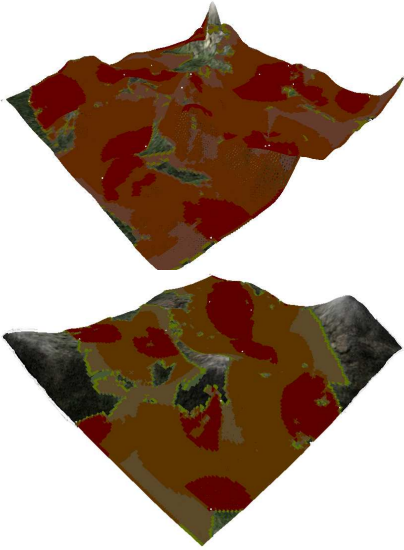


Fig. 7. Coverage of Sensors for a) Rough Terrain, and b) Smooth Terrain

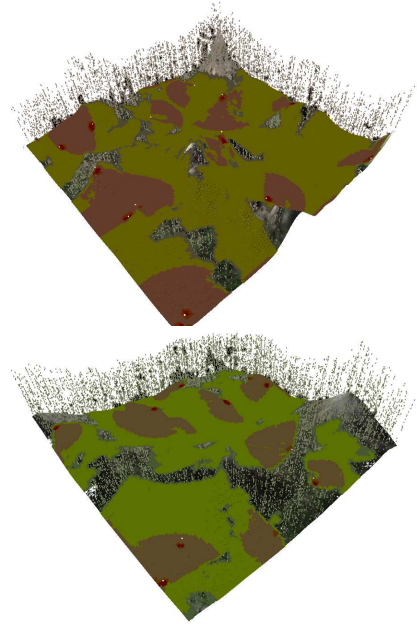


Fig. 8. Coverage of Sensors with Weather Occlusion for a) Rough Terrain, and b) Smooth Terrain

Due to the partial occlusion, the visibility of sensors in the solution and the visibility of enemy sensors decreases (see Fig. 8). For the case of rough terrain (see Fig 8.a), the utility of visibility decreases to 0.2682 and the utility of cost value is almost equal to the nonoccluded version of the same terrain. On the contrary, the utility of stealth increases to 0.9672 due to the decrease in the visibility of enemy sensors. Hence, the total utility value decreases to 0.56 which causes up to 22% change. Similar decrease is observed for the case of the smooth terrain.

V. CONCLUSIONS

This paper presents a new hybrid evolutionary algorithm (HEA) for deploying and configuring multiple sensors on a 3-D terrain by considering our multi-attribute utility-based framework [5] that combines three conflicting objectives: visibility maximization, stealth maximization, and cost minimization. The Contribution-Based Crossover (CBX) and the Proximity-Based Crossover (PBX) are two novel specialized crossover operators proposed in this study, which compromise domain specific information on sensor placement problem. We conduct an experimental study on a large set of synthetic 3-D

terrains with various characteristics. The results of experiments clearly point out the effectiveness and robustness of our HEA-based solution under all values of several algorithm- and problem-specific parameters.

ACKNOWLEDGMENTS

This research was supported by The Scientific and Technological Research Council of Turkey (TUBITAK) with a research grant (No. 106E159). Additionally, parts of the computations have been carried out by using UYBHM at ITU through a grant (No. 20432008). A preliminary version of this paper was published in the Proceedings of 2009 IEEE International Conference on Intelligent Systems Design and Applications (ISDA) [20]. The authors would like to thank to anonymous referees for their helpful comments and suggestions to improve this manuscript.

REFERENCES

- [1] E. Riseman, H. Schultz, W. Lim, B. Draper, and A. Hanson, "Deadalus battlefield visualization system," in *Proc. ARPA Image Understanding*, 1996.
- [2] J. Wang, C. Niu, and R. Shen, "Priority-based target coverage in directional sensor networks using genetic algorithm," *Computers and Mathematics with Applications*, Article in Press.
- [3] M. Marengoni, B. A. Draper, A. Hanson, and R. Sitaraman, "A system to place observers on a polyhedral terrain in polynomial time," *Image and Vision Computing*, vol. 18, no. 10, pp. 773–780, 2000.
- [4] N. Tezcan and W. Wang, "Self-orienting wireless multimedia sensor networks for occlusion viewpoints," *Computer Networks*, vol. 52, pp. 2558–2567, 2008.
- [5] H. Topcuoglu, M. Ermis, and M. Sifyan, "Positioning and utilizing sensors on a 3d terrain: Part 1 - theory and modelling," *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, Accepted for publication.
- [6] T. Back, D. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*. New York: Inst. Physics Publishing and Oxford Univ. Press, 1997.
- [7] T. Back, *Evolutionary Algorithms in Theory and Practice*. Oxford Univ. Press, New York, 1996.
- [8] A. Eiben and J. Smith, *Introduction to Evolutionary Computing*. Springer Verlag, 2003.
- [9] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.
- [10] Z. Michalewicz and D. B. Fogel, *How to Solve It: Modern Heuristics*. Springer-Verlag, 2000.
- [11] B. R. Fox and M. B. McMahon, "Genetic operators for sequencing problems," in *Proceedings of the First Workshop on Foundations of Genetic Algorithms and Classifier Systems*, G. Rawlins, Ed. Morgan Kaufmann, 1990.
- [12] J. J. Grefenstette, "Incorporating problem specific knowledge into genetic algorithm," in *Genetic Algorithms and Simulated Annealing*, L. Davis, Ed. Morgan Kaufmann Publishers, 1987.
- [13] P. Bonissone, R. Subbu, N. Eklund, and T. R. Kiehl, "Evolutionary algorithms + domain knowledge = real-world evolutionary computation," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 256–280, 2006.
- [14] P. Moscato, "On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms," Caltech Concurrent Computation Program Report, Tech. Rep. 826, 1989.
- [15] ——. Memetic algorithms' home page. [Online]. Available: www.ing.unlp.edu.ar/cetad/mos/memetic_home.html
- [16] N. J. Radcliffe and P. D. Surry, "Formal memetic algorithms," in *Evolutionary Computing, AISB Workshop*, ser. Lecture Notes in Computer Science, T. C. Fogarty, Ed., vol. 865. Springer, 1994, pp. 1–16.
- [17] B. Nystrom. (2000) Terrain generation tutorial: Hill algorithm. [Online]. Available: <http://www.robot-frog.com/3d/hills/index.html>
- [18] A. Fournier, D. Fussel, and L. Carpenter, "Computer rendering of stochastic models," *Communications of the ACM*, vol. 25, no. 6, pp. 371–384, 1982.
- [19] D. Montgomery, *Design and Analysis of Experiments*. John Wiley-Sons Inc., 2005.
- [20] H. Topcuoglu, M. Ermis, and M. Sifyan, "Hybrid evolutionary algorithm for sensor placement on a 3-d terrain," in *Proceedings of IEEE International Conference on Intelligent Systems Design and Applications*, 2009.

Hybrid Evolutionary Algorithms for Sensor Placement on a 3D Terrain

Haluk Rahmi Topcuoglu
Computer Engineering Department,
Marmara University,
Kuyubasi, 34722, Istanbul, Turkey
Email: haluk@marmara.edu.tr

Murat Ermis
Industrial Engineering Department,
Turkish Air Force Academy,
Yesilyurt, 34149, Istanbul, Turkey
Email: m.ermis@hho.edu.tr

Mesut Sifyan
Computer Engineering Department,
Marmara University,
Kuyubasi, 34722, Istanbul, Turkey
Email: mesutsifyan@marun.edu.tr

Abstract—In this paper, we propose a framework for deploying and configuring a set of given sensors in a synthetically generated 3-D terrain with multiple objectives on conflicting attributes: maximizing the visibility of the given terrain, maximizing the stealth of the sensors and minimizing the cost of the sensors used. Because of their utility-independent nature, these complementary and conflicting objectives are represented by a multiplicative total utility function model, based on multi-attribute utility theory. In addition to theoretic foundations, this paper also present a hybrid evolutionary algorithm based technique to solve the sensor placement problem. It includes specialized operators for hybridization, which are problem-specific heuristics for initial population generation, intelligent variation operators which comprise problem specific knowledge, and a local search phase. The experimental study validates finding the optimal balance among the visibility, the stealth and the cost related objectives.

Keywords—Sensor planning; multi-attribute utility theory; hybrid genetic algorithms.

I. INTRODUCTION

In the military operations, reconnaissance, surveillance and target acquisition can include a plurality of sensor platforms that are used to collect information about an area under surveillance and play a vital role [1]. In order to detect position of foes, some sensors should be placed to cover a certain terrain to provide maximum visibility while maintaining sensors' stealth.

In this paper, we develop a framework and a novel solution approach for determining the optimal number of sensors, locating and setting their orientational sensor-specific parameters in a synthetically generated 3-D terrain with multiple objectives. Our solution approach relies on rational trade-off between three conflicting objectives which are maximizing the coverage area while maintaining the maximum stealth, and minimizing the total acquisition cost of deploying the sensors.

Motivated by our constructed framework, this paper explores the employing of a hybrid evolutionary algorithm for sensor placement and orientation problem. Simple evolutionary algorithms are generally poor for solving the complex combinatorial problems [2]. GAs are usually strengthen with the domain-specific characteristics [3], [4], [5], and they are combined with specialized heuristics to produce hybrid systems, which are called with different names including hybrid evolutionary algorithms and memetic algorithms [6].

In this paper, we propose two specialized crossover operators called the Contribution-Based Crossover (CBX) and the Proximity-Based Crossover (PBX) that comprise the domain specific information on sensor placement problem, and a local search technique for improving the quality of solution. Experiments on synthetic 3-D terrains with various characteristics are conducted in order to present the effectiveness of our GA-based framework. The results of the experimental study clearly show that our proposed approach is very successful in deploying and utilizing sensors by considering the multiple objectives.

The remainder of the paper is organized as follows: In Section 2, we first present an overview to Multiple Attribute Utility (MAU) theory, which is followed by our novel multi-attribute utility function model and its sub-objective formulations. Section 3 gives our hybrid GA-based formulation for solving the sensor optimization problem. Performance evaluation and experimental study is discussed in Section 4; and Section 5 concludes the paper.

II. MULTI-ATTRIBUTE UTILITY FUNCTION FOR SENSOR OPTIMIZATION PROBLEM

Utility analysis is a widely preferred multi-objective optimization method, since it enables, especially in military applications, the testing of various scenarios (such as risk averse, risk prone, etc.) under uncertainty or limited data cases. If there is more than one objective, utility analysis gets very complicated and can only be applied if certain decomposition conditions (additivity, independency, etc.) are met [7], [8].

If there is more than one objective which are both conflicting and at the same time supporting each other and if all objectives are independent, multiplicative or multi-linear utility function may be used. Our basic assumption states that in a military operation, the perception behavior of the sensors should rely on rational trade-off between three conflicting criteria (priorities). These three criteria are: maximizing the information about the land, minimizing the data collected by the enemy and minimizing the total cost of the positioned sensors.

In our formulation, the total utility value is computed as a multiplicative function over the given three attributes. Formally, the total function to maximize, $U(A, S, P)$, of scanning an area A using a set S of sensors which are located on a set

P of polygons (i.e. the sensor s_i located on the polygon p_k), is represented with the following formulation:

$$U(A, S, P) = w_{vis} U_{vis}(A, S, P) + w_{st} U_{st}(A, S, P) + w_{cost} U_{cost}(S, P) + w_{vis} w_{st} U_{vis}(A, S, P) U_{st}(A, S, P) + w_{vis} w_{cost} U_{vis}(A, S, P) U_{cost}(S, P) + w_{st} w_{cost} U_{st}(A, S, P) U_{cost}(S, P) + w_{vis} w_{st} w_{cost} U_{vis}(A, S, P) U_{st}(A, S, P) U_{cost}(S, P) \quad (1)$$

where $U_{vis}(A, S, P)$ is the utility of visibility of area A by the set of sensors S located on the set of polygons P ; $U_{st}(A, S, P)$ is the utility of stealth of the set of sensors S located on set of polygons P and $U_{cost}(S, P)$ is the utility of the cost of the sensors S located on set of polygons P . In this equation, w_{vis} , w_{st} , w_{cost} are the weights (coefficients) of visibility, stealth and cost utility functions, respectively, where $0 \leq w_{vis}, w_{st}, w_{cost} \leq 1$ and $w_{vis} + w_{st} + w_{cost} = 1$. These weights are set based on experimentation on a given terrain by considering various military scouting missions.

Additionally, we also consider the total utility of each sensor s_j located on polygon p_k in our computations, which is represented by $U^j(A, s_j, p_k)$. By using the Equation 1, this term requires $U_{vis}^j(A, s_j, p_k)$, $U_{st}^j(A, s_j, p_k)$ and $U_{cost}^j(A, s_j, p_k)$ terms, which are the sensors-specific utility of visibility, stealth and cost, respectively.

A. Computing the Utility of Visibility

The value of utility of visibility is derived by using the amount of visibility of the terrain, which is computed by adding the visibility of all polygons on it. Formally, the utility of visibility of area A by the set of sensors S (located at set of polygons P), $U_{vis}(A, S, P)$ is computed using the Equation 2,

$$U_{vis}(A, S, P) = \frac{\sum_{p_i \in A} V(S, P, p_i) \times W_{p_i}}{\sum_{p_i \in A} W_{p_i}} \quad (2)$$

where, W_{p_i} is the weight of the polygon p_i , which indicates the importance of the polygon, and $V(S, P, p_i)$ is the visibility value of polygon p_i by using the set of sensors S located on the set of polygons P , which is computed by the average of visibility of the points on polygon p_i . Since any point can be recognized by multiple sensors (with different visibility values), maximum visibility of the point is considered in computing the visibility value of polygon p_i . In our study, four points, i.e., the three corner points and the center of mass, are considered as the selected polygon points. The term $V_{s_j}(s_j, p_k, b)$ is the visibility of the destination point b (which can be one of the four points of polygons) from the sensor s_j located at source point a (which is the center of mass of polygon p_k). This term formally defined by Equation 3,

$$V_{s_j}(s_j, p_k, b) = (1 - \eta_{s_j} \times \frac{D(a, b)}{\Delta_{s_j}}) \times (1 - \max_{p_c \in a \rightarrow b} \psi_c^W) \times (1 - \max_{p_c \in a \rightarrow b} \psi_c^O) \quad (3)$$

where $D(a, b)$ is the distance between point a and point b ; Δ_{s_j} is the depth of view and η_{s_j} is the range effect

coefficient of sensor s_j . The range values varies with respect to different types of the problem addressed; i.e., there will be three different range values (for detection, recognition, identification) of each sensor.

Weather density (ψ_i^W) and object density (ψ_i^O) of polygon p_i , where their values are in the range $0 \leq \psi_i^W, \psi_i^O \leq 1$, are used to compute the permeability value of a ray that traverses from an origin to a predefined destination through the given polygon p_i by considering object and weather conditions over the polygon.

The second term is the *weather permeability value*, which is derived by the density values of weather conditions. The term ψ_c^W is the weather density over a polygon p_c where p_c is a polygon that is in between point a and point b . The density values of weather conditions over all polygons in between points a and b are considered as part of LOS algorithms ([9], [10]) and the maximum value is returned if no intermediate point (between a and b) is obstructed by terrain.

The last term in Equation 3 is the *object permeability value*, which is set by using the density value of objects. There can be two types of objects located on synthetically generated terrain, which are natural objects such as trees and artificial objects such as buildings. It should be noted that the values of weather density (ψ_c^W) and object density (ψ_c^O) of polygon p_c are in the range $0 \leq \psi_c^W, \psi_c^O \leq 1$.

B. Determining the Utility of Stealth

The utility of stealth value for a set of sensors that are already located on the terrain is derived by subtracting the cost of the total visibility of the located sensors (by using enemy or opponent objects) from one. For this purpose, a predefined number of opponent objects of different types are scattered across the terrain randomly (by utilizing angle and distance constraints), as part of a given scenario m . These objects are the vehicles carrying opponent sensors. In our experimental study, the angle-based locational attributes (such as viewing angle, depth of view etc.) of opponent sensors can be set with those values of either best or worst sensor in our system.

The utility of stealth of a set of S sensors that are positioned on a set of polygons P by considering r different scenarios (for setting the opponent objects) is computed with the following equation

$$U_{st}(A, S, P) = \frac{\sum_{x=1}^r (1 - \sum_{s_i \in S} V_E(E, P, p_{s_i}) \times R_U(A, s_i, p_{s_i}))}{r} \quad (4)$$

where $V_E(E, P, p_{s_i})$ is the maximum visibility of the sensor s_i (located on polygon p_{s_i}) from the set of opponent vehicles E which are located on set of positions P . As in the sensor visibility, it is computed by,

$$V_E(E, P, p_{s_i}) = \max_{e_j \in E} \{V_E(e_j, p_k, p_{s_i})\} \quad (5)$$

The $V_E(e_j, p_k, p_{s_i})$ term in this equation is the visibility of a single point on the terrain (where the sensor s_i is located) from an opponent sensor located on polygon p_k . Here, this term is the dual of the term $V_S^P(s_j, p_k, b)$; therefore it is also computed with the Equation 3.

The $R_U(A, s_i, p_{s_i})$ term is the ratio of the utility of visibility of the given sensor s_i to the cumulative utility of visibility of all sensors, as shown in the following equation.

$$R_U(A, s_i, p_{s_i}) = \frac{U_{vis}(A, s_i, p_{s_i})}{\sum_{s_j \in S} U_{vis}(A, s_j, p_{s_j})} \quad (6)$$

It should be noted that the denominator in the previous equation is not the overall utility of visibility, but it is the cumulative utility of visibility. It is due to the fact that overall utility of visibility may be even equal to utility of visibility of a single sensor, which may generate a negative value in Equation 4. If a sensor with a high utility of visibility value (and therefore a high utility ratio) is completely seen by an opponent object, this will significantly decrease the utility of stealth.

The locational attributes (polygon number, heading and tilt angles, sensor types) of $r * |E|$ opponent sensors are set at the beginning of the program execution, where $|E|$ is the set of opponents considered at each scenario.

C. Computing the Utility of Cost

In our study, the term cost of a given sensor s_i includes two separate meanings a) the (normalized) financial cost of sensor, $NCost_F(s_i)$, and b) the (normalized) placement cost of sensor to its current location (i.e., polygon k), which is represented by $NCost_L(s_i, p_k)$. Based on these terms, the utility of cost for a set of sensors S placed on a set of locations P is formally defined by

$$U_{cost}(S, P) = 1 - \left\{ \frac{\sum_{s_i \in S} (\omega_F \times NCost_F(s_i))}{|S|} + \frac{\sum_{s_i \in S} (\omega_L \times NCost_L(s_i, p_k))}{|S|} \right\} \times N|S|. \quad (7)$$

It should be noted that financial and placement (locational) costs are independent, and ω_F and ω_L are the weights (from the range $[0..1]$) of the two cost terms, respectively. In our experiments, $\omega_F = 0.7$ and $\omega_L = 0.3$, unless otherwise specified. The placement cost term in the right side of this equation is computed by $NCost_L(s_i, p_k) = P_L(s_i, p_k) \times \varphi_{s_i}$ where $P_L(s_i, p_k)$ is the locating probability of sensor s_i on polygon p_k ; and φ_{s_i} is a sensor specific constant that is set to 1, unless otherwise specified. The locating probability values of all terrain polygons are set by considering the heights and slopes of polygons and characteristics of the sensors (i.e., sensor carriers). In order to simplify the model, this term can be set with the weights of polygons based on the second method explained in Section II-A. In Equation 7, $N|S|$ is the normalized value of sensors usage, which is computed by

$$N|S| = \frac{|S|}{E(S)}, \quad (8)$$

where $|S|$ is the number of sensors used in the solution, and $E(S)$ is the expected value of the number of sensors for the given terrain, which is equal to the mean value of upper and lower limits of sensor usage.

III. HEA-BASED SENSOR PLACEMENT

In order to build hybrid evolutionary algorithms (HEA), there are various methods to incorporate specialized operators and domain specific knowledge with evolutionary algorithms. We consider problem-specific heuristics for the initial population generation. There are novel intelligent variation operators presented in our study, such as Contribution-Based Crossover and Proximity-Based Crossover operators that incorporate problem specific knowledge. Additionally, a local search phase is applied on the output of the variation operators.

In our GA-based approach, each solution contains the type and the locational attributes (position, heading angle, tilt angle) of sensors. There is no restriction on sensor quantity; therefore string representation supports variable-length chromosomes. Additionally, this work has no restriction on the order of sensors in a given solution. A steady-state Genetic Algorithm is applied which generates one individual at each iteration. We consider tournament selection mechanism and the tournament size is varied in the experiments. The value of the fitness function is set with the total utility value.

A. Initial Population Generation

The first phase to generate a solution for initial population is to determine the number of sensors considered in the solution, which is set randomly between the upper and the lower limit of the sensor quantity. The sensor mode (detection, recognition, identification) and the horizontal depth of view (Δ) on the given sensor mode are considered in order to calculate the limits. After the sensor quantity of a solution is determined, the type of each sensor is set randomly by preserving the inverse proportionality to sensor capacities (i.e., the view ranges with respect to given sensor mode). Then, locations of sensors in each solution are determined with a heuristic which aims to distribute the selected sensors evenly on the given terrain.

B. Crossover Operation

Our GA-based approach includes a set of variation operators (crossover and mutation operators) and a local search phase that is applied in between crossover and mutation operators. In our study we consider three different crossover operators, which are Contribution-Based Crossover (CBX), Proximity-Based Crossover (PBX) and Cut and Splice Crossover (CSX). It should be note that CBX and PBX are the ones that consider problem-specific knowledge. All of the three crossover operators generate a single offspring as the output.

1) *Contribution Based Crossover (CBX) Operator:* The main idea behind this operator is to carry a sensor to the offspring from one of the parents based on its contribution, which is expressed in terms of utility of the sensor. The sensors in both parents are examined in this operator. One sensor is selected randomly from each parent and the better one, which has higher total utility value than the other sensor, is moved into the offspring as the first sensor of the offspring. Then, at each step, one of the remaining sensors from each parent is selected and the best one of selected sensors is moved to the offspring if the sensor keeps acceptable proximity

distance with all sensors that are already in the offspring. The calculation steps of acceptable proximity distance is explained in the following part.

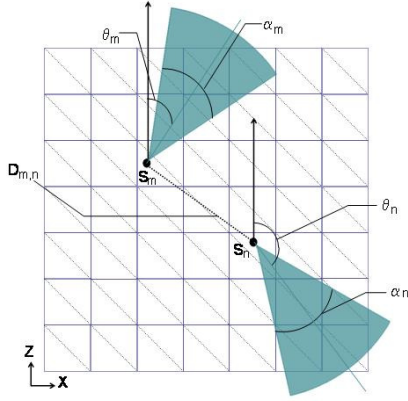


Fig. 1. Contribution Based Crossover.

Assume that the sensor s_m is the sensor which has the highest total utility value (highest contribution value) among the two sensors that are selected from the remaining sensors of each parent solution; and assume that s_m is already located on polygon m with the heading angle θ_m as given in Figure 1. Assume that s_n is one of the sensors which is already copied to offspring; and s_n is located on polygon n with the heading angle θ_n . Assume that (horizontal) viewing angles of s_m and s_n are α^m and α^n , respectively. In order to decide whether sensor s_m located on polygon m will be accepted for the offspring, the following two inequalities are considered.

$$D(n, m) > (\Delta_n + \Delta_m) \times \mu \quad (9)$$

$$|\theta^m - \theta^n| > \frac{\alpha^m + \alpha^n}{2} \times \kappa \quad (10)$$

The right side of the first inequality is the acceptable proximity distance. The distance between sensors should be equal or greater than the acceptable proximity distance, which is computed by multiplying the summation of depth of views with a problem specific constant, μ . The depth of view value of each sensor is based on the sensor mode considered in the experiments.

If s_m validates the first inequality, it is selected for the offspring; then, the second inequality is not considered any more. Similarly, if it fails in the first inequality but it validates the second one, the sensor is still moved to the offspring. If the angular distance between heading angles of sensors (the left side in the second inequality) is greater than the summation of half of horizontal viewing angles of two sensors, the candidate sensor is copied since they are looking at different directions even if the distance between them is less than the acceptable proximity distance. The term κ , an angular constant related with overlapping ratio of horizontal viewing angles of sensors, is set to 0.8 unless otherwise specified.

If it does not validate both of the inequalities, the candidate sensor is dropped and sensor with the next highest total utility value is considered. This process is repeated as long as the number of sensors ($|S|$) moved to the offspring is less than the upper limit, which is computed by using the following inequality.

$$|S| \leq (|S_1^P| + |S_2^P|) * \epsilon \quad (11)$$

In this inequality, $|S_1^P|$ and $|S_2^P|$ are the number of sensors allocated in the first and the second parents of the offspring, respectively; and ϵ is a constant which is less than 1.

2) *Proximity Based Crossover (PBX) Operator*: In this operator, firstly, a crossover point k is selected randomly by considering $1 \leq k \leq m - 1$ where m is the minimum of the sensor quantities of the two parent solutions. Starting from the first sensor, k sensors of the first parent are copied to the offspring. The next phase is to copy sensors from the second parent. Starting from the first element of the second parent, sensors are copied to the offspring in the order by considering the inequalities given in Equation 10. The first equation is for testing the proximity distance; and if the current sensor of the second parent passes this test for all sensors already in the offspring, it is copied to the offspring. Otherwise the second test related with viewing angle is applied.

3) *Cut and Splice (CSX) Operator*: This operator is similar to the original single-point crossover operator proposed in the literature. As in the previous operator, a crossover point is selected randomly from the range $[1..m - 1]$ where m is the minimum of number of sensors exist in two parents. Then, both parents are separated at the given crossover point, and two children are created by exchanging the tails; and the better one is selected as offspring. The tests of proximity distance and viewing angle are not applied in this operator, and there is no upper bound for number of sensors considered.

C. Local Search Phase

After a solution is generated using the crossover operator, the local search phase targets on improving the quality of the solution by modifying angular attributes of sensors. At each iteration of the local search the sensor with the minimum utility of visibility value is selected.

The heading angle is updated by adding a predefined increment amount (δ_H) repeatedly until the first improvement on utility of visibility for the given sensor. If there are large number of polygons on the terrain, δ_H can be low; otherwise a higher value can be set. The next phase is to modify the tilt angle of the sensor in order to improve sensor's visibility. An increment amount (δ_T) is added to the tilt angle of the given sensor repeatedly. This process is repeated until all possible alternatives covered.

This process is repeated until one of the following two conditions occurs: a) the cumulative improvement ratio on the utility of visibility of the modified sensors is greater than or equal to 20%, b) at a least 10% of sensors in the original solution is considered as part of the local search process. Only local improvements of sensors are considered and number of

sensors to be examined is limited in order to bound the running time of this phase.

D. Mutation Operators

Mutation operators in our study can be considered in two categories, which are the locational mutation operators, and the angular mutation operators.

1) *Locational Mutation Operators*: There are three types of mutation operators in this category which deal with the locations of sensors: the *update*, the *delete* and the *insert* operators. The other parameters of our experimentation is set with the default values.

The first operator updates the location of a sensor in two steps: i) selecting the sensor, and ii) selecting the new location. The sensor for the update operator is selected either randomly or it is the one which has the minimum utility of visibility value. On the other hand, the new location is selected randomly either from the whole terrain, or from the same region where the original sensor is located on. The second operator deletes one of the sensors, which can be either a randomly selected sensor or the one which has the minimum utility of visibility value. The last operator in this category adds a new sensor to the solution on a polygon that is selected randomly either from the whole terrain or from the region which already includes the sensor with the minimum utility of visibility value. The sensor type and the heading and the tilt angles of the new sensor is set randomly by considering the feasibility ranges of angles.

2) *Angular Mutation Operator*: This category is to update the angular properties of a sensor, which is either selected randomly or the sensor that minimizes the utility of visibility value. There are two operators in this category, which are *mutation on the heading angle* and *mutation on the tilt angle*. The former one updates the heading angle of the sensor randomly, and the latter one updates the tilt angle randomly from a predefined feasible range. One extension is to select one of the two mutation operators from this class non-uniformly. Specifically, mutation on the heading angle can be selected with a higher probability than mutation on the tilt angle.

In addition to these two classes of mutations, updating the sensor type is also considered as a mutation operator. In our experimental study, type of a randomly selected sensor is updated randomly at every 100 generations; if the update causes an improvement, then it is accepted; otherwise it is rejected. The update on the sensors type is applied for both of the mutation classes given above.

IV. EXPERIMENTAL STUDY

In this section, we present the results of experiments that evaluate the effectiveness of our algorithm. The experiments in this study were performed on a cluster of PCs, each of which has Intel Xeon 2.33GHz processor running Linux operating system. The default values of general parameters in our hybrid evolutionary algorithm listed in Table I are considered.

TABLE I
DEFAULT SETTINGS OF SELECTED PARAMETERS OF EXPERIMENTATION

Parameter	Value
-Sensor Allocation Probability in a Terrain Region (center, random)	(20%, 80%)
-Coefficient for Depth of View in Crossover Operators	$\mu = 0.4$
-Coefficient for Sensor Quantity Inherited from Parents to Offspring	$\epsilon = 0.5$
-Improvement Ratio in Local Search Phase	$I_R = 20\%$
-Sensor Ratio in Local Search Phase	$SR = 10\%$
-Number of Enemies (detection, recognition and identification modes)	(10, 40, 100)
-Tournament Size	5
-Number of Generations for Termination	1000

The first set of experiments is for identifying the values of GA-parameters, which are the population size, the type of the crossover operator, the type of the mutation operator and the sensor selection criteria in mutation. The population size is assigned from the set $\{30, 50, 100\}$. Three crossover operators and two categories of mutation operators given in Section III are considered in the experiments. Additionally, the source sensors for mutation is selected either randomly or the one which has minimum visibility. The combination of those parameters constructs 54 different cases, each of which is run with 30 replications. Therefore, a total of 1620 tests are conducted in the first set of experiments. On the other hand, the other parameters of our experimentation is set with the default values. Specifically, initial population of our algorithm is set randomly without any heuristic. The increment amount of heading angle (δ_H) is set to 25° , which is one-fourth of the minimum horizontal viewing angle of sensors considered. The increment amount of the tilt angle is set to $\delta_T = 3^\circ$. A visibility-dominant mission by selecting recognition mode of sensors with a rough terrain is considered.

The optimization results are obtained by setting the goals for the total utility value and generating the optimal conditions. When the results of experiments are examined, the population size is set to 50, the contribution-based crossover is selected, and the sensors with the minimum visibility value is considered for the mutation operator. However, the mutation operator does not have a significant effect on total utility, when a visibility-dominant mission is performed.

The second set of experiments evaluates the effects of the following parameters on the performance, which are the strategy for selecting the initial population, increment values for the heading and the tilt angles, the mission-specific utility weights and the terrain type. Initial population is set either randomly or based on the heuristic presented in Section III. The increment value in heading and tilt angles are assigned from the sets $\{25, 50, 75\}$ and $\{3, 5, 7, 10\}$, respectively. Three different missions (visibility-dominant, stealth-dominant and cost-dominant cases) and two terrain types (smooth and rough terrains) are taken into account in the experiments. The coefficients ($w_{vis}, w_{st}, w_{cost}$) given in Equation 1 are set with $\{0.6, 0.3, 0.1\}$, $\{0.3, 0.6, 0.1\}$, and $\{0.25, 0.25, 0.5\}$ for visibility-dominant, stealth-dominant and cost-dominant

missions, respectively. Based on the analysis of variance, it is found that the first three parameters do not significantly affect the total utility values. Both the terrain type and the mission type affect the quality of solutions as expected.

A. Performance Evaluation and Discussion

In this part, we present the effectiveness of our algorithm for detection, recognition and identification modes of sensors on both smooth and rough terrains. Since sensors have limited range for all modes, a steep slope on the view cone of a sensor decreases the visibility range of the sensor. Therefore, the total utility value observed on a smooth terrain is higher than the total utility value on a rough terrain, when comparable number of sensors are utilized on both terrains (see Tables II & III). Three different sensor modes on both the visibility-dominant (M1) and the stealth-dominant (M2) missions are considered in these tables.

TABLE II
PERFORMANCE RESULTS ON A SMOOTH TERRAIN WITH DIFFERENT SENSOR MODES

Sensor Mode	Mission Type	Algorithm	Number of Sensors	Total Utility
Detection	M1	RS	13.9	0.5267
		HEA	14.7	0.7419
	M2	RS	9.7	0.6894
		HEA	14.7	0.9216
Recognition	M1	RS	80.7	0.5524
		HEA	59.7	0.7061
	M2	RS	54.9	0.6465
		HEA	62.3	0.8731
Identification	M1	RS	227.7	0.5138
		HEA	178.2	0.6474
	M2	RS	162.7	0.6620
		HEA	170.3	0.8409

The performance comparison of the HEA with a random search method (RS) is presented for all sensor modes and mission types pairs in Tables II and III. The best individual in a randomly generated initial population of the HEA with 50 individuals is the output of the RS method for each test. The individual utility values and the total utility of the solutions generated by the HEA are significantly outperforms those of the RS method for both smooth and rough terrains. Additionally, the required number of sensors to cover a 3-D terrain varies according to the selected sensor mode. Since, the behavioral attributes of sensors including depth of view, horizontal and vertical angles for the detection mode are better than those values for the other modes, fewer sensors are required to cover in the detection mode for both smooth and rough terrains.

V. CONCLUSIONS

Positioning and utilizing multiple sensors for acquisition of a given environment is one of the fundamental research topics in various domains including military operations, computer vision and robotics. The contributions of this paper can grouped

TABLE III
PERFORMANCE RESULTS ON A ROUGH TERRAIN WITH DIFFERENT SENSOR MODES

Sensor Mode	Mission Type	Algorithm	Number of Sensors	Total Utility
Detection	M1	RS	11.3	0.4645
		HEA	16.3	0.6545
	M2	RS	8.0	0.6516
		HEA	16.3	0.8561
Recognition	M1	RS	77.3	0.4975
		HEA	63.0	0.6638
	M2	RS	44.5	0.6321
		HEA	61.3	0.8385
Identification	M1	RS	214.3	0.4777
		HEA	180.6	0.6121
	M2	RS	141.9	0.6564
		HEA	166.5	0.8148

in two-folds. Firstly, we present a novel multi-attribute utility-based framework for deploying and configuring multiple sensors in a 3D terrain that combines three conflicting objectives into a unified total utility function, which are maximizing the coverage area while maintaining the maximum stealth and minimizing the total acquisition cost of deploying sensors. Secondly, this paper presents a new hybrid evolutionary algorithm (HEA) based solution for the constructed framework. The computational study clearly points out the effectiveness and robustness of our HEA-based solution under various values of several experimental parameters.

ACKNOWLEDGMENTS

This research was supported by The Scientific and Technological Research Council of Turkey (TUBITAK) with a research grant (no. 106E159). Additionally, parts of the computations have been carried out by using UYBHM at ITU through a grant (20432008).

REFERENCES

- [1] E. Riseman, H. Schultz, W. Lim, B. Draper, and A. Hanson, "Deadalus battlefield visualization system," in *Proc. ARPA Image Understanding*, 1996.
- [2] Z. Michalewicz and D. B. Fogel, *How to Solve It: Modern Heuristics*. Springer-Verlag, 2000.
- [3] B. R. Fox and M. B. McMahon, "Genetic operators for sequencing problems," in *Proceedings of the First Workshop on Foundations of Genetic Algorithms and Classifier Systems*, G. Rawlins, Ed. Morgan Kaufmann, 1990.
- [4] J. J. Grefenstette, "Incorporating problem specific knowledge into genetic algorithm," in *Genetic Algorithms and Simulated Annealing*, L. Davis, Ed. Morgan Kaufmann Publishers, 1987.
- [5] G. Raidl, "Hybrid evolutionary algorithms for combinatorial algorithms," Habilitation Thesis, Vienna University of Technology, Vienna, Austria, 2002.
- [6] P. Moscato. Memetic algorithms' home page. [Online]. Available: www.ing.unlp.edu.ar/cetad/mos/memetic_home.html
- [7] R. Keeney and H. Raiffa, *Decision with Multiple Objectives: Preferences and Value Tradeoffs*. New York: John Wiley and Sons, 1993.
- [8] J. Dyer, "Remarks on the analytic hierarchy process," *Management Science*, vol. 36, no. 3, pp. 249–259, Mar. 1990.
- [9] M. D. Proctor and W.J.Gerber, "Line-of-sight attributes for a generalized application program interface," *the Journal of Defense Modeling and Simulation*, vol. 1, no. 1, pp. 43–57, 2004.
- [10] D. Handerson, "Modterrain: A proposed standart for terrain representation in entity level simulation," Master's Thesis, Naval Postgraduate School, California, 1999.

A New 3-D Wireless Multimedia Sensor Network Simulation Environment for Connected Coverage Problems

Haluk Topcuoglu

Computer Engineering Department
Marmara University
Kuyubasi, 34722, Istanbul, Turkey
e-mail: haluk@marmara.edu.tr

Murat Ermis

Industrial Engineering Department
Turkish Air Force Academy
Yesilyurt, 34149, Istanbul, Turkey
e-mail: m.ermis@hho.edu.tr

Ilker Bekmezci

Computer Engineering Department
Turkish Air Force Academy
Yesilyurt, 34149, Istanbul, Turkey
e-mail: i.bekmezci@hho.edu.tr

Mesut Sifyan

Computer Engineering Department
Marmara University
Kuyubasi, 34722, Istanbul, Turkey
e-mail: mesutsifyan@marmara.edu.tr

(submitted to: Simulation: Transactions of the Society for Modeling and Simulation International)
(2nd Round Revision)

Abstract

Wireless Multimedia Sensor Network (WMSN) is a network of wirelessly interconnected sensors that can gather multimedia information, like sound and vision. One of the most important design issues of WMSN is connected coverage. Although there are many studies about coverage for WMSN, most of them are based on two dimensional terrain assumptions. However, especially for outdoor applications, 3D terrain structure affects the performance of WMSN remarkably. In this paper, a novel 3D WMSN simulation environment for connected coverage issues is presented. There are four main modules of the simulation environment. The Terrain Generator (TerGen) generates a synthetic 3D landscape with different weather conditions (snow, rain, fog), object occlusions (artificial or natural objects) and toughness levels (smooth or rough). The Scenario Editor (SenEd) is used to define various sensor types that have various behavioral and locational attributes. The outputs of TerGen and SenEd are the inputs of the Simulator Engine (SimEn) which simulates the WMSN and gives the performance results. The Optimization Module (OptMod), which is optional, can be used to determine the location of the sensors optimally, while satisfying some constraints. Different scenarios are simulated to show the capabilities of the simulation environment. The performance results show that 3D terrain structure affects the coverage performance of WMSN directly. The object occlusions and weather conditions are also very important for WMSN coverage.

Keywords: *WMSN, connected coverage, simulator*

A New 3-D Wireless Multimedia Sensor Network Simulation Environment for Connected Coverage Problems

1. INTRODUCTION

In recent years, with the pace of the developing micro-electro-mechanical systems (MEMS) technology, it has been possible to integrate battery operated sensor, computing power and low power wireless communication components into one small size device. Wireless sensor networks (WSN) are composed of a large number of sensor nodes, which are densely deployed either inside the phenomenon or very close to it [1]. Wireless multimedia sensor networks (WMSN), which can be considered as a new type of WSN, is a network that consists of wireless sensor network devices that have ability to retrieve, process and store multimedia data such as video, audio and images [2]. Connected coverage is one of the most important design considerations of WMSN systems. WMSN nodes should be deployed so that while the covered area is maximized, the nodes of WMSN must be able to form a connected network [3]. In this paper, a new 3D WMSN simulator for connected coverage problems is presented.

Connected coverage concept is firstly investigated for wireless sensor networks (WSN). Huang and Tseng have presented an extensive survey about WSN connected coverage problems [4]. Several simulators are developed and used for connected coverage issues for WSN. Some of them are general purpose network simulators which have an extension for WSN and some of them are developed specifically for WSN. OPNET [5], ns-2 [6], GloMoSim [7] are examples for general purpose network simulators. TOSSIM [8], EmStar [9] and SENSE [10] are examples for WSN simulators. There are a number of WSN simulator tool surveys in the literature. [11-13].

Although WSN and WMSN are similar, WMSN has its own unique behaviors for coverage [3]. While scalar sensors can be modeled as circle, multimedia sensors are modeled as sector. This is why the solutions for WSN connected coverage problems cannot be applied to WMSN directly. The coverage problem for video sensors is an old and well-known problem [14]. However, connected coverage issues for video sensors take place along with the development of WMSN. To the best of our knowledge, the only study about connected coverage in WMSN is presented in [15]. In this paper, Han *et al.* propose the problems of deploying a minimum number of directional sensors to form a connected network to cover and two approximation solutions.

The terrain structure of the WMSN affects the connected coverage performance of the WMSN directly. Mostly, the existing simulators and studies support 2D terrain representation, which is quite acceptable assumption for indoor applications [16, 17]. However, for outdoor applications, like border surveillance, a 2D terrain representation is not enough to simulate coverage and connectivity issues for the WMSN. If the terrain is represented in 3D, more realistic results can be produced for outdoor WMSN systems [18]. There are a few number of 3D based visibility analysis and coverage model studies available. In [19], Zhao *et al.* propose a novel 3D visibility model. However their approach is specially designed for visual tagging problem and it can not be directly used for coverage problems. The model proposed in [20] is also 3D based, but this model is only valid for convex regions. In [21], another 3D based visibility model is presented especially for the presence of random object occlusion. However, this model does not take into account of other occlusion types, like weather occlusions or natural occlusions. Becker *et al.* introduce a discrete 3D space modeling in [22]. In this model, it is assumed that the target space to be observed is the volume where a human being might navigate. It is not sufficient to model large scale outdoor applications for coverage issues.

In this paper, a new 3D coverage model and a new WMSN simulation environment based on the new coverage model is presented for connected coverage problems. The new simulation environment generates 3D environments synthetically, covers the terrain with different user defined objects, determines the weather conditions, deploys the multimedia sensors on the generated terrain, sets up the

characteristics of each sensor, and collects the performance results. It also has an optional optimization module that can locate the multimedia sensors according to the specified optimization parameters and constraints.

The remainder of this paper is organized as follows: In Section 2, the architecture of the simulation environment is introduced. In Section 3, the perception model is discussed. In Section 4, we present the coverage and the connectivity models. The sample scenarios which are designed to show the capabilities of the simulation environment and resulting simulation experiment performances are presented in Section 5. In Section 6, the conclusion is stated.

2. THE ARCHITECTURE OF THE SIMULATION ENVIRONMENT

There are four main modules of the simulation environment. Synthetic terrain generator, TerGen, generates the 3D landscape with specific weather conditions (snow, rain, fog), and natural details (lakes, rivers, forest, etc.). Scenario Editor, SenEd, sets the general system parameters of different sensor systems like video camera, night-vision, or mobile radar. All the data produced by TerGen and SenEd are given to the simulator engine, SimEn, and SimEn processes the scenario and produces the performance results. Optimization module, OptMod, is an optional component that tries to find the optimal location and utilization of the sensors for a given set of parameters. The user of the system can set the parameters and the constraints. The general system infrastructure of the simulation environment is illustrated in Figure 1.

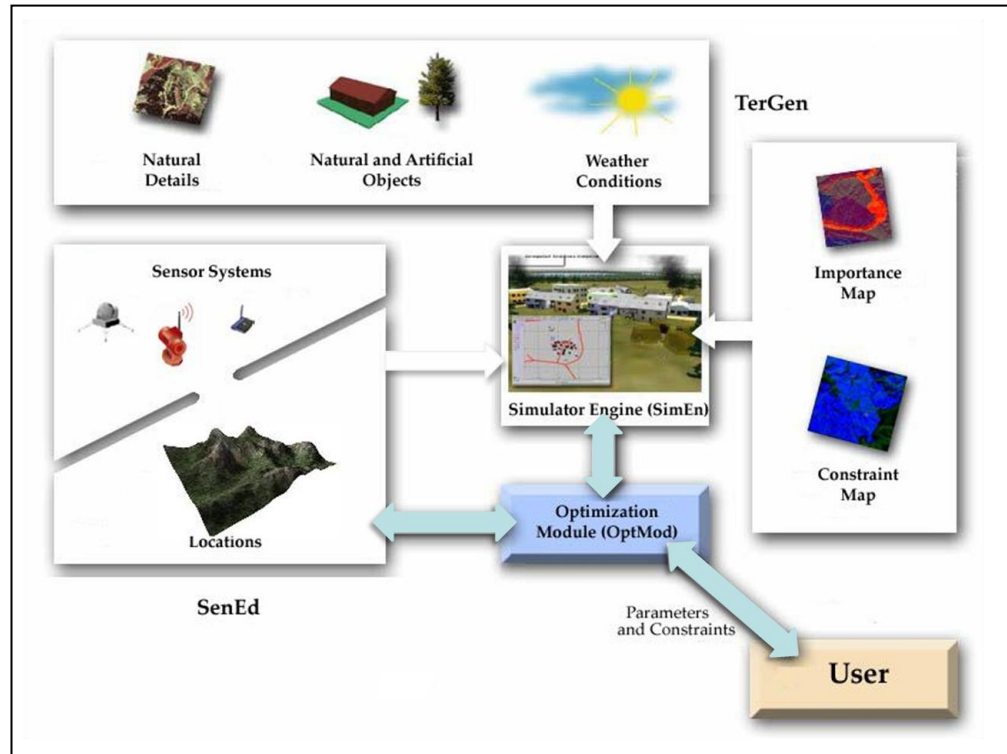


Figure 1. General system infrastructure of our simulation environment.

2.1. Terrain Generator Module (TerGen)

TerGen generates 3D synthetic terrains that can be covered with different objects and weather conditions. TerGen produces six different map data which are height map, polygon map, objects map,

weather condition map, importance map and constraint map. The height map defines the 3D structure of the terrain which is represented by a set of triangular polygons. These polygons are stored in the polygon map. TerGen can cover the terrain with natural or artificial objects that affect the visibility conditions directly. The object data are stored in the object map. Another parameter that affects the visibility is weather conditions. TerGen supports to create different weather conditions by the help of the weather condition map. The weather condition data can be set either for the whole terrain (global) or for a user-selected region of the terrain (local). The importance map is another map produced by TerGen. If there are some areas that have higher priority than the rest of the terrain, the user can provide this information in the importance map. The constraint map is used to restrict locating sensors on certain regions.

TerGen includes two different algorithms for height map generation, which are the Hill Algorithm [23] and the Diamond-Square Algorithm [24]. Hill Algorithm starts with a 2D flat terrain and considers random displacements. The generated terrain is normalized and the valleys are flattened out in order to provide a more realistic terrain. Diamond-Square Algorithm also starts with 2D flat terrain, and applies the diamond and the square steps repeatedly. An additional post processing phase takes place to smooth the unnatural looking spikes [25]. In Figure 2, two different sample terrains are presented.

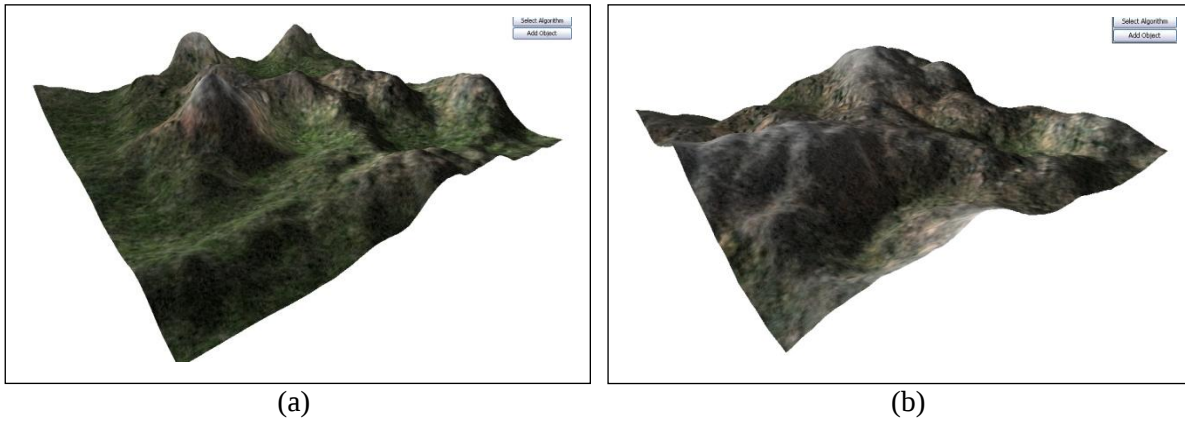


Figure 2. Sample terrains generated by (a) the Hill Algorithm, and (b) the Diamond-Square Algorithm.

Each generated terrain is represented by a set of triangular polygons. For each polygon, the following attributes are stored in the polygon map:

- *Polygon number*, which is a unique number for each polygon,
- *The coordinates of the polygon corners*,
- *Normal vector*,
- *The center of mass point*,

All the remaining maps generated by TerGen are based on the polygon map.

Object map stores the polygon number, object type, object density and object height data. Object map considers two basic object types; natural and artificial objects. Natural objects can be forest, bush, or lake, and artificial objects are buildings on the terrain.

Weather condition map, which also uses polygon structure, sets various weather conditions on the terrain, including rain, snow, fog, and temperature value. There are two different categories for the weather map. Global weather map holds weather information which is valid for all polygons of the terrain. Heat and humidity information is stored in the global weather map. Local weather information is stored in the local weather map. It includes the type and the density information for each polygon. Type information supports rain, fog and snow. Density data which are used in the visibility model of the simulation environment change between 0 and 1.

There may be certain regions that have more importance than the rest of the terrain. This can be thought as a common passage used by intruders. TerGen module provides an interface to set different importance levels and constraints for specific regions of the generated terrain. Importance map contains the importance value for each polygon. While default importance value is 0.5, the importance value can be set to 0.0, 0.25, 0.50, 0.75, or 1.0.

The last map generated by TerGen is the constraint map. A user may put constraints on certain areas on the terrain to restrict locating sensors on these regions. When a polygon is set as constrained, no sensor can be located on that specific polygon. This feature of TerGen can be used for rough geographic areas, or hostile regions. In Table I, a summary of maps generated by TerGen is presented.

TABLE I. SUMMARY OF MAPS GENERATED BY TERGEN.

Maps	Properties	
	Description	Stored Data
Height	3D terrain height	Height
Polygon	Polygons that compose 3D structure of the terrain	Polygon number Polygon corner coordinates Normal vector Center of mass point
Objects	Natural and artificial objects deployed on the terrain	Polygon number Object type Object density Object height
Global weather	Weather conditions for all the polygons of the terrain.	Heat Humidity Type (fog,rain,snow) Density (between 0 and 1)
Local weather	Weather conditions for specific regions.	Polygon number Type (fog,rain,snow) Density (between 0 and 1)
Importance	Importance level of the polygons.	Polygon number Value (0.0, 0.25, 0.50, 0.75, 1.0)
Constraint	Forbidden regions to put sensors.	List of constrained polygon number

2.2. Scenario Editor (SenEd)

The maps generated by TerGen are an input for simulator engine (SimEn). Another input for SimEn is scenario information produced by SenEd. The Scenario Editor Module (SenEd) is used to define the parameters of various sensor systems.

In this simulation environment, there are four different built-in sensor types which are CCD camera, night vision camera, thermal camera, and mobile radar. Basic attributes of a sensor are composed of behavioral and locational attributes. While behavioral attributes specify the visibility capabilities, locational attributes are related with the positional characteristics. The behavioral attributes of a sensor, s_i , are as follows:

- *Depth of view of the sensor*, Δ_i . It specifies the range of visibility for the given sensor.
- *Horizontal viewing angle*, α_i . The viewing angle indicates the wideness of viewing area of the sensor.
- *Vertical lower and upper viewing angles*, which are β_L and β_U , respectively. These angles bound the vertical viewing area of the sensor.
- *Communication range*, r_i . s_i can communicate with another sensor if it is in the communication range of s_i .

- *Financial cost of the sensor, $CostF_i$.* It is the financial cost of the sensor.

Depth, horizontal angle and vertical angle values are different for detection, recognition and identification modes. Locational attributes are as follows:

- *The location of the sensor.* It is stored as the polygon number, ps_i , on which the sensor is located. Based on our assumptions, each sensor can be located only on the center of mass of the polygon.
- *The heading angle, θ_i .*
- *The tilt angle, σ_i .* It is the angle between sensor and its carrier which is in the range of $(-90 + \beta_L) \leq \sigma_i \leq (90 - \beta_U)$, where β_L, β_U are the (vertical) lower and upper viewing angles of the sensor s_i .

All the sensor attributes are illustrated in Figure 3. In Figure 3, the sensor resides at point O and E_p is the XY projection of E.

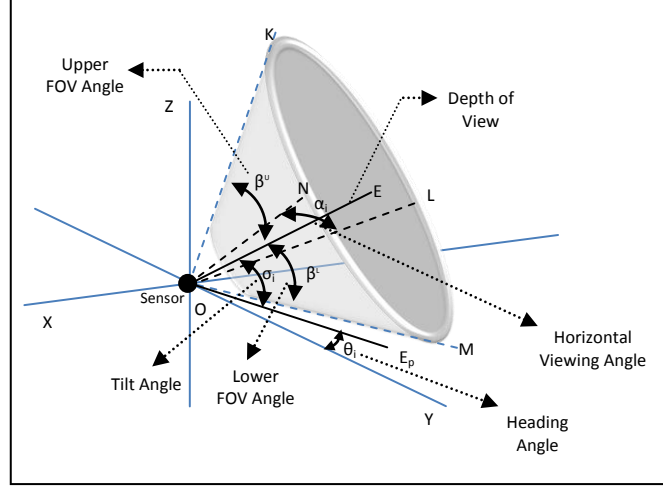


Figure 3. Behavioral and locational attributes of a sensor.

A multi-mode approach is considered for the characteristics of the sensors in order to provide more accurate and meaningful results for the perception process. Multi-mode sensor architecture is designed to characterize the differences among detection, recognition and identification concepts.

Detection is the active searching process to determine the existence of an object. In detection mode, only the prominent characteristics of the object can be determined. In recognition mode, sensor can perceive the object's real type and the some additional properties that cannot be determined in detection mode. For example, while detection mode can only give information that there is an object in a certain polygon, recognition mode can determine whether the object is a human or animal, soldier or civilian, natural or artificial construct. In identification mode, sensors can distinguish the friendly targets or hostile ones, and acquire the detailed information about the objects.

3. PERCEPTION MODELING FOR THE MULTIMEDIA SENSOR SYSTEMS

In this section, we present the perception procedure of a sensor s_i located on a polygon p_k . The perception calculations are performed in three phases:

1. Horizontal Field-of-View (FOV) test for all polygons.
2. Vertical Field-of-View (FOV) test on the polygons selected in Horizontal FOV test.
3. Applying Line-of-Sight (LOS) algorithm on the polygons selected in the Horizontal and Vertical FOV tests.

Horizontal and Vertical FOV tests are run for constructing the view-cone of the sensor, s_i . LOS algorithm is used to select only the polygons that are visible from s_i .

3.1. Horizontal FOV Test

Main purpose of the horizontal FOV test is to provide a proper filtering in XZ axis. In Figure 4, a sensor and its horizontal FOV test is presented.

The required inputs for horizontal FOV test are the location, the heading angle, the depth of view and the horizontal viewing angle of the sensor. The output of horizontal FOV is a set of polygons in the horizontal FOV area of the sensor.

Figure 4 represents XZ projection of the terrain that is visualized as polygons. In this figure, the sensor, s_i , is placed on point O; heading angle is $\theta_i = \angle ZOB$, and the horizontal viewing angle is $\alpha_i = \angle AOC$. Testing all the polygons is computationally intensive process. To determine the candidate polygons for FOV test, we first form a bounding rectangle that includes O, A, B and C points. This rectangle is shown as KLMN rectangle in Figure 4. Then, these points are extended to the nearest polygon points so that it will form a new rectangle (K'L'M'N') so that new rectangle will bound and include KLMN rectangle. Each polygon within the K'L'M'N' rectangle is tested for visibility within $\angle AOC$. In order to decrease the complexity of horizontal FOV test, we assumed that a polygon can pass FOV test if the average visibility of the three corner points and the center of mass point of the polygon is greater than zero. The polygons which are within the K'L'M'N' and visible from the sensor can pass horizontal FOV test.

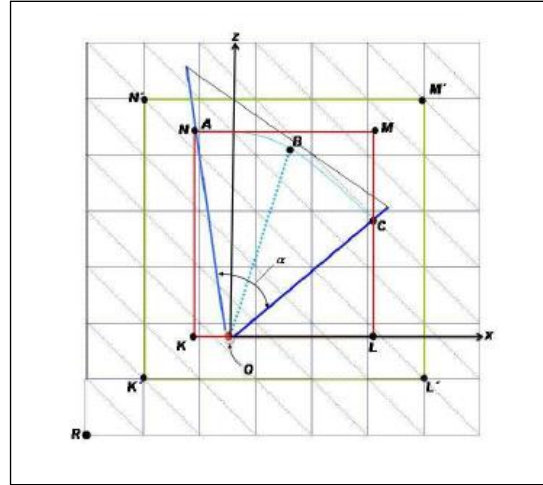


Figure 4. A Sample terrain for the horizontal FOV test.

3.2. Vertical FOV Test

The polygons which can pass from the horizontal FOV test is the input for the vertical FOV test. The upper viewing angle, the lower viewing angle and the tilt angle are the main attributes used in vertical FOV test.

In vertical FOV test, the first step is to calculate the sensor's lower slope, ρ_L , and upper slope, ρ_U , which can be calculated as in the following equation.

$$\rho_U = \tan(\sigma + \beta^U), \quad \rho_L = \tan(\sigma - \beta^L) \quad (1)$$

For each point to which the vertical FOV test is applied, the slope between the sensor and the selected point is calculated. If the slope satisfies the following inequality, then the point passes the vertical FOV test.

$$\min(\rho_U, \rho_L) < \tan \alpha < \max(\rho_U, \rho_L) \quad (2)$$

3.3. Line-of-Sight (LOS) Algorithms

In order to calculate LOS between two points, the height information of the ray and the terrain play a critical role. For each polygon between the points, the height of the terrain at that point is compared with the height of the ray for the same point. If the terrain is higher at the given point, it is concluded that the ray is blocked. If the ray is not blocked, there is LOS between the two points.

There are two different methods to determine the height value of each (x, z) point, which are the interpolation method and the nearest post method [26]. In this simulation environment, we implement the Janus Algorithm [27] as an interpolation-based algorithm and the Bresenham Algorithm [26] as a nearest-post based algorithm.

4. THE COVERAGE AND THE CONNECTIVITY MODELS

In this section, the mathematical formulation of coverage and connectivity of a WMSN is given by using the proposed perception modeling approach.

4.1. The Coverage Model

The perception model that includes horizontal *FOV*, vertical *FOV* and *LOS* tests, determines the polygon set in the scope of the sensor. It does not consider the weather condition or object occlusions. Our coverage model uses perception model to calculate the coverage of the WMSN terrain.

The coverage of a set of sensors, S , on a set of polygons P , $Cov(S, P)$, is formulated in Equation 3,

$$Cov(S, P) = \frac{\sum_{p_i \in P} V(S, p_i) \times w_{p_i}}{\sum_{p_i \in P} w_{p_i}} \quad (3)$$

where w_{p_i} is the weight of the polygon p_i , and $V(S, p_i)$ which is computed by the Equation 4, is the visibility value of polygon p_i ,

$$V(S, p_i) = Avg(\max_{s_j \in S} (V_{s_j}(s_j, p_k, b))), \text{ where } b \in p_i \quad (4)$$

where, V_{s_j} is the visibility of the point b , from the sensor s_j where resides at the center of mass point of p_k . A certain point can be perceived by multiple sensors. In this case, the maximum visibility is selected as the visibility of the point. $V(s_j, p_i)$ is simply the average of the maximum visibility values of the points in p_i . Calculating all the visibility values for all points in a polygon is computationally expensive. In this simulation environment, three corners and the center of mass point of the polygons are selected as representative points for visibility calculations.

$V_{s_j}(s_j, p_k, b)$, which is the visibility of the point b from the sensor s_j where resides in the center of mass point of p_k , is formally defined in Equation 5.

$$V_{s_j}(s_j, p_k, b) = \left(1 - \eta_{s_j} \times \frac{d(a, b)}{\Delta_{s_j}}\right) \times \left(1 - \max_{p_c \in a \rightarrow b} \mu_c^w\right) \times \left(1 - \max_{p_c \in a \rightarrow b} \mu_c^o\right) \quad (5)$$

The first term represents the decreasing visibility sourced from the distance. η_{s_j} is the range effect coefficient of sensor s_j ; a is the coordinate of the sensor; $d(a, b)$ is the Euclidean distance between the sensor and the target point b ; Δ_{s_j} is the depth of view of s_j . The visibility of the sensors decreases, when the distance between the target point and the sensor increases. The second term, the weather permeability, is the effect of the weather conditions. In this term, μ_c^w represents the weather density of the polygon p_c .

Each polygon has a weather density value in the weather map that represents snow, fog or rain. If weather density is higher, the visibility is lower. The weather permeability is defined as the maximum of the weather densities between the sensor and the target point. The last term is the object permeability, which is the object version of weather permeability. In 3D terrain representation, a polygon can be covered with objects. Different objects may have different transparency levels. This information is stored as object density. μ_c^o represents the object density of the polygon pc. The object permeability is the maximum of the object densities between the sensor and the target point. If there is an object with higher object density between the sensor and the point, object permeability gets lower.

4.2. The Connectivity Model

Connectivity is an important constraint for our problem. A network is connected if a node can communicate with all the other nodes. In order to test the network against connectivity, a connection assumption is needed. In most of the studies, it is assumed that if two sensors are in a certain communication range, they can communicate with each other. On the other hand, due to weather conditions or occlusions, they may not be able to communicate even if they are located in the range.

In this simulation environment, it is assumed that two sensor nodes s_i and s_j are connected, if the visibility between the sensors is greater than a predefined communication threshold (τ), and the distance between them is less than the communication range of the sensors, r_i , and r_j . Formally, a connection between two sensor nodes s_i and s_j are established based on the following equation:

$$con(s_i, s_j) = \begin{cases} 1, & \text{if } V_{s_j}(s_i, p_k, b) \geq \tau \text{ and } d(s_i, s_j) \leq r_i, r_j \\ 0, & \text{o/w} \end{cases} \quad (6)$$

$\forall (s_i, s_j) \text{ where } s_i \text{ and } s_j \in S$

where b is the coordinate of s_i , and $d(s_i, s_j)$ is the Euclidean distance between the sensors.

5. ILLUSTRATIVE SCENARIOS AND PERFORMANCE RESULTS

In this section, a number of scenarios with different parameters are generated and the performance results of the experiments using these scenarios are given in order to show the capabilities of the simulation environment.

5.1. Sample Scenarios

The simulation environment's Terrain Generator, TerGen, can generate terrains with different toughness levels (smooth, rough). It also can cover the terrain with weather and object occlusions. The sample scenarios are summarized in Table II. The scenarios are designed to show the performance difference sourced from the toughness of the terrain and the occlusions.

TABLE II. TERRAIN STRUCTURES OF THE SAMPLE SCENARIOS.

<i>Toughness</i>	<i>Occlusion Type</i>
Smooth	No occlusion
	Object occlusion
	Weather occlusion
	Weather and object occlusion
Rough	No occlusion
	Object occlusion
	Weather occlusion
	Weather and object occlusion

There are four different groups of scenarios about occlusions: no occlusions, object occlusions, weather occlusions, weather and object occlusions. There are two toughness levels in the scenarios: smooth and rough. The simplest scenario is the smooth terrain with no occlusion. The most complex one is the rough terrain with weather and object occlusions. While a smooth terrain sample with no occlusion and with weather and object occlusions are presented in Figure 5, a rough terrain sample with no occlusion and with weather and object occlusion is presented in Figure 6.

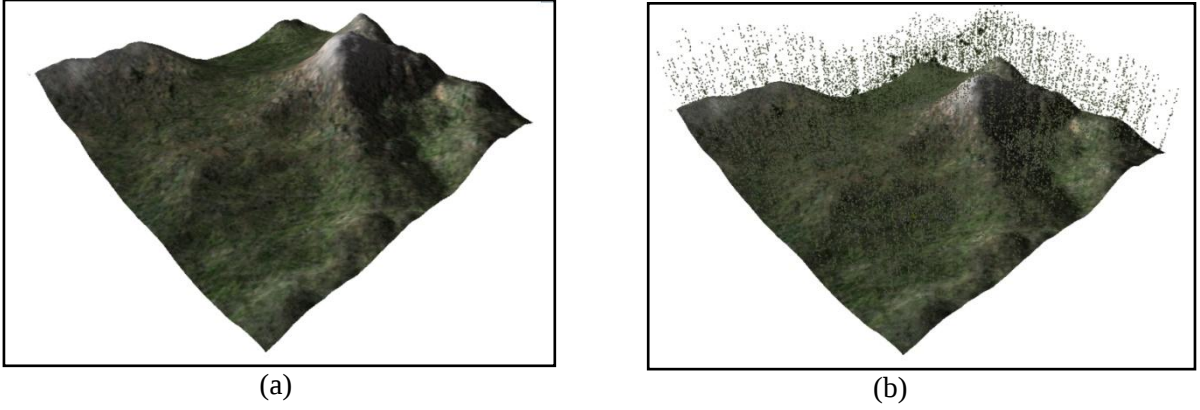


Figure 5. Smooth terrain with (a) no occlusion, (b) global weather and object occlusion.

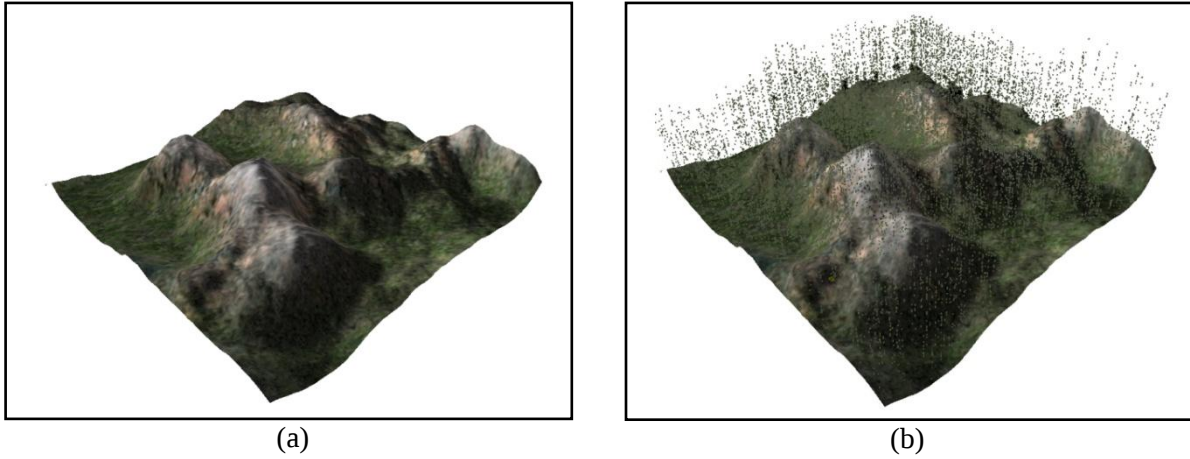


Figure 6. Rough terrain with (a) no occlusion, (b) global weather and object occlusion.

Scenario Editor (SenEd) is used to set sensor parameters. SenEd can define different types of multimedia sensors, like camera, microphone, and radar. In this paper, we use CCD camera based wireless multimedia sensor. The parameters of the sensor are presented in Table III. In the scenarios, only detection mode is considered.

In order to show the whole capabilities of the simulation environment, we also use Optimization Module (OptMod) in the scenarios. While the simulation environment can only simulate a specific WMSN deployment scenario, it can also be used as an optimization tool to locate and utilize the sensors.

OptMod support different optimization techniques. The user can implement different techniques to locate and to utilize the sensors for maximum coverage and connectivity. Optimization problem can be expressed in terms of single objective or multiobjective function. In this paper, we define our problem as a multiobjective function of cost and coverage, and we try to find optimal solution with Genetic Algorithm (GA).

TABLE III. BEHAVIORAL PARAMETERS OF THE SENSOR USED IN THE SCENARIOS.

<i>Parameters</i>	<i>Mode</i>	<i>Value</i>
Depth of view	Detection	600 m
	Recognition	300 m
	Identification	150 m
Horizontal viewing angle	Detection	160°
	Recognition	120°
	Identification	80°
Vertical lower and upper viewing angles	-	50°
		50°
Communication range	-	300 m

In this paper, our optimization problem is to maximize the coverage area and to minimize total acquisition cost of the sensors, while satisfying the connectivity constraints. Let a terrain set P , and a sensor set S are given. The problem is formulated as follows:

$$\begin{aligned}
& \max \quad z_1 = Cov(S, P) \\
& \min \quad z_2 = \sum_{i \in S} CostF_i / E(S) \\
& \text{subject to} \\
& \quad s_i \text{ in } G(S, E), \quad \forall s_i \in S
\end{aligned} \tag{7}$$

Here, $CostF_i$ is the financial cost of sensor i , and $E(S)$ is the expected value of the number of sensors for the given terrain. $G(S, E)$ is a connected graph where vertices of G are the sensor node set S and edges of G are the connections of the WMSN. The edges of G are set by using Equation 6.

Since the problem described in Equation 7 is a multiobjective optimization one, a unified objective function is needed. The complementary and conflicting objectives, cost and coverage, are represented by a multiplicative total utility function model based on the multi-attribute utility theory. The total utility function is presented in Equation 8.

$$U_{tot} = w_{cov} U_{cov} + w_{cost} U_{cost} + w_{cov} w_{cost} U_{cov} U_{cost} \tag{8}$$

where U_{cov} is the utility of the coverage of the terrain, and U_{cost} is the utility of the total cost of the sensors. In this equation, w_{cov} and w_{cost} are the weights of the coverage, and the cost utility functions respectively, where $w_{cov} + w_{cost} = 1$. These weights are set based on experimentation on a given terrain by considering various military scouting missions.

The total utility function is accepted as the fitness function of GA. The details of the GA techniques, which are not the main contribution of this paper, can be found in [28].

5.2. Performance Results

Firstly, the experiments are performed in order to find the most effective parameters of our GA method in OptMod, such as crossover type and mutation type. The second category of experiments, which focuses on the effects of various terrains with different toughness levels, occlusions and weather conditions, are conducted to show the capabilities of the simulation environment. In this paper, only the second category experiment results are presented. While our WMSN simulation environment is coded in Java programming language, the GA approach of our framework (OptMod) is coded in C programming

language. The computational experiments are conducted on a cluster of machines, each of which has an Intel Xeon 2.33 Ghz. processor running the Linux operating system.

In order to show the capabilities of the simulation environment, the effects of various factors on the quality of solutions are measured using the scenarios generated before and related discussions are provided. The resulting coverage maps are presented in Figure 7-10.

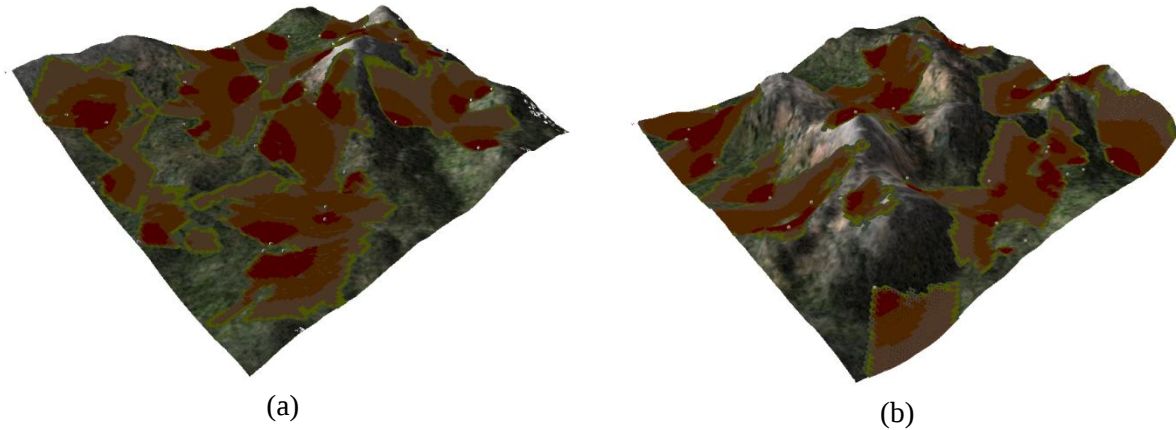


Figure 7. Coverage map of the (a) smooth, (b) rough terrain without object and weather occlusions.

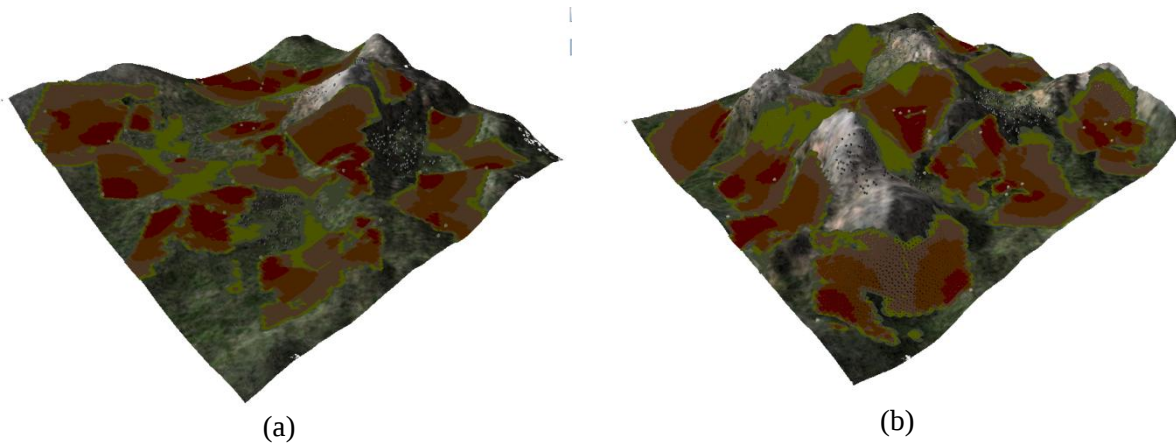


Figure 8. Coverage map of the the (a) smooth, (b) rough terrain with object occlusion.

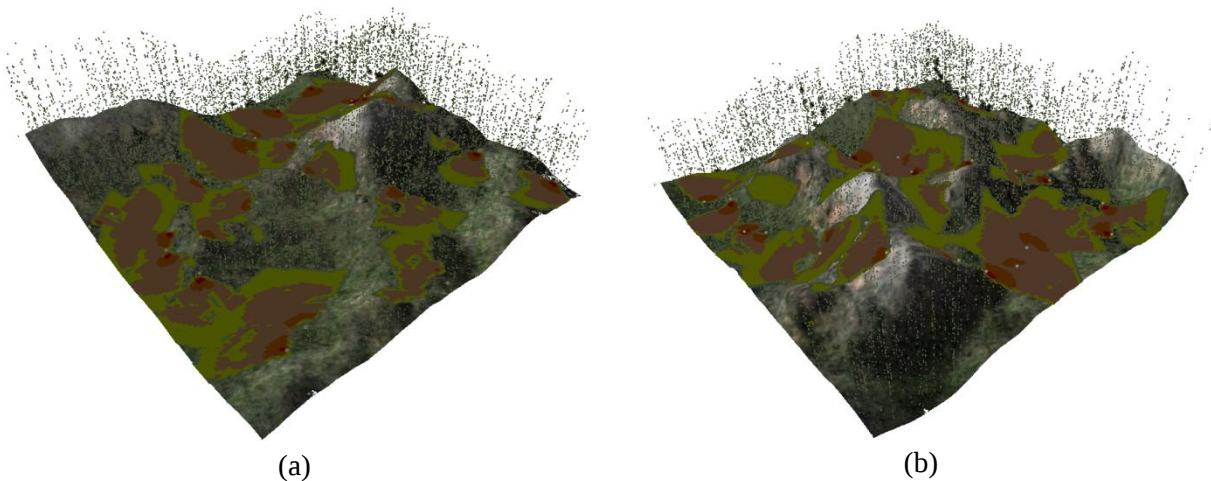


Figure 9. Coverage map of the (a) smooth, (b) rough terrain with weather occlusion.

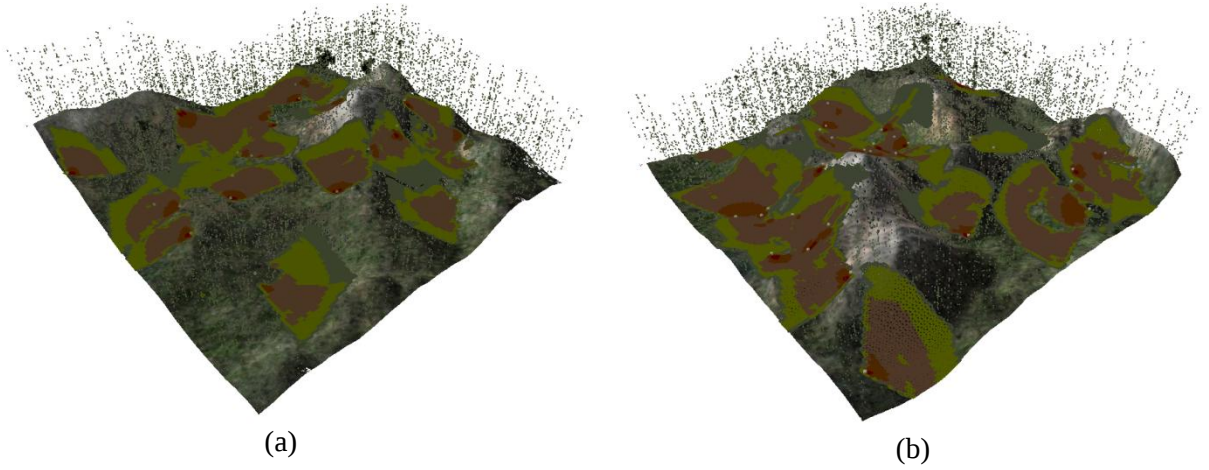


Figure 10. Coverage map of the (a) smooth, (b) rough terrain with object and weather occlusions.

Since the scenarios include different terrain structures and different occlusion types, the resulting coverage area are highly affected and are different in each case. It should be noted that each terrain is represented by a set of fixed number of triangular polygons.

Performance comparisons of the scenarios with respect to fitness function presented in Equation 8 are presented in Table IV.

TABLE IV. PERFORMANCE COMPARISON OF THE SCENARIOS.

<i>Scenario</i>	<i>Fitness Value</i>	<i>Connected Solution (%)</i>
Smooth – no occlusion	0.5838	96
Smooth – object occlusion	0.5559	96
Smooth-weather occlusion	0.5245	98
Smooth-weather & obj. occlusion	0.5143	100
Rough-no occlusion	0.5552	94
Rough-object occlusion	0.5515	80
Rough-weather occlusion	0.5010	82
Rough-weather & obj. occlusion	0.4911	72

In a nutshell, the 3D terrain structure and the occlusion level strongly affect the resulting coverage area. The best solution is obtained with smooth terrain with no occlusion scenario. The occlusions degrade the performance of the system. Another factor that affects the performance is terrain toughness. When the terrain toughness is rough, the GA can find the worse solutions than the smooth ones. The lowest fitness function is obtained for the rough terrain with object and weather occlusions. Besides, the percentages of connected solutions are very close to 100. That is, in the worst case, approximately 40 individuals out of 50 ones in population have a connected solution. This also points out the effectiveness and robustness of our GA-based solution approach under various values of several experimental parameters.

6. CONCLUSION

Wireless multimedia sensor networks (WMSN) are a new family of Wireless Sensor Networks (WSN) that allow retrieving video and audio streams, still images, as well as scalar sensor data. Especially, for military missions, like border surveillance or gathering intelligence, WMSN has many promising

application areas. Connected coverage concept is one of the most important design issues for WMSN. WMSN nodes should be deployed so that while the covered area is maximized, nodes must be able to form a connected network. Although there are some researches about WMSN connected coverage problems on 2D terrain which is an acceptable assumption for indoor applications, there are very limited studies about connected coverage problems on 3D terrains. In this paper, a new WMSN simulation environment for connected coverage on 3D terrain is presented.

The new simulation environment is composed of four main modules, TerGen, SenEd, SimEn, OptMod. Terrain Generator, TerGen, generates six different maps to represent a synthetic 3D terrain. Height and polygon map are generated for representing the 3D structure of the terrain. TerGen can cover the terrain with natural or artificial objects with its object map. The weather map can set different weather conditions for a certain area of terrain or for all the terrain. By the help of the importance map, it is possible to mark polygons for different levels of importance. The constraint map can restrict the area that can hold the sensors. SenEd, which is another module of the new simulation environment, creates basic behaviors, locations and parameters of the sensors. SenEd can hold different sensor parameters for detection, recognition, and identification mission modes. The outputs of TerGen and SenEd are the inputs of Simulator Engine, SimEn. SimEn takes the synthetically generated 3D terrain, sensor parameters and calculates the coverage and connectivity performance results.

While this simulation environment can be used just for simulating a proposed system, it can also be used for optimization purposes. Optimization Module, OptMod, can be customized for different optimization techniques to optimize WMSN systems for certain objectives and constraints. For example, OptMod can locate the sensors to maximize coverage while maintaining connectivity.

In order to show the capabilities of the simulator, a series of scenarios with different terrain complexities and different occlusions are generated. In order to try to find optimal sensor deployment and utilization, a GA-based solution approach is implemented in OptMod. The simulation and optimization results show that the terrain structure and the occlusions highly affect the connected coverage performance of the WMSN.

ACKNOWLEDGMENT

This research was supported by The Scientific and Technological Research Council of Turkey (TUBITAK) with a research grant (no. 106E159). Additionally, parts of the computations have been carried out by using UYBHM at ITU through a grant (20432008).

REFERENCES

1. Akyildiz I.F. and Su W., Sankarasubramaniam Y. and Cayirci E., "Wireless Sensor Networks: A Survey", *Computer Networks*, vol. 38 (4), pp. 393–422, April 2002.
2. Yick J., Mukherjee B., and Ghosal D., "Wireless sensor network survey," *Computer Networks*, vol.52, no.12, pp. 2292-2330, August 2008.
3. Akyildiz I.F., Melodia T., and Chowdury K.R., "Wireless multimedia sensor networks: A survey," *Wireless Communications, IEEE*, vol.14, no.6, pp. 32-39, 2007.
4. Huang C.F. and Tseng Y.C., "A Survey of Solutions to the Coverage Problems in Wireless Sensor Networks", *Journal of Internet Technology*, Vol. 6, No. 1, pp. 1-8, 2005.
5. OPNET Inc., <http://www.opnet.com/>, last visited on 06/25/09.
6. Network Simulator - NS-2, <http://www.isi.edu/nsnam/ns/> last visited on 06/25/09.

7. Zeng X., Bagrodia R., Gerla M., "GloMoSim: a Library for Parallel Simulation of Large-scale Wireless Networks", Proceedings of the 12th Workshop on Parallel and Distributed Simulations -- PADS '98, May 26-29, 1998 in Banff, Alberta, Canada.
8. Levis, P., Lee, N., Welsh, M., and Culler, D., "Tossim: accurate and scalable simulation of entire tinyos applications", Proceedings of the 1st international conference on Embedded networked sensor systems, New York, NY, USA: ACM Press, 2003, pp. 126-137.
9. Girod L., Elson J., Cerpa A., T. Stathopoulos, N. Ramanathan, and D. Estrin, "Emstar: a software environment for developing and deploying wireless sensor networks," Proceedings of the annual conference on USENIX Annual Technical Conference, Berkeley, CA, USA: USENIX Association, 2004, p. 24.
10. Sameer S., Kim W., and Agha G., "SENS: A Sensor, Environment and Network Simulator," Proceedings of 37th Annual Simulation Symposium, Arlington, VA, 2004.
11. Mekni M., Moulin B., "A Survey on Sensor Webs Simulation Tools," Proceedings of the Second International Conference on Sensor Technologies and Applications, pp.574-579, Cap Esterel, France, 2008.
12. Singh C.P., Vyas O. P., Tiwari M.K., "A Survey of Simulation in Sensor Networks," International Conferences on Computational Intelligence for Modelling, Control and Automation, December 10-12, 2008, pp.867-872.
13. Korkalainen M., Sallinen M., Kärkkäinen N., Tukeya P., "Survey of Wireless Sensor Networks Simulation Tools for Demanding Applications," ICNS, Fifth International Conference on Networking and Services, April 20-25, 2009, pp.102-106.
14. O'Rourke J., Art Gallery Theorems and Algorithms. Oxford University Press, 1987.
15. Han X. C., Lloyd E.L., Shen C.C., "Deploying Directional Sensor Networks with Guaranteed Connectivity and Coverage", Proceedings of IEEE SECON 2008, San Francisco, CA, June 16-20, 2008.
16. Horster H., Leinhart R., "Optimal Placement of Multiple Visual Sensors", In "Multi-Camera Networks - Principles and Applications", Aghajan H., Cavallaro A., ACADEMIC PRESS, 2009, pp: 117-138.
17. Tezcan, N. and Wang, W., "Self-orienting wireless multimedia sensor networks for occlusion-free viewpoints", Computer Networks, vol. 52, no. 13, Sep. 2008, pp. 2558-2567.
18. Ma H., Zhang X., Ming A., "A Coverage-Enhancing Method for 3D Directional Sensor Networks", INFOCOM 2009: The 28th Conference on Computer Communications, April 19-25, 2009.
19. Zhao J.; Cheung S.C.; Nguyen T., "Optimal Camera Network Configurations for Visual Tagging," IEEE Journal of Selected Topics in Signal Processing, vol.2, no.4, pp.464-479, Aug. 2008.
20. Ram, S., Ramakrishnan, K. R., Atrey, P. K., Singh, V. K., and Kankanhalli, M. S., "A design methodology for selection and placement of sensors in multimedia surveillance systems", In Proceedings of the 4th ACM international Workshop on Video Surveillance and Sensor Networks, October 27, 2006, pp. 121-130.
21. Mittal, A. and Davis, L. S., "A General Method for Sensor Planning in Multi-Sensor Systems: Extension to Random Occlusion", International Journal of Computer Vision, vol. 76, no. 1, Jan. 2008, pp. 31-52.

22. Becker, E., Guerra-Filho, G., and Makedon, F., "Automatic sensor placement in a 3D volume", In Proceedings of the 2nd international Conference on Pervasive Technologies Related To Assistive Environments, Corfu, Greece, June 09 - 13, 2009, pp. 1-8.
23. Nystrom B., Terrain Generation Tutorial: Hill Algorithm, 2000.
24. Fournier A., Fussel D., and Carpenter L., "Computer Rendering of Stochastic Models", Communications of the ACM, vol. 25, no. 6, pages 371-384, 1982.
25. Grumet M., Terrain Modelling, Technical Report, Institute of Computer Graphics and Algorithms, Vienna University of Technology, 2004.
26. Proctor M.D., and Gerber W.J., "Line-of-sight Attributes for a Generalized Application Program Interface", The Journal of Defense Modeling and Simulation, Vol. 1, No. 1, 43-57, 2004.
27. Handerson D., "ModTerrain: A Proposed Standart for Terrain Representation in Entity Level Simulation", MS Thesis, Naval Postgraduate School, California, 1999.
28. Topçuoğlu H., Ermiş M., Sifyan M., "Positioning and Utilizing Sensors on a 3D Terrain: Part II - Solving with a Hybrid Evolutionary Algorithm", submitted to IEEE Transactions on Systems Man and Cybernetics – Part C: Applications.

Path Planning for Mobile Sensor Platforms on a 3-D Terrain Using Hybrid Evolutionary Algorithms

Mesut Sifyan, Haluk Rahmi Topcuoglu, and Murat Ermis

Abstract—In this paper, a novel hybrid method for path planning problem of multiple mobile sensors on a 3-D terrain is proposed. Our method proceeds in two phases: the global path-planning phase, and the local path planning phase. The first phase constructs a connectivity graph generated by a probabilistic roadmap (PRM) method and selects the control points of sensors' paths from the set of nodes generated by the PRM method. In the local path-planning phase, a hybrid evolutionary algorithm is proposed to determine the intermediate points, which are between control points of sensors' paths in order to complete the paths. The local-path planner considers the accessibility of control points, smoothness of each path, visibility of terrain covered by mobile sensors and the total cost of all paths (i.e. the total length of all paths). The experimental study points out the effectiveness of our framework under various terrain and sensor characteristics.

I. INTRODUCTION

Path planning problems exist in various fields such as robotics, computer aided design, manufacturing, installation systems, computer animations, and aviation applications [1]. Path planning algorithms aim to solve problems by computing a set of continuous configurations to reach goal positions from initial positions by avoiding collisions. Path planning process is a computationally intensive process and it has exponentially growing complexity with respect to the dimension of configuration space. For this reason, finding exact solution in three dimensional environments is an NP-hard problem [2]. Complexity of path planning problem also depends on the sensors (robots or mobile platforms) to be involved in path planning together with the degree of freedom of the environment. In multiple paths planning, the communication between mobile sensors and collision avoidance in dynamic environment may cause overhead when compared with single path planning.

Several methods have been proposed in the literature for global path planning including path planning with Voronoi diagrams [3], cell decomposition [4], and sampling based path planning [5], [6], [7], [8]. Since gathering global information about the environment is a complex and time consuming process, sampling based techniques like probabilistic roadmap (PRM) [6] or rapidly exploring random tree (RRT) [7] can be used to generate paths efficiently. Despite

of the fact that the paths generated by using probabilistic roadmap are not optimum solutions, it is one of the fastest algorithms for multiple paths planning problem and it gives configuration space that does not include any obstacle.

Coverage path planning is a specific type of path planning problem. There are many robotic applications related with coverage path planning like cleaning room [9], land mine detection [10], autonomous underwater covering [11], and agricultural crop harvesting [12]. In most of these applications, coverage problem generally aims to reach all possible points on the environment. There are several other studies on coverage path planning that considers artificial potential field [13], and neural networks [14]. While some of these studies focus on single robot systems, some of them are related with multi-robot systems. Although there are various studies on path planning including coverage, most of the algorithms were designed to consider only two dimensional environments [15], [16].

In this study, a novel hybrid EA-based framework for path planning of multiple mobile sensor platforms on a 3-D environment is presented by considering multiple objectives. The visibility of the terrain covered by mobile sensors, the smoothness of paths, total costs of all paths, and accessibility of paths are the objectives used in our study. Additionally, our model considers collision detection of static obstacles on the terrain and collision avoidance with other sensors considered in the problem.

The remainder of the paper is organized as follows: In Section 2, we present the global path planning phase of our novel framework. Section 3 gives the details of our hybrid EA-based solution approach for the local path planning phase. Performance evaluation and experimental study is discussed in Section 4, and Section 5 concludes the paper.

II. GLOBAL PATH PLANNING FOR DETERMINING CONTROL POINTS

Our solution framework is performed in two phases: a) *the global path planning phase* which generates control points of sensors' paths by considering only visibility and distance criteria in order to reach all accessible regions in a given terrain, and b) *the local path planning phase* which fills intermediate points between control points in order to complete the paths based on accessibility, smoothness, visibility and cost criteria.

The global path planning phase requires three consecutive steps in order to determine the control points of sensors. Firstly, nodes of a connectivity graph are generated by using the probabilistic roadmap (PRM) technique. Then, a set of

Mesut Sifyan is with the Computer Engineering Department, Marmara University, Goztepe Kampusu, Kuyubasi, Istanbul, 34722, Turkey (email: mesutsifyan@marun.edu.tr).

Haluk Rahmi Topcuoglu is with the Computer Engineering Department, Marmara University, Goztepe Kampusu, Kuyubasi, Istanbul, 34722, Turkey (email: haluk@marmara.edu.tr).

Murat Ermis is with the Industrial Engineering Department, Turkish Air Force Academy, Yesilyurt, Istanbul, 34149, Turkey (email:m.ermis@hho.edu.tr).

sample nodes from the graph are selected based on their locations and distances among them. Finally, assignments of control points to the paths of mobile sensors are performed with a function called VPD (visibility-per-distance) function. In this section we present the details of those three steps of the global path planning phase.

A. Probabilistic Roadmap Generation

In a multiple path planning problem, multiple initial-goal queries increase the complexity of the path planning algorithm. There are a set of sampling based path planning techniques that spend substantial time for preprocessing models for future queries [1]. Probabilistic roadmap (PRM) is a sampling based technique which construct a topological graph called *roadmap* that efficiently solves multiple queries [6]. The paths of sensors can be easily generated on the roadmap to reach from each of sensors' initial positions to their target positions without any collision. PRM method is used for the global path planning, which requires two phases of computation [6]:

- *Preprocessing Phase.* In our framework, the number of candidate nodes to be considered in PRM is set and the terrain is separated into square regions whose count is equal to the number of candidate nodes. Each candidate node in the corresponding region is located on the center of the region with a probability of 20%, and on any polygon from the region with a probability of 80%. Then, a subset of candidate nodes in the neighborhood of each node are selected in order to provide a connectivity graph. This operation is performed by the radius method [6] where all nodes within a circle of radius r centered at the location of a given node i are selected as the neighborhood set of the node i (see Figure 1). Then, each nodes in the neighborhood set is tested whether it is accessible from the node i or not based on sensor and terrain characteristics. Specifically, accessibility from the node i to the node j means that the slope of each polygon on the route between node i and node j is less than the maximum slope of climbing for the given sensor. It also considers whether any obstacles of the terrain are on the route or not as well. Each accessible line segment between any two nodes is represented as an edge in the connectivity graph. The nodes which cannot access to any neighbor are not included in the graph. Therefore, the generated graph represents the accessible regions on the terrain.
- *Querying Phase.* The path of each sensor platform is generated by considering the connectivity graph based on the initial-goal pair of the sensor. The nodes and edges are selected from the graph by using the Dijkstra's shortest path algorithm. The generated paths are free from any collision with static obstacles and steep regions to climb.

Number of candidate nodes, neighborhood distance of the nodes and maximum slope of climbing are the three critical parameters that can affect performance of the generated

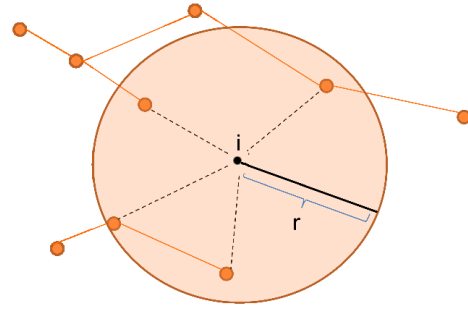


Fig. 1. Selection of nodes in the neighborhood of a node by radius method

probabilistic roadmap. The number of candidate nodes affects adaptability of the generated PRM graph to the terrain. In rough terrains, nodes which are deployed densely affects the size of search space directly. On the other hand, only smooth parts of the terrain can be represented in search space if nodes are deployed sparsely.

Since radius method is used for neighbor selection in our framework, the edge count of a node directly depends to the length of radius (i.e., the neighborhood distance). If the radius of the circle is very small, the number of neighbor nodes would be very small and the possibility of representing a node in the graph would decrease accordingly. However, if the radius of the circle is very large, the computation time of querying phase would become exhaustive due to increase in the number of edges of the node.

Another parameter of the mobile sensor platforms is the maximum slope of climbing. If the maximum slope value increases, the size of the search space will increase due to the increase in the number of represented node in the graph.

B. Determining the Control Points

This step determines control points from the nodes of the connectivity graph by considering all accessible regions of the terrain. If there are steep regions like hill or valley, sensor platforms cannot climb to these regions but control points can be selected around these regions in order to increase the visibility of the whole terrain by surveilling inaccessible regions with moving around it (see Figure 2). This step is conducted in two substeps:

- Firstly, node selection is performed by considering the number of neighbors connected to the node. Starting from the least number of edges to the most, a node is selected as a control point if there is a predefined distance from the node to the previously selected control points. Nodes which have few edges are generally located around steep regions like valley and hill. In order to prevent selecting a node located near the boundaries of the terrain, the node are replaced with one of its neighbors that is not located in the boundary region. The following substep may increase the quantity of control points.
- If there are accessible regions that is not represented

with a control point, additional control points are selected from the nodes of the connectivity graph in order to get smoother paths.

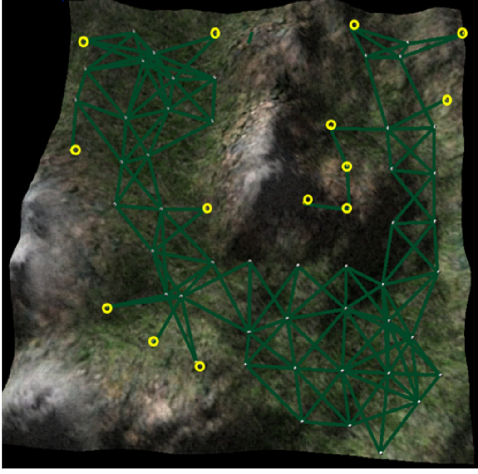


Fig. 2. Selection of control points on the terrain

C. Assigning the Control Points to Sensor Paths

As the final step in global path planning, selected control points are assigned to the paths of sensor platforms by using a function, called VPD (visibility-per-distance) function, which is adapted from the term M_{PDV} given in [17]. VPD function considers change rate on both the visibility of given terrain and the path length whereas M_{PDV} considers visibility of a point and distance traveled to visit the point. By VPD function, the contribution of a control point for each sensor path is calculated and the control point is assigned to the path that gives the highest contribution value. If a control point is very close to a path, it is assigned to that path without any calculation. The formal definition of VPD is given below:

Let S be the set of all sensors in the problem and $s_i \in S$ be a sensor platform in the problem. Let k_q be a control point generated in previous steps of local path planning phase, and let K^a be the set of control points that are already assigned to a path. The term $V(s_i, K_i)$ represents the visibility of the sensor s_i on its path that consists of set of control points K_i . The VPD function is computed by Equation 1, where $\Delta V(S, k_q)$ represents the ratio of the current visibility value, $V(S, K^a)$, and the new visibility value after adding the control point k_q in the path, $V(S, K^a + k_q)$:

$$VPD = \frac{\Delta V(S, k_q)^\alpha}{\Delta d(s_i, k_q)^{5-\alpha}} \quad (1)$$

In this equation, $\Delta d(s_i, k_q)$ represents the ratio of current path length without considering new control point k_q and the new path length after adding the control point k_q to the path of sensor s_i . Additionally, the term α represents the weight of the visibility in VPD function, where $0 \leq \alpha \leq 5$. If $\alpha = 5$, adding control point considers the visibility parameter, and if $\alpha = 0$ it considers the distance parameter.

III. LOCAL PATH PLANNING PHASE FOR DETERMINING INTERMEDIATE POINTS

For the local path planning phase, a hybrid steady-state evolutionary algorithm (HEA) is proposed. It generates a single individual at each iteration after the crossover and mutation operators, which is replaced with the worst individual in population. Parent selection is performed by tournament selection mechanism with a tournament size of 5. In order to provide hybridization, both problem-specific methods included in initial population and problem specific variation operators are considered in our study. This section presents the details of our hybrid evolutionary algorithm based local path planner.

A. String Representation

Initially, the string representation of each path includes coordinates of control points of the path in the order from initial to goal location, and the string representations of sensors are concatenated in order to provide string representation of the solution. The string representation of each sensor is extended with variable number of intermediate points added to each sub-path, where a sub-path is a path between two control points that are neighbors based on their positions on the string (see Figure 3).

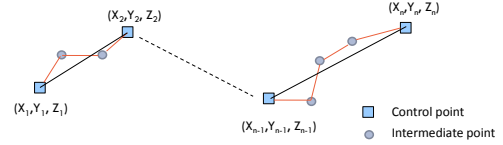


Fig. 3. The representation of a sensor's path

B. Fitness Function Evaluation

The fitness function evaluation of a solution is computed by considering the visibility, cost, smoothness and accessibility criteria. Specifically, fitness value of a solution R , which is the combination of paths of sensors considered, is computed by Equation 2:

$$f(R) = V(A, S, P)^{w_v} (w_d D(R) + w_s S(R))^{1-w_v} - w_c C(R) \quad (2)$$

where w_v, w_d, w_s, w_c are the weights of visibility ($V(A, S, P)$), cost ($D(R)$), smoothness ($S(R)$) and accessibility ($C(R)$) components, respectively. Here, $0 \leq w_v, w_d, w_s \leq 1$ and $w_c > 1$. Since, the main emphasize in this study is the coverage of the terrain, the visibility and the other terms including cost and smoothness terms are separated in the equation, and the weight of visibility (w_v) is used to determine the effect of these terms.

1) *Visibility Component*: The calculation of visibility for mobile sensor platforms is based on the visibility calculation for static sensors [18]. As in the case of static sensors, the perception model includes generation of polygons which are located on the covered region and the calculation of the visibility values for those polygons. Since there is a finite number of points on a path, the path can be considered

as a set of line segments. The area perceived by a sensor is localized by using horizontal field-of-view (FOV) and vertical FOV tests, where these tests are applied on each line segment in the path.

- *Horizontal FOV test.* The purpose of horizontal FOV test is to filter points on the terrain by considering XZ axis. The inputs required to perform this method are heading angle, horizontal viewing angle, depth of view, and the initial and target points of line segment. The view cone of stationary sensors, which is represented as two triangles, is converted to two trapezoids for a mobile sensor that moves on a line segment to limit the computation. Assume that a sensor located in position O_{i-1} moves to position O_i in Figure 4. Point filtering for this line segment is provided by checking the points on the terrain whether they are in one of these trapezoids, $O_{i-1}A_{i-1}A_iC_i$ and $O_{i-1}B_{i-1}B_iC_i$ or not.

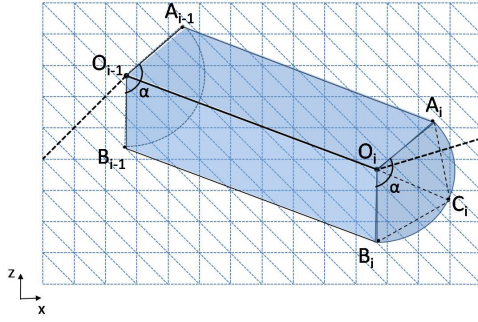


Fig. 4. Horizontal FOV test on a line segment of a path

- *Vertical FOV test.* The point set generated from horizontal FOV test is used in vertical FOV test as input. The tilt angle, upper and lower vertical viewing angles of the sensor and the location of points are used in vertical FOV test. Since the tilt angle parameter of a mobile sensor changes with respect to the slope of each polygon on a line segment, vertical FOV test for static sensors is repeated for all polygons on the line segment with different tilt angles. In Figure 5, vertical FOV test of a line segment is shown with different tilt angles.

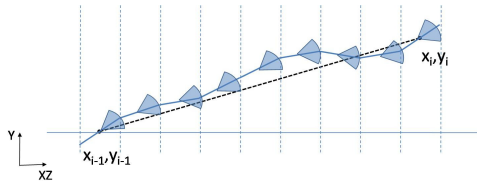


Fig. 5. Vertical FOV test applied on a line segment

All points that pass vertical FOV test are examined in order to determine whether there is any occlusion between the point and the polygon on the line segment. It is performed by a line-of-sight (LOS) algorithm.

Our framework supports both Janus and Bresenham Algorithms. LOS algorithm is applied to each position in the path. Finally, the visibility values of points that pass LOS test are calculated by using the visibility formula given in [18].

- *Visibility Computation.* The visibility component of the fitness function is equal to visibility of the given terrain, which is computed by adding the visibility of all polygons that is performed by the perception model explained in previous part. In Equation 3, W_{p_i} is the weight of polygon p_i , and $V(S, p_i)$ is the visibility value of polygon p_i by using the set of S mobile sensors.

$$V(A, S, P) = \frac{\sum_{p_i \in A} V(S, p_i) \times W_{p_i}}{\sum_{p_i \in A} W_{p_i}} \quad (3)$$

The visibility of a polygon is computed by Equation 4. Since a point can be recognized by multiple sensors and different locations of the same mobile sensor, maximum visibility value of the point is considered in calculating the visibility of a polygon. In Equation 4, the maximum visibility value of point b in polygon p_i is selected from the paths (R) of a set of mobile sensors (S) that maximizes the visibility value when sensor (s_j) passes polygon (p_k) on its path (R_{s_j}). The visibility value of polygon p_i is computed by the average of visibility of the selected polygon points, which are the three corners and the center of mass of the polygon.

$$V(S, p_i) = \frac{\sum_{b \in p_i} \max_{s_j \in S, p_k \in R_{s_j}} \{V_{s_j}(s_j, p_k, b)\}}{n} \quad (4)$$

2) *Cost Component:* The cost component is calculated by using the length of sensors' paths. Both the average of path costs of sensors ($d(R_s)$), and the ratio between the cost of the shortest path and the longest path ($\min(d(R_s)) / \max(d(R_s))$) are considered in Equation 5. If the length of paths are close to each other, the movements of sensors from initial to goal points may take approximately equal time. Here, λ constant is used to determine the importance of path length difference and $\lambda = 0.50$ unless otherwise specified.

$$D(R) = \frac{\sum_{s \in S} d(R_s)}{|S|} \times \left(\frac{\min_{s \in S} \{d(R_s)\}}{\max_{s \in S} \{d(R_s)\}} \right)^\lambda \quad (5)$$

$$d(R_s) = 2 - \left\{ \sum_{i \in R_s} d(m_i, m_{i+1}) / E(r) \right\}$$

To normalize path length of a sensor, it is divided by estimated path length value given in $E(r)$, which is computed by using horizontal field-of-view, the size of the terrain, and average value of all polygons' slopes (see Equation 6). In this equation, the area of the terrain by considering XZ plane is divided into the number of mobile sensors to calculate the estimated area covered by one sensor given in term APS (area per sensor). By equalizing the APS value to perceived area computed in horizontal FOV test to form an equation and solving this equation gives the expected path

length, $E(r)$. In Equation 6, κ constant is used to get an approximated value of estimated path length in 3-D.

$$E(r) = \frac{(APS - \Delta_s^2 \times \sin(\alpha/2))}{(2 \times \Delta_s \times \sin(\alpha/2))} \times \kappa \quad (6)$$

3) *Smoothness Component*: Although it is assumed that mobile sensor platforms have advanced capabilities, the turning angle of sensors must be greater than a predefined value to prevent possible problems in changing the direction. In calculation of smoothness component, the angle between any adjacent two line segments ($\alpha(l_i, l_{i+1})$) are computed and the narrow ones which may prevent turning are determined. In Equation 8, in order to increase the effect of the number of infeasible angles on smoothness, all values are multiplied with each other.

$$s(R_j) = \prod_{i \in R_j} \alpha(l_i, l_{i+1}) / \min(\alpha), \quad (7)$$

where $\alpha(l_i, l_{i+1}) < \min(\alpha)$

Then, smoothness of the solution is computed by taking the average of smoothness values of each path (see Equation 8).

$$S(R) = \frac{\sum_{i=1}^n s(R_i)}{n} \quad (8)$$

4) *Accessibility Component*: Accessibility component calculation for a solution is performed by testing the accessibility from the initial point to the target point for each line segment. In accessibility test, which is also used in probabilistic road map generation section, the slope of each polygon and the objects located on the path are considered. Accessibility component is calculated by dividing the number of polygons which fails the accessibility test ($o(p_i)$) to the number of polygons which is located on any of sensors' paths (R_s) in the solution.

$$C(R) = \frac{\sum_{s \in S} \sum_{i \in R_s} o(p_i)}{\sum_{s \in S} R_s} \quad (9)$$

C. Initial Population Generation

The first step of initial population generation is to select control points. In order to get variety in individuals of initial population, control points are randomly selected from sets of assigned control points and their alternatives which are determined in the global path planning stage. The alternatives are selected from control points' neighborhood region.

After the selection of control points, intermediate points are determined to generate sub-paths. In generating sub-paths, primary and secondary intermediate points, which are selected from nodes generated in PRM, are used. Primary intermediate points are the points on the shortest path between any adjacent two control points and secondary intermediate points are the points which connect any two of primary intermediate points by considering accessibility (see Figure 6). When selecting intermediate points, all primary and secondary intermediate points are ordered and divided into

two groups with respect to their locations. After determining the number of intermediate points to be selected, half of them are selected from the first group and the second half of them are selected from the second group randomly.

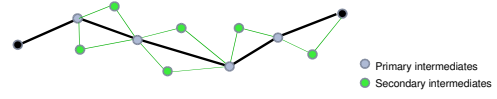


Fig. 6. Primary and secondary intermediate points that are used to generate sub-path

The final step in initial population generation is to determine the heading and tilt angles. To get maximum coverage on a path, the heading and tilt angles of sensor can be adaptively changed for each line segment. In the problem of multiple sensors, a region can be surveilled by more than one sensor. So, heading angle and/or tilt angle of one of these sensors can be oriented to a different region for making the solution more efficient. In our study, the heading and tilt angles are either set randomly or the heading angle is oriented to the direction of each line segment and tilt angle is set with the slope of the polygon that the sensor passes.

D. Crossover Operators

Since the number of control points on a path for a specific sensor is same for all individuals in a given solution, and the locations of those control points are in close vicinity due to the heuristic-based initial population generation, our crossover operation is only performed on sub-path level of individuals. Two different crossover operators are presented:

- *Visibility Based Crossover*: In this operator, the visibility value of each sub-path for a given sensor is computed in both parents firstly. For each sub-path pair, the one which has the highest visibility value is copied to the offspring. This operation is performed for all sensors given in the problem.
- *Random Crossover*: As in the previous method, the visibility values for sub-path pairs are computed. For each sub-path pair, sub-path for the offspring is randomly set from one of the parents with a probability of 0.4. For the rest of the offspring, sub-paths are selected from the parent which has higher fitness value. This operation is performed for all sensors.

E. Mutation Operators

We consider four different mutation operators where three of them are applied on intermediate points and the last one is for the control points.

- *Intermediate Point Insertion*: If the solution includes an infeasible path, which is due to narrow angle between two line segments or due to existence of a line segment which is inaccessible, a new intermediate point is located randomly within the rectangular area that bounds the infeasible line segment randomly. This process is repeated for a predefined number of trials in order to generate a feasible line segment; otherwise, the mutation operation is not performed. If this mutation operation

TABLE I
DEFAULT VALUES OF PARAMETERS USED IN GLOBAL PATH PLANNING

Parameter	Value
Percentage of sensors allocated in a terrain region	(80% - center) (20% - random location)
Minimum distance between any two control points	250
Minimum distance of control points to the boundaries of terrain	90
Maximum edge count in selection of control points	5
Maximum distance of control point to a path to assign it directly	40
Sensor quantity	3

TABLE II
PARAMETERS USED FOR GLOBAL PATH PLANNING WITH THEIR ALTERNATIVES

Node Quantity	Neighborhood Radius	VPD Constant	Control Point Quantity	Mission Weights		
				Visibility (w_v)	Cost (w_d)	Smoothness (w_s)
81	200	2.0	7	0.70	0.65	0.65
100	250	3.0	10	0.50	0.35	0.35
121		4.0		0.30		

TABLE III
OPTIMUM VALUES OF GLOBAL PATH PLANNING PARAMETERS ON A ROUGH TERRAIN

Mission Weights			Node Quantity	Neighborhood Radius	VPD Constant	Control Point Quantity
Visibility	Cost	Smoothness				
0.70	0.65	0.35	100	250	2.0	10
0.70	0.35	0.65	81	250	2.0	10
0.50	0.65	0.35	100	250	3.0	10
0.50	0.35	0.65	81	250	2.0	10
0.30	0.65	0.35	100	250	3.0	10
0.30	0.35	0.65	100	250	4.0	7

is selected and if there is not any infeasible path, a new intermediate point is inserted into the sub-path which has the minimum visibility with a predefined probability.

- *Intermediate Point Update:* If the solution includes an infeasible path, one of the intermediate points on the infeasible line segment is updated with another point around from the same region as in the previous approach. If there is no intermediate point update, a point is selected on the sub-path which has the minimum visibility with a probability of 25% and selected randomly with a probability of 75%. After the selection, its location is changed randomly within a feasible range.
- *Intermediate Point Deletion:* If there is an infeasible path in the solution, one of the intermediate points is removed that causes infeasible line segment. Otherwise, this operation is not performed.
- *Control Point Update:* The location of a randomly selected control point is changed with a point which is in the same neighborhood region with the removed one.

F. Collision Avoidance in Dynamic Environment

In our framework, a collision avoidance method is implemented to prevent collision among sensor platforms in a solution. It tests whether there is an intersection between the paths of any two sensor platforms. If there is an intersection, the arrival distances between initial points of sensors and the intersected point are calculated. If the difference between

arrival distances is less than a predefined threshold value, it is expected that there will be a collision occurrence between the sensors. To prevent collision, a new intermediate point is inserted to one of the paths around the intersection region. Because of changing the arrival distance of each sensor platform to the intersected point, this type of collision is avoided. In our experimental study, this module is applied on all population at every 200 iterations.

IV. EXPERIMENTAL STUDY

In this section, we present the results of computational experiments in order to identify the settings of various parameters and to evaluate the performance of our algorithm. For all experiments, the number of sensors in the problem and their initial and goal positions are predetermined. The experiments in this study are conducted in two phases.

A. Experiments on Global Path Planning

Performance evaluation of this phase of our algorithm is performed with different missions and various terrain types. The parameters used in global path planning are separated into two groups, which are predefined parameters and experimental parameters. The default values of constant parameters do not change for any computational experiment and they are listed in Table I. In Table II, the parameters to be examined are given with their alternatives.

In pre-experimental stage, the variables which are listed in Table II are evaluated to determine the most suitable values. In these experiments, six different missions are tested

TABLE IV
OPTIMUM VALUES OF GLOBAL PATH PLANNING PARAMETERS ON A SMOOTH TERRAIN

Mission Weights			Node Quantity	Neighborhood Radius	VPD Constant	Control Point Quantity
Visibility	Cost	Smoothness				
0.70	0.65	0.35	100	200	2.0	10
0.70	0.35	0.65	81	200	3.0	10
0.50	0.65	0.35	100	200	2.0	10
0.50	0.35	0.65	81	200	3.0	10
0.30	0.65	0.35	100	200	2.0	10
0.30	0.35	0.65	100	200	2.0	10

TABLE V
DEFAULT VALUES OF PARAMETERS USED IN LOCAL PATH PLANNING

Parameter	Value
Population size	50
Generation limit	2000
Tournament size	5
Minimum rotation angle	30°
Intermediate point ratio copied from primary and secondary intermediate points	0.60
Percentage of intermediate points in the solution (primary,secondary)	66% - 33%
Coefficient of the importance of path length difference in cost component	0.50
Crossover ratio in random crossover	0.4
Accessibility weight (w_c)	5.0
Maximum difference of arrival distances in collision detection	25
Iteration count required to perform collision avoidance	200

TABLE VI
PARAMETERS USED FOR LOCAL PATH PLANNING WITH THEIR ALTERNATIVES

Crossover	Mutation	Mission Weights		
		Visibility (w_v)	Cost (w_d)	Smoothness (w_s)
1. Visibility Based Crossover	1. Intermediate Point Insertion	0.70	0.65	0.65
2. Random Crossover	2. Intermediate Point Update	0.50	0.35	0.35
	3. Intermediate Point Deletion	0.30		
	4. Control Point Update			
	5. Random (1 - 4)			

with 432 combinations of parameters where the tests are performed on different terrain types (rough and smooth terrains). The most suitable values of parameters are listed for rough and smooth terrains in Table III and Table IV, respectively. Maximum slope of climbing is set to 45° for all experiments presented in this paper.

B. Experiments on Local Path Planning

In the second phase, performance evaluation is done with different missions on both rough and smooth terrains that are used in previous stage by using the specified values of global path planning parameters. The parameters of local path planning are also separated into two groups which are predefined parameters and experimental parameters. Those parameters with their alternative values are listed in Table V and Table VI, respectively.

In order to find best values of local path planning parameters in Table VI, a pre-experimentation phase was established. In this experiment, 120 combinations of parameters are tested for 6 different missions; and each combination is run with 30 replications. In Table VII and Table VIII, the optimum values of these parameters are given for rough and smooth terrains respectively.

The specified parameters of local path planning, which

TABLE VII
BEST VALUES OF LOCAL PATH PLANNING PARAMETERS ON A ROUGH TERRAIN

Mission Weights			Crossover	Mutation
Visibility	Cost	Smoothness		
0.70	0.65	0.35	1	1
0.70	0.35	0.65	1	1
0.50	0.65	0.35	1	5
0.50	0.35	0.65	1	5
0.30	0.65	0.35	1	5
0.30	0.35	0.65	1	5

TABLE VIII
BEST VALUES OF LOCAL PATH PLANNING PARAMETERS ON A SMOOTH TERRAIN

Mission Weights			Crossover	Mutation
Visibility	Cost	Smoothness		
0.70	0.65	0.35	1	5
0.70	0.35	0.65	1	1
0.50	0.65	0.35	1	5
0.50	0.35	0.65	1	5
0.30	0.65	0.35	1	5
0.30	0.35	0.65	1	5

are listed in Tables VII, and VIII, show that visibility-based crossover significantly outperforms other crossover operator; and applying randomly selected mutation operator outperforms all mutation operators presented in this study. Finally, the overall performance of the algorithm is examined by considering the specified values of global and local path planning parameters. The terrains generated for previous experiments are used to evaluate the effect of local path planning on paths that are generated by global path planning. In Table IX and Table X, the performance comparisons of local path planning and global path planning algorithms are given where maximum slope of climbing is 45° .

TABLE IX

PERFORMANCE RESULTS OF DIFFERENT MISSION TYPES ON A ROUGH TERRAIN

Mission Weights			Fitness	
Visibility	Cost	Smoothness	GPP	LPP
0.70	0.65	0.35	0.5905	0.6954
0.70	0.35	0.65	0.6311	0.7093
0.50	0.65	0.35	0.6216	0.7385
0.50	0.35	0.65	0.6651	0.7150
0.30	0.65	0.35	0.6743	0.8386
0.30	0.35	0.65	0.6960	0.8300

TABLE X

PERFORMANCE RESULTS OF DIFFERENT MISSION TYPES ON A SMOOTH TERRAIN

Mission Weights			Fitness	
Visibility	Cost	Smoothness	GPP	LPP
0.70	0.65	0.35	0.5899	0.6901
0.70	0.35	0.65	0.6229	0.7149
0.50	0.65	0.35	0.6200	0.7762
0.50	0.35	0.65	0.6674	0.7271
0.30	0.65	0.35	0.6618	0.8435
0.30	0.35	0.65	0.7010	0.8301

In the performance comparison of local path planning (LPP) phase with global path planning (GPP) phase, fitness function presented in LPP phase is used. For all missions, LPP phase improves the quality of solutions generated in GPP phase by considering the components of fitness function. As a specific example, when the weights of visibility, cost and smoothness are selected as 0.30, 0.65 and 0.35, respectively, the quality of solution is improved 24% in a rough terrain and 27% in a smooth terrain. Since the visibility component is dominant in GPP phase, LPP phase improves the quality of solutions by considering the cost and smoothness criteria, additionally.

It was observed that GPP algorithm adapts its parameters to the environment to get maximum contribution. Therefore, terrain and sensor characteristics do not have a great effect on the overall performance of the algorithm by considering all missions. It is shown by considering the performance results in Tables IX, and X.

V. CONCLUSIONS

We present a novel hybrid framework for the path planning of multiple mobile sensors on a 3-D terrain. The contribu-

tions of this paper can be grouped in two-folds. In the first phase of the framework, the control points are determined with the help of the probabilistic road map (PRM) and modified visibility-per-distance function. In the second phase, a new hybrid evolutionary algorithm is presented for utilizing intermediate points that are between the control points in order to complete the routes of mobile sensors.

ACKNOWLEDGMENTS

This research was supported by The Scientific and Technological Research Council of Turkey (TUBITAK) with a research grant (No. 106E159).

REFERENCES

- [1] S. L. Valle, *Planning Algorithms*. Cambridge University Press, 2006.
- [2] P. Xavier, "Approximation algorithms for optimal kinodynamic robot plans," PhD Dissertation, Computer Science Department, Cornell University, Ithaca, New York, 1992.
- [3] P. Bhattacharya and M. Gavrilo, "Voronoi diagram in optimal path planning," in *The 4th International Symposium on Voronoi Diagrams in Science and Engineering*, United Kingdom, 2007.
- [4] D. Zhu and J. Latombe, "New heuristic algorithms for efficient hierarchical path planning," in *IEEE Transaction on Robotics and Automation I*, 1991.
- [5] J. Latombe, *Robot Motion Planning*. Kluwer, Boston, 1991.
- [6] L. Kavvaki, P. Svetska, J. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," in *IEEE Transactions on Robotics and Automation*, 1996.
- [7] S. L. Valle, "Rapidly exploring random trees: a new tool for path planning," Technical Report, Computer Science Department, Iowa State University, Iowa, USA, 1998.
- [8] J. Barraquand and J. Latombe, "A monte-carlo algorithm for path planning with many degrees of freedom," in *Proceedings IEEE International Conference on Robotics and Automation*, 1990.
- [9] C. Luo, S. Yang, and X. Yuan, "Real-time area covering operations with obstacle avoidance for cleaning robots," in *Proceedings of the 2002 IEEE International Conference on Intelligent Robots and Systems*, Switzerland, 2002.
- [10] H. Najjaran and N. Kircanski, "Path planning for a terrain scanner robot," in *Proceedings of 31st International Symposium of Robotics*, Canada, 2000.
- [11] S. Hert, S. Tiwari, and V. Lumelsky, "A terrain covering algorithm for an auv," *Autonomous Robots*, vol. 3, no. 2, pp. 91–119, 1996.
- [12] M. Ollis and A. Stentz, "First results in vision based crop line tracking," in *Proceedings of the IEEE International Conference on Robotics Automation*, Minneapolis, 1996, pp. 951–956.
- [13] A. Pirzadeh and W. Snyder, "A unified solution to coverage and search in explored and unexplored terrains using indirect control," in *Proceedings of IEEE International Conference on Robotics Automation*, 1990, pp. 2113–2119.
- [14] S. Yang and C. Luo, "A neural network approach to complete coverage path planning," *IEEE Transactions on Systems, Man and Cybernetics - Part B*, vol. 34, no. 1, 2004.
- [15] E. Acar and H. Choset, "Robust sensor-based coverage of unstructured environments," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001.
- [16] J. Tian, Y. Zheng, H. Zhu, and L. Shen, "A mpc and genetic algorithm based approach for multiple uavs cooperative search," in *International Conference on Computational Intelligence and Security*, China, 2005.
- [17] M. Mohan, R. Sawhney, K. M. Krishna, K. Srinathan, and M. B. Srikanth, "Covering hostile terrains with partial and complete visibilities: On minimum distance paths," in *IEEE International Conference on Intelligent Robots and Systems*, 2008, pp. 2572–2577.
- [18] H. Topcuoglu, M. Ermis, and M. Sifyan, "Hybrid evolutionary algorithms for sensor placement on a 3d terrain," in *The International Conference on Intelligent Systems Design and Applications (ISDA)*, Italy, 2009.

TÜBİTAK PROJE ÖZET BİLGİ FORMU

Proje No: 106E159
Proje Başlığı: Karma Evrimsel Algoritmalar Yardımı ile Temsili Olarak Üretilmiş bir Alan Üzerinde Sensörlerin Konumlandırılması ve Parametrelerinin Tayini
Proje Yürütücüsü ve Araştırmacılar: Prof. Dr. Haluk Topcuoglu (Proje Yürütücüsü) Yrd. Doç. Dr. Murat Ermiş (Araştırmacı) Mesut Sifyan (Araştırmacı)
Projenin Yürütüldüğü Kuruluş ve Adresi: Marmara Üniversitesi, Bilgisayar Mühendisliği Bölümü Göztepe Kampüsü, 34722 Kadıköy / ISTANBUL
Destekleyen Kuruluş(ların) Adı ve Adresi: TÜBİTAK, Atatürk Bulvarı, No:22 Kavaklıdere/ANKARA 06100
Projenin Başlangıç ve Bitiş Tarihleri: 01/02/2007- 30/10/2009
Öz (en çok 70 kelime): Projemizde, temsili olarak üretilen 3-boyutlu bir arazi için sensörlerin konumlandırılması ve parametrelerinin tayinini içeren çok ölçütlü fayda teorisi tabanlı yeni bir model tasarlanmış ve gerçekleştirilmiştir. Bu modele yönelik olarak, özgün bir karma evrimsel algoritma geliştirilmiştir. Yapılan deneysel çalışmalar, çözümümüzün çeşitli askeri uygulamalarda için etkin bir şekilde kullanılabileceğini göstermektedir. Projemiz kapsamında, hareketli sensörlerin güzergah planlama problemi için, çıkılabilecek en yüksek eğim ve en düşük dönüş açısı gibi kısıtların dikkate alındığı yeni bir çözüm yöntemi de geliştirilmiştir.
Anahtar Kelimeler: Karma Evrimsel Algoritmalar, Sensör Planlanması, Çok-Ölçütlü Eniyileme, Çok-Ölçütlü Fayda Teorisi, Yerel Arama, Olasılıksal Yol Haritası
Fikri Ürün Bildirim Formu Sunuldu mu? Evet ☐ Gerekli Değil ☒ Fikri Ürün Bildirim Formu'nun tesliminden sonra 3 ay içerisinde patent başvurusu yapılmalıdır.

Projeden Yapılan Yayınlar:

- “Positioning and Utilizing Sensors on a 3D Terrain: Part I – Theory and Modeling”, Haluk Rahmi Topcuoglu, Murat Ermis and Mesut Sifyan, IEEE Transactions on Systems, Man and Cybernetics: Part C: Applications and Review. *(kabul edildi)*
- “Positioning and Utilizing Sensors on a 3D Terrain: Part II – Solving with Hybrid Evolutionary Algorithm”, Haluk Rahmi Topcuoglu, Murat Ermis and Mesut Sifyan, IEEE Transactions on Systems, Man and Cybernetics: Part C: Applications and Review. *(kabul edildi)*
- “A New 3-D Wireless Multimedia Sensor Network Simulation Environment for Connected Coverage Problems”, Haluk Rahmi Topcuoglu, Murat Ermis, İlker Bekmezci and Mesut Sifyan, Simulation: Transactions of the Society for Modeling & Simulation International. *(düzeltme aşamasında)*
- “On the Optimal Deployment of Wireless Multimedia Sensor Networks Using Genetic Algorithms for Three-Dimensional Terrains”, Haluk Rahmi Topcuoglu, Murat Ermis, İlker Bekmezci and Mesut Sifyan, Computer Networks. *(değerlendirme aşamasında)*
- “Hybrid Evolutionary Algorithms for Sensor Placement on a 3D Terrain”, Haluk Rahmi Topcuoglu, Murat Ermis and Mesut Sifyan, International Conference on Intelligent Systems Design and Applications (ISDA), Pisa, İtalya, Kasım 2009.
- “Path Planning for Mobile Sensor Platforms Using Hybrid Evolutionary Algorithms on a 3D Terrain”, Mesut Sifyan, Haluk Rahmi Topcuoglu ve Murat Ermis, 2010 IEEE World Congress on Computational Intelligence (WCCI 2010), Barcelona, İspanya, Temmuz 2010.