

MIMO-FBMC/OQAM İletim Teknolojisi İçin Ayrık Karga Arama Algoritması Tabanlı Gelişmiş Bir Sembol Algılama Yöntemi

Şakir Şimşir*¹ 

*¹ Nevşehir Hacı Bektaş Üniversitesi Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği, NEVŞEHİR

(Alınış / Received: 11.06.2024, Kabul / Accepted: 01.08.2024, Online Yayınlanma / Published Online: 30.08.2024)

Anahtar Kelimeler

Karga Arama Algoritması,
FBMC/OQAM,
Sembol Algılama,
MIMO,
Maksimum Olasılık

Öz: Son yıllarda geliştirilmiş önde gelen iletim teknolojileri arasında yer alan çoklu-giriş çoklu-çıkış - filtre bankası çoklu taşıyıcı/ofset dördün genlik modülasyonu (MIMO-FBMC/OQAM), kablosuz iletişime has birçok problemi çözme yetisine sahip olmakla birlikte, alıcısında etkili bir sembol dedektörüne gereksinim duymaktadır. Bu çalışmada, MIMO-FBMC/OQAM sistemi için hem düşük karmaşık hem de yüksek sembol algılama performansına sahip yeni bir sembol dedektörü geliştirilmiştir. Bunun için, sembol vektörlerini kusursuz bir şekilde algılayabilen ancak bu işlemi oldukça yüksek bir araştırma maliyetine sebep olan kapsamlı arama prosedürü kullanarak gerçekleştiren klasik maksimum olasılık (ML) dedektörü modifiye edilmiştir. Söz konusu modifikasyonda, yüksek maliyetli kapsamlı arama prosedürü devre dışı bırakılarak, yerine sembol vektörlerinin optimizasyon işlemi gerçekleştirmek üzere ayrık karga arama algoritması (DCSA) entegre edilmiştir. Böylelikle, geleneksel ML yönteminin oldukça yüksek olan araştırma maliyeti düşürülmekle kalmayıp, optimuma yakın bir sembol algılama performansı elde edilmiştir.

An Advanced Symbol Detection Method Based on Discrete Crow Search Algorithm for MIMO-FBMC/OQAM Transmission Technology

Keywords

Crow Search Algorithm,
FBMC/OQAM,
Symbol Detection,
MIMO,
Maximum Likelihood

Abstract: Although multiple-input multiple-output – filter bank multicarrier/offset quadrature amplitude modulation (MIMO-FBMC/OQAM), which is among the prominent transmission technologies developed in recent years, has the capability of solving various problems peculiar to wireless communication, it requires an efficient symbol detector at its receiver. In this study, a new symbol detector with both low complexity and high symbol detection performance has been developed for MIMO-FBMC/OQAM system. For this, the classical maximum likelihood (ML) detector, which can detect the symbol vectors perfectly but performs this process by using an exhaustive search procedure that causes a quite high searching cost, was modified. In this modification, the high-cost exhaustive search procedure was disabled and instead, discrete crow search algorithm (DCSA) was integrated to carry out the optimization process of the symbol vectors. Thus, not only the pretty high search cost of the traditional ML method was reduced, but also a near-optimum symbol detection performance was achieved.

*İlgili Yazar, email: sakirsimsir@nevsehir.edu.tr

1. Giriş

Filtre bankası çoklu taşıyıcı/ofset dördün genlik modülasyonu (FBMC/OQAM: Filter bank multicarrier/offset quadrature amplitude modulation) [1, 2], gelecek nesil kablosuz sistemlerde geleneksel dikgen frekans bölme çözümlerinin (OFDM: Orthogonal frequency division multiplexing) [3, 4] yerini alma potansiyeline sahip önemli dalga formlarından bir tanesidir. FBMC/OQAM dalga formu, OFDM sistemine ait eksiklikleri ortadan kaldıracak benzersiz özellikleri ile öne çıkmaktadır. Örneğin; önemli oranda semboller arası girişime neden olan yüksek seviyeli yan loblara sahip olmak, OFDM sisteminin en önemli dezavantajları arasında yer alırken, filtre bankalarının kullanımı sayesinde elde edilen düşük yan loblar, FBMC/OQAM sisteminde neredeyse ihmal edilebilir düzeyde semboller arası girişime yol açmaktadır. Bunun yanı sıra, FBMC/OQAM sisteminde döngüsel önek kullanımına ihtiyaç duyulmamaktadır. Bu nedenle, OFDM sistemine göre daha yüksek veri oranlarına ulaşmak mümkün hale gelmektedir. Ayrıca FBMC/OQAM sistemi, iletim sinyalinin çok düşük bitişik kanal sızıntı oranına sahip olması sayesinde bilişsel radyo uygulamalarıyla tam olarak uyumludur. Öte yandan hem frekans hem de zamanda iyi lokalize edilmiş darbe şekli kullanımı, FBMC/OQAM'yi mobil ortam için daha uygun bir iletişim sistemi haline getirmektedir [1, 2]. Yukarıda bahsedilen tüm bu üstün özelliklerin, çoklu antenler üzerinden veri iletimi gerçekleştirilerek çoklu-giriş çoklu-çıkış (MIMO: Multiple-input multiple output) teknolojisinin avantajlarıyla birleştirilmesi durumunda, FBMC/OQAM sisteminin kapasitesi çok daha yüksek bir seviyeye ulaşmaktadır [5, 6]. Sinyallerin çoklu anten yapısı üzerinden iletilmesi, sistem performansına önemli faydalar sağlamaktadır. Çoklu anten kullanımının sağladığı en önemli faydalardan birisi, çok yönlü kanalların sönmeme etkilerinin ciddi oranda azaltılmasıdır. Böylelikle, daha düşük bit hata oranı (BER: Bit error rate) ile veri iletimi gerçekleştirmek mümkün hale gelmektedir. MIMO teknolojisinin bir diğer önemli avantajı ise kapasite artışıdır. Anten sayısındaki artış, sistem kapasitesinin yükselmesine yol açmaktadır. Bunun sonucunda, daha yüksek veri oranlarında iletişim sağlanabilmektedir [5, 6].

Diğer taraftan, MIMO-FBMC/OQAM alıcısında sembol vektörlerini en az hata ile algılayan ve bunu çok fazla işlem yüküne neden olmadan gerçekleştirebilen verimli bir sembol dedektörüne gereksinim duyulmaktadır. Sıfır zorlama (ZF: Zero forcing) dedektörü [7], kolay uygulanabilir yapısı nedeniyle sembol algılama problemini çözmek için ilk akla gelen yöntemlerden birisidir. Ancak söz konusu dedektör, çok basit yapısı sayesinde düşük bir karmaşıklığa sahip olsa da hiçbir sistemde tolere edilemeyecek bazı kritik zayıflıklara sahiptir. Zorlu kanal koşullarına olan hassasiyet, ZF yönteminin öne çıkan zayıflıklarından bir tanesidir. Bu zayıflık, ilgili yöntemin sembol algılama performansını önemli ölçüde düşürmektedir. Bunun yanı sıra, anten sayısı arttıkça ZF dedektörünün sembol algılama performansı daha da kötüye gitmektedir [7]. Maksimum olasılık (ML: Maximum likelihood) stratejisi [8], ZF ile birlikte farklı iletim teknolojilerinde en çok kullanılan yöntemlerden birisi olması sebebiyle, MIMO-FBMC/OQAM sisteminde sembol dedektörü olarak faydalanmak için ciddi bir seçenek olarak değerlendirilebilir. ZF dedektörünün aksine, sembol algılama performansı ML yönteminin en güçlü noktasıdır. En zayıf yanı ise sahip olduğu yüksek araştırma karmaşıklığıdır. Kullanmış olduğu kapsamlı arama prosedürü, ML stratejisinin optimum çözüme ulaşmasını garanti ederek kusursuz bir sembol algılama performansı sergilemesini sağlarken, ilgili arama prosedürü aynı zamanda araştırma karmaşıklığında aşırı bir artışa neden olmaktadır [8]. Sistem karmaşıklığında minimum artışa neden olarak sembol vektörlerini en yüksek doğrulukta algılama kapasitesine sahip verimli bir sembol dedektörü ihtiyacı göz önünde bulundurulduğunda ne ZF ne de ML dedektörü MIMO-FBMC/OQAM sistemindeki sembol algılama problemi için ideal bir çözüm olamaz.

Geleneksel ML yönteminde kapsamlı arama prosedürü kullanılarak sembol algılama işlemi gerçekleştirilirken, iletilme olasılığı en yüksek olan optimum sembol vektörü bulunmaya çalışılır. Bunun için iletilme olasılığı bulunan her bir sembol kombinasyonu, kanalın bozucu etkisine maruz kalarak alıcıya ulaşan sembol vektörü ile arasındaki Öklid mesafesi bakımından test edilir. En düşük Öklid mesafesine sahip olan sembol kombinasyonu, en yüksek iletilme olasılığına sahip sembol vektörü olarak ML dedektörü çıkışından elde edilir [8]. Diğer taraftan, ayrık araştırma uzayında yer alan her bir sembol kombinasyonu için Öklid hesabı gerçekleştirilerek optimum sembol vektörünü elde etmeye çalışmak yerine, sembol vektörlerini optimize ederek çok daha düşük bir araştırma maliyeti ile optimuma yakın bir çözüme ulaşmak mümkündür. ML dedektöründeki söz konusu optimizasyon işlemi yerine getirmek için, düşük araştırma maliyetleriyle tatmin edici sonuçlara ulaşabilme yeteneklerinden dolayı metasezgisel algoritmalar en doğru seçim olacaktır. Makalede ele alınan sembol algılama probleminde, her biri farklı bir karmaşık dördün genlik modülasyonu (QAM: Quadrature amplitude modulation) sembol kombinasyonundan oluşan sembol vektörleri direkt olarak ayrık uzayda optimize edilmektedir. Bu amaçla, sembol vektöründe yer alan her bir QAM sembolü, eşdeğeri olan ikili sayıya dönüştürülür. Böylelikle, her bir sembol vektörü ayrık uzayda optimize edilmek üzere ikili bit dizisi haline getirilmiş olur. Elde edilen ikili bit dizilerini optimize edebilmek için metasezgisel algoritmaların etkili ayrık versiyonlarına ihtiyaç duyulmaktadır. Bu ihtiyaç göz önünde bulundurularak, ikili uzayda etkili bir araştırma yeteneğine sahip olan ayrık karga arama algoritması (DCSA: Discrete crow search algorithm) [9] önerilmiştir. Bu çalışmada, klasik CSA algoritmasının [10] söz konusu ayrık varyantı ML dedektörüne sembol optimizatörü olarak entegre edildikten sonra, DCSA-ML olarak adlandırılan

yeni bir sembol algılama stratejisi geliştirilmiştir. Geliştirilen bu yeni strateji, ML ve ZF gibi geleneksel sembol dedektörlerinin yanı sıra ayırık ikili harmoni arama [11] tabanlı ML (DBHS-ML: Discrete binary harmony search-ML) ve ayırık yapay arı kolonisi [12] tabanlı ML (disABC-ML: Discrete artificial bee colony-ML) gibi gelişmiş ML stratejileri ile sembol algılama performansı bakımından karşılaştırılmıştır.

Literatürde farklı metasezgisel yaklaşımlar kullanılarak çeşitli iletişim sistemlerinde sembol vektörlerini algılamaya yönelik bazı çalışmalar mevcuttur. [13]'de, MIMO iletişim sistemlerinde sembol vektörlerinin algılanması için parçacık sürü optimizasyonu (PSO: Particle swarm optimization) algoritmasının standart ve ikili versiyonlarının kullanıldığı birkaç adet düşük karmaşıklık ML stratejisi geliştirilmiştir. [14]'te yazar, klasik ML yönteminin sahip olduğu aşırı karmaşıklığı azaltıp MIMO-dikgen olmayan çoklu erişim (MIMO-NOMA: MIMO-Non orthogonal multiple access) sisteminde potansiyel olarak kullanılabilir hale getirmek için geri izleme arama algoritmasından (BSA: Back-tracking search algorithm) yararlanmışır. [15]'te yazarlar, masif MIMO sistemindeki sembol algılama problemini çözmek için ABC algoritmasının doğrusal sembol dedektörü ile birlikte kullanıldığı bir sembol algılama stratejisi önermişlerdir. [16]'te, geniş-MIMO sistemlerde sembol dedektörü olarak kullanabilmek için PSO ve karınca koloni optimizasyon (ACO: Ant colony optimization) algoritmaları birleştirilerek hibrit bir optimizasyon algoritması geliştirilmiştir. [17]'da yazarlar, MIMO-OFDM alıcısında sembol vektörlerini düşük bir hesaplama yükü ile elde edebilmek için farksal gelişim (DE: Differential evolution) algoritması tabanlı ML stratejisi geliştirmişlerdir.

Bu çalışmanın literatüre olan temel katkıları aşağıdaki gibi özetlenebilir:

1. Ayırık karga arama algoritması, haberleşme alanında karşılaşılan sembol algılama problemine literatürde ilk defa uygulanmıştır.
2. MIMO-FBMC/OQAM sisteminde sembol algılama probleminin çözümü için DCSA-ML olarak isimlendirilen geleneksel ML yönteminin modern bir versiyonu geliştirilerek yeni bir yaklaşım sunulmuştur. Bu yaklaşımda, DCSA isimli etkili bir ayırık CSA varyantı sembol optimizatörü olarak kullanılmakta olup, optimum sembol vektörüne ulaşmak için sistematik bir arama işlemi gerçekleştirilmektedir.
3. Önerilen DCSA algoritmasının ML dedektörüne entegre edilmesiyle birlikte, büyük miktarda karmaşıklık kazancı elde edilmiştir.
4. DCSA-ML yöntemi, üstün yakınsama yeteneği sayesinde DBHS-ML, disABC-ML ve geleneksel ZF dedektörlerinden daha iyi bir bit hata oranı performansı sergilemiştir.

Makalenin bundan sonraki kısmı şu şekilde düzenlenmiştir: 2. Bölümde, MIMO-FBMC/OQAM sistemi açıklanmıştır. 3. Bölümde, MIMO-FBMC/OQAM sistemindeki sembol algılama problemi matematiksel olarak tanımlanmıştır. 4. Bölümde, DCSA algoritması sunulmuştur. 5. Bölümde, geliştirilmiş olan DCSA-ML stratejisi anlatılmıştır. 6. Bölümde simülasyon sonuçları verilmiş olup, 7. Bölümde makale sonlandırılmıştır.

2. MIMO-FBMC/OQAM İletim Sisteminin Çalışma Prensibi

Bu bölümde, MIMO-FBMC/OQAM iletim teknolojisi kullanılarak sinyallerin nasıl iletildiği ve iletilen sinyallerin alıcıda tekrardan nasıl elde edildiği en sade haliyle anlatılmıştır [5]. FBMC/OQAM sisteminde $\alpha_{n,t}$ şeklinde ifade edilen gerçek değerli sembollerin çoklu anten yerine tek bir anten kullanılarak iletilmesi durumunda, alıcıya ulaşan $y_{n,t}$ sinyalinin matematiksel ifadesi aşağıdaki gibi olmaktadır:

$$y_{n,t} \approx h_{n,t} \cdot (\alpha_{n,t} + j \cdot \beta_{n,t}) + \theta_{n,t} \quad (1)$$

Burada parantez içerisindeki ifadenin imajiner kısmını oluşturan $\beta_{n,t}$ içsel girişimi temsil ederken, $\theta_{n,t}$ alınan sinyale eklenmiş olan gürültü bileşenine karşılık gelmektedir. $h_{n,t}$ ise kanal katsayılarını ifade etmektedir. Bunun yanında, yukarıdaki denklemde alt simge olarak kullanılan n ve t , sırasıyla alt taşıyıcı ve zaman indisleridir. Öte yandan, FBMC/OQAM sisteminde çoklu anten üzerinden iletim gerçekleştirmek üzere vericide A_T adet, alıcıda ise A_R adet anten kullanıldığı takdirde, alıcının j 'inci anteninde elde edilen sinyal aşağıdaki gibi tanımlanmaktadır:

$$y_{n,t}^{(j)} = \sum_{i=1}^{A_T} h_{n,t}^{(ji)} \cdot (\alpha_{n,t}^{(i)} + j \cdot \beta_{n,t}^{(i)}) + \theta_{n,t}^{(j)} \quad (2)$$

Burada i verici anten indisi, j ise alıcı anten indisidir. Örneğin, alıcı tarafta j 'inci anten tarafından alınan sinyal $y_{n,t}^{(j)}$ şeklinde ifade edilirken, j 'inci alıcı anten ile i 'inci verici anten arasındaki kanal katsayısı $h_{n,t}^{(ji)}$ tarafından temsil edilmektedir. Denklem (2)'nin matris tabanlı gösterimi aşağıdaki gibidir:

$$\begin{bmatrix} y_{n,t}^{(1)} \\ \vdots \\ y_{n,t}^{(A_R)} \end{bmatrix} = \underbrace{\begin{bmatrix} h_{n,t}^{(11)} & \cdots & h_{n,t}^{(1A_T)} \\ \vdots & \ddots & \vdots \\ h_{n,t}^{(A_R 1)} & \cdots & h_{n,t}^{(A_R A_T)} \end{bmatrix}}_{\mathbf{H}_{n,t}} \underbrace{\begin{bmatrix} \alpha_{n,t}^{(1)} + j \cdot \beta_{n,t}^{(1)} \\ \vdots \\ \alpha_{n,t}^{(A_T)} + j \cdot \beta_{n,t}^{(A_T)} \end{bmatrix}}_{\alpha_{n,t} + j \cdot \beta_{n,t}} + \underbrace{\begin{bmatrix} \theta_{n,t}^{(1)} \\ \vdots \\ \theta_{n,t}^{(A_R)} \end{bmatrix}}_{\boldsymbol{\theta}_{n,t}} \quad (3)$$

$$\mathbf{Y}_{n,t} = \mathbf{H}_{n,t} \cdot (\boldsymbol{\alpha}_{n,t} + j \cdot \boldsymbol{\beta}_{n,t}) + \boldsymbol{\theta}_{n,t} \quad (4)$$

Burada alınan sinyaller, kanal katsayıları, içsel girişimli gerçek değerli semboller ve gürültü bileşenleri sırasıyla $A_R \times 1$, $A_R \times A_T$, $A_T \times 1$ ve $A_R \times 1$ boyutlu matrisler ile ifade edilmektedir.

3. Problemin Tanımlanması

3.1. FBMC/OQAM için matris tabanlı sistem açıklaması

FBMC/OQAM sistemini matris formunda ifade ederek sembol algılama problemini daha sade ve anlaşılır kılmak mümkündür [6]. FBMC/OQAM vericisinde kullanılan prototip filtre, her biri D sayıda zaman örneği $\mathbf{g}_{n,t} \in \mathbb{C}^{D \times 1}$ iletim vektörlerinin birleşiminden oluşan $\mathbf{G} \in \mathbb{C}^{D \times NT}$ iletim matrisi şeklinde aşağıdaki gibi tanımlanabilmektedir:

$$\mathbf{G} = [\mathbf{g}_{1,1} \ \mathbf{g}_{2,1} \ \cdots \ \mathbf{g}_{N,1} \ \mathbf{g}_{1,2} \ \mathbf{g}_{2,2} \ \cdots \ \mathbf{g}_{N,2} \ \cdots \ \mathbf{g}_{1,T} \ \mathbf{g}_{2,T} \ \cdots \ \mathbf{g}_{N,T}] \quad (5)$$

Yukarıdaki ifadeden de anlaşılacağı üzere, \mathbf{G} matrisinde yer alan $\mathbf{g}_{n,t}$ iletim vektörlerinin toplam sayısı $N \times T$ 'ye eşit olmaktadır. Burada N alt taşıyıcı sayısına karşılık gelirken, T sembol sayısını belirtmektedir. Matris tabanlı tanımlama, aşağıda gösterildiği gibi gerçek değerli semboller için de yapılabilir:

$$\boldsymbol{\alpha} = \text{vec} \left\{ \begin{bmatrix} \alpha_{1,1} & \cdots & \alpha_{1,T} \\ \vdots & \ddots & \vdots \\ \alpha_{N,1} & \cdots & \alpha_{N,T} \end{bmatrix} \right\} = [\alpha_{1,1} \ \alpha_{2,1} \ \cdots \ \alpha_{N,1} \ \alpha_{1,2} \ \alpha_{2,2} \ \cdots \ \alpha_{N,2} \ \cdots \ \alpha_{1,T} \ \alpha_{2,T} \ \cdots \ \alpha_{N,T}]^T \quad (6)$$

Burada $\boldsymbol{\alpha} \in \mathbb{C}^{NT \times 1}$ ile temsil edilen sembol vektörü, Denklem (5)'te tanımlanan iletim matrisi ile çarpılarak $\mathbf{s} \in \mathbb{C}^{D \times 1}$ şeklinde ifade edilen iletim sinyali üretilmektedir:

$$\mathbf{s} = \mathbf{G}\boldsymbol{\alpha} \quad (7)$$

İletim işleminden sonra alıcı girişine ulaşan sinyalin matematiksel ifadesi aşağıdaki gibi olacaktır:

$$\mathbf{r} = \mathbf{H}\mathbf{s} + \boldsymbol{\theta} = \mathbf{H}\mathbf{G}\boldsymbol{\alpha} + \boldsymbol{\theta} \quad (8)$$

Burada $\mathbf{H} \in \mathbb{C}^{D \times D}$ konvolüsyon matrisi, çok yollu kanalı modellemektedir. $\mathbf{r} \in \mathbb{C}^{D \times 1}$ alıcıya gelen sinyale karşılık gelirken, $\boldsymbol{\theta}$ Gauss gürültüsünü sembolize etmektedir. Son olarak, aşağıdaki formülasyon kullanılarak alınan semboller elde edilmektedir:

$$\mathbf{y} = \mathbf{G}^H \mathbf{r} = \mathbf{G}^H \mathbf{H}\mathbf{G}\boldsymbol{\alpha} + \mathbf{G}^H \boldsymbol{\theta} \quad (9)$$

3.2. FBMC/OQAM sisteminde blok frekans yayılımı

FBMC/OQAM sistemindeki diklik kısıtlamasının neden olduğu içsel girişim, OFDM sistemine kolaylıkla uygulanabilen MIMO yöntemlerinin FBMC/OQAM sistemine direkt olarak uygulanmasını imkânsız hale getirmektedir. Diğer taraftan, [6]'da önerilen blok frekans yayılım stratejisi ile karmaşık diklik yeniden sağlanarak FBMC/OQAM sistemini OFDM sisteminde kullanılan MIMO metotlarının tamamı ile uyumlu hale getirmek mümkündür. Bu çalışmada, MIMO algılama metotlarının klasik MIMO-OFDM sistemine olduğu gibi direkt olarak MIMO-FBMC/OQAM sistemine de uygulanabilmesi için, FBMC/OQAM sisteminde blok frekans yayılım yaklaşımından faydalanılmıştır [6].

Blok frekans yayılım yaklaşımı kullanılan FBMC/OQAM vericisinde, yayma işlemi için $\mathbf{P} \in \mathbb{C}^{\frac{NT}{2} \times \frac{NT}{2}}$ ile ifade edilen bir ön kodlama matrisinden yararlanır. Söz konusu yayma işleminde, $\mathbf{x} \in \mathbb{C}^{\frac{NT}{2} \times 1}$ QAM sembol dizisi, ilgili ön kodlama matrisi ile aşağıdaki gibi çarpılarak $\mathbf{a} \in \mathbb{C}^{NT \times 1}$ gerçek değerli sembol dizisi elde edilir:

$$\mathbf{a} = \mathbf{P}\mathbf{x} \quad (10)$$

Alıcıda ise yayma işleminin tam tersi olan aşağıdaki işlem yerine getirilir:

$$\mathbf{y} = \mathbf{P}^H \mathbf{y} \quad (11)$$

Denklem (11)'i aşağıdaki gibi genişletilmiş formda da ifade etmek mümkündür:

$$\mathbf{y} = \mathbf{P}^H \mathbf{G}^H \mathbf{H} \mathbf{G} \mathbf{P} \mathbf{x} + \mathbf{P}^H \mathbf{G}^H \boldsymbol{\theta} \quad (12)$$

3.3. Sembol algılama probleminin formülasyonu

Blok frekans yayılım prosedürü uygulanmış tek antenli bir FBMC/OQAM sisteminde $\mathbf{x} \in \mathbb{C}^{\frac{NT}{2} \times 1}$ matrisi, $\frac{NT}{2}$ adet bileşene sahip QAM sembol dizisini temsil etmektedir. İlgili sistemde, QAM sembollerini A_T adet verici ve A_R adet alıcı anten üzerinden iletmek üzere MIMO anten konfigürasyonu kullanılması durumunda, $\mathbf{x} \in \mathbb{C}^{\frac{NT}{2} \times 1}$ sembol matrisi içerisindeki her bir element, $x_{n,t} = [x_{n,t}^{(1)}, x_{n,t}^{(2)}, \dots, x_{n,t}^{(A_T)}]^T$ şeklinde tanımlanan A_T uzunluklu sembol vektörü haline dönüşmektedir. Burada n ve t , sırasıyla alt taşıyıcı ve sembol indislerini belirtmektedir. Benzer şekilde, MIMO-FBMC/OQAM sisteminin A_R adet antene sahip alıcısında, $x_{n,t}$ sembol vektörünün alıcıya ulaşmış olan formu, $y_{n,t} = [y_{n,t}^{(1)}, y_{n,t}^{(2)}, \dots, y_{n,t}^{(A_R)}]^T$ şeklinde ifade edilmektedir. Buna göre, iletilen ve alınan sembol vektörlerinin içermiş oldukları sembol sayıları sırasıyla verici ve alıcı anten sayılarına eşit olacaktır. Bunun haricinde, $x_{n,t}$ vektöründe yer alan QAM sembollerinden herhangi birinin alabileceği karmaşık değer sayısı, QAM modülasyon seviyesi tarafından belirlenen yıldız kümesi noktalarının sayısına (yıldız kümesi boyutu) eşittir. MIMO-FBMC/OQAM sisteminde kullanılan QAM modülasyon tekniğinin yıldız kümesi boyutu Q değişkeni ile sembolize edilecek olursa, $x_{n,t}$ vektöründe yer alan her bir QAM sembolünün alabileceği değer sayısı Q' ya eşit olacaktır. A_T uzunluklu $x_{n,t}$ sembol vektörünün A_T adet QAM sembolü içerdiği göz önünde bulundurulduğunda, iletebilecek olası sembol dizilerinin sayısı Q^{A_T} olarak hesaplanacaktır. Geleneksel ML yönteminde, Q^{A_T} adet alternatif $x_{n,t}$ sembol dizisi arasından maksimum iletilme olasılığına sahip olanı bulmak amacıyla, söz konusu sembol dizilerinin her biri aşağıdaki eşitlik vasıtasıyla değerlendirilmektedir:

$$x_{n,t}^* = \arg \min_{x_{n,t}} \left\{ \left\| y_{n,t} - H_{n,t} \cdot x_{n,t} \right\|^2 \right\} \quad (13)$$

Yukarıdaki ifadede, olası $x_{n,t}$ sembol kombinasyonlarının her biri, ilgili $A_R \times A_T$ boyutlu $H_{n,t}$ kanal matrisi ile çarpılır. Bu çarpımlar sonucunda elde edilen vektörlerin her biri, alınan sinyale olan Öklid mesafesi bakımından test edilir. Son olarak, $H_{n,t}$ kanal matrisi ile çarpıldığında alınan sinyale minimum Öklid mesafesini veren $x_{n,t}$ vektörü, iletilme olasılığı en yüksek olan optimum sembol vektörü olarak belirlenir. Geleneksel ML dedektörü, $x_{n,t}^*$ ile sembolize edilen optimum sembol vektörünü bulabilmek için, olası her bir $x_{n,t}$ kombinasyonu başına bir Öklid hesabı gerçekleştirmek zorundadır. Buna göre, sembol algılama işlemi sonuçlanana kadar gerçekleştirilen toplam Öklid hesabı sayısı, olası sembol kombinasyonlarının toplam sayısına (Q^{A_T}) ulaşacaktır.

Denklem (13)'te yapılan varsayımın aksine, $H_{n,t}$ tarafından temsil edilen gerçek kanal katsayıları pratik uygulamalarda alıcı tarafından hatasız bir şekilde kestirilememektedir. Diğer bir ifade ile, gerçek uygulamalarda alıcı, Denklem (13)'teki gibi kusursuz bir kanal bilgisine sahip değildir. Bu nedenle bu çalışmada, kanal kestirim

hataları da göz önünde bulundurularak, ML tabanlı sembol algılama işlemi aşağıdaki gibi yeniden formüle edilmiştir:

$$x_{n,t}^* = \arg \min_{x_{n,t}} \left\{ \left\| y_{n,t} - \hat{H}_{n,t} \cdot x_{n,t} \right\|^2 \right\} \quad (14)$$

Burada $\hat{H}_{n,t}$, aşağıda tanımı verilen hatalı kanal katsayılarına karşılık gelmektedir [18].

$$\hat{H}_{n,t} = H_{n,t} + e \cdot \omega \quad (15)$$

Yukarıdaki eşitlikte, gerçek kanal katsayılarına eklenmiş olan $e \cdot \omega$ ifadesinde, e kestirim hatasını belirlerken, ω birim varyansa ve sıfır ortalamaya sahip karmaşık Gauss değişkene karşılık gelmektedir.

4. Ayrık Karga Arama Algoritması (DCSA)

Bilim dünyasına sunulmasının üzerinden henüz on yıl bile geçmemiş olmasına rağmen karga arama algoritması (CSA: Crow search algorithm) [10], çeşitli alanlardaki başarılı uygulamaları sayesinde popülaritesini önemli ölçüde artırmıştır. Ancak söz konusu algoritma başlangıçta sürekli optimizasyon problemleri için geliştirilmiş olup, ML dedektöründe QAM sembol vektörlerinin optimizasyonu gibi kombinatoriyal bir optimizasyon problemi doğrudan ayrık uzayda çözmeye uygun değildir. Bu nedenle bu çalışmada, orijinal CSA algoritmasının döngüsel bit çevirme tabanlı oldukça dinamik bir araştırma mekanizmasına sahip DCSA [9] isimli etkili bir ayrık versiyonundan faydalanılmıştır. DCSA algoritmasında, D -boyutlu ayrık çözüm uzayındaki karga pozisyonları aşağıdaki gibi tanımlanmaktadır:

$$w_z^{(d)} = \left[w_z^{(1)}, w_z^{(2)}, \dots, w_z^{(D)} \right] \quad , \quad z = 1, 2, \dots, Z \quad (16)$$

Burada z karga indisini ifade ederken, Z sürüde yer alan toplam karga sayısını belirtmektedir. $w_z^{(d)} \in \{0, 1\}$ olup $w_z^{(d)}$ vektörünün her bir elemanı ya "0" ya da "1" değeri almaktadır. Her bir karganın güncel en iyi yiyecek pozisyonunu sakladığı hafıza vektörü ise aşağıdaki gibi tanımlanmaktadır:

$$m_z^{(d)} = \left[m_z^{(1)}, m_z^{(2)}, \dots, m_z^{(D)} \right] \quad , \quad z = 1, 2, \dots, Z \quad (17)$$

Mevcut karga popülasyonunda pozisyon güncellemesi gerçekleştirilirken şu şekilde bir yol izlenmektedir: Öncelikle sürüde yer alan Z adet kargadan her biri için $r_z \in [0, 1]$ olacak şekilde rastgele bir sayı üretilir. Buradaki r_z değişkeni, z 'inci karga için üretilen rastsal sayıya karşılık gelmektedir. Ardından ilgili karga için üretilmiş olan r_z sayısı, AP şeklinde sembolize edilen farkındalık olasılığı parametresi ile karşılaştırılır. Optimizasyon işlemine başlamadan önce belirlenmesi gereken AP parametresinin değer aralığının $[0, 1]$ olduğu unutulmamalıdır. Bu noktadan itibaren ortaya çıkması muhtemel iki farklı durum mevcuttur:

Durum 1:

$r_z \geq AP$ olması durumunda, döngüsel bit çevirme mekanizması [19] kullanılarak z 'inci karganın ayrık uzaydaki pozisyonu güncellenir. Bunun için öncelikle ilgili karga pozisyonu aşağıdaki gibi bit çevirme işlemine tabi tutulur:

$$w_z^{(d), yeni} = flip(w_z^{(d)})_{c_z} \quad (18)$$

Burada $w_z^{(d)}$ vektörünün komşuluğunda yeni bir pozisyon üretmek için, ilgili vektörün c_z 'inci boyutunda yer alan elemana $flip(\cdot)_{c_z}$ operatörü kullanılarak bit çevirme işlemi uygulanmaktadır. Pozisyon vektörlerinin ikili bit dizilerinden oluşması sebebiyle, $w_z^{(d)}$ vektörünün c_z 'inci boyutunda yer alan elemanın bit çevirme işlemine tabi tutulması demek, ilgili elemanın 0'dan 1'e ya da 1'den 0'a dönüştürülmesi anlamına gelmektedir. Burada c_z , z 'inci karganın bit çevirme indisini belirtmektedir. Söz konusu indisin başlangıç değeri, her bir karga için $c_z = 1$ olacak şekilde ayarlanır. Bit çevirme işlemi gerçekleştirildikten sonra yeni çözüm için uygunluk hesabı yapılır. Eğer yeni üretilen $w_z^{(d), yeni}$ komşu çözüm, $w_z^{(d)}$ çözümüne göre daha iyi bir uygunluk değerine sahipse, $w_z^{(d)}$ çözümünün yerini alır. Ters durumda, $w_z^{(d)}$ çözümünde herhangi bir güncelleme yapılmaz. Bunun yanı sıra, söz konusu yeni

çözüm, z' inci karganın $m_z^{(d)}$ hafıza vektöründe saklanan güncel en iyi çözümü ile uygunluk değeri bakımından karşılaştırılır. $w_z^{(d),yeni}$ çözümünün uygunluk değeri, $m_z^{(d)}$ hafıza vektöründe kayıtlı olan çözümün uygunluk değerinden daha iyi ise, $w_z^{(d),yeni}$ çözümü güncel en iyi çözüm olarak $m_z^{(d)}$ vektörüne kaydedilip z' inci karga için hafıza güncellemesi gerçekleştirilir. Ters durumda ilgili karganın hafızasında herhangi bir değişiklik gerçekleştirilmez. Söz konusu hafıza ve pozisyon güncellemeleri tamamlandıktan sonra, z' inci kargaya ait bit çevirme indisinin değeri aşağıdaki gibi 1 artırılarak gelecek iterasyonlarda $r_z \geq AP$ koşulunun z' inci karga için yeniden sağlanması durumunda, bit çevirme işleminin aynı eleman üzerinde gerçekleştirilmeyip bir sonraki vektör elemanından devam etmesi garanti altına alınır:

$$c_z = c_z + 1 \quad (19)$$

Optimizasyon işlemi boyunca bit çevirme işleminin her bir karga için döngüsel bir şekilde sürmesini sağlamak amacıyla, Denklem (19)'da gerçekleştirilen güncellemenin ardından c_z indisi, aşağıdaki kontrol mekanizmasından geçirilir:

$$c_z = \text{mod}(c_z - 1, D) + 1 \quad (20)$$

Burada $\text{mod}(c_z - 1, D)$ operatörü, $(c_z - 1)$ 'in D 'ye bölümünden kalanı vermektedir. Örneğin, z' inci karga için $r_z \geq AP$ şartının sağlandığı her iterasyonda, ilgili karganın bit çevirme indisi Denklem (19)'da gösterildiği gibi 1 artırılır. Bu nedenle, iterasyonlar ilerledikçe, c_z indisinin $D + 1$ değerine ulaşarak toplam vektör boyutu sayısını (vektör uzunluğunu) aşması kaçınılmaz hale gelecektir. Denklem (20)'de verilen kontrol mekanizması sayesinde, c_z indisinin D 'den daha büyük bir değer almasına izin verilmez. c_z indisi $D + 1$ değerine her ulaştığında, ilgili kontrol mekanizması ile hemen başlangıç noktasına yani 1 değerine döndürülür. DCSA algoritmasında komşu çözüm üretimi için döngüsel bit çevirme prosedürünün kullanılması, ayrık çözüm uzayının oldukça yoğun ve sistematik bir şekilde araştırılmasını mümkün kılmaktadır.

Durum 2:

$r_z < AP$ ise, z' inci karga ayrık uzayda rastgele bir pozisyona kaydırılır. Bunun için ilgili karganın pozisyon vektörü aşağıdaki gibi mutasyon işlemine tabi tutulur:

$$w_z^{(d),mutant} = \text{mut}_{1 \leq d \leq D} (w_z^{(d)}, NB) \quad (21)$$

Yukarıda verilen mutasyon operatörü vasıtasıyla, $w_z^{(d)}$ pozisyon vektörünün $1 \leq d \leq D$ aralığındaki boyutlarında yer alan elemanlarından rastgele seçilen NB sayıda bit mutasyona uğratılır ("0" ise "1", "1" ise "0" yapılır). Bu şekilde kontrollü olarak gerçekleştirilen mutasyon işlemi sayesinde, DCSA algoritmasının yerel minimumlara takılma olasılığı en aza indirilmektedir.

5. Ayrık Karga Arama Algoritması Tabanlı ML Yöntemi

Bir tür ikili optimizasyon algoritması olması sebebiyle, DCSA algoritmasının karmaşık QAM sembollerinin optimizasyonu için ML dedektörüne doğrudan entegrasyonu mümkün değildir. Diğer taraftan, bu gibi ayrık optimizasyon algoritmalarını sembol algılama probleminde etkili bir şekilde kullanmanın kolay bir yolu vardır. Optimize edilecek sembol vektörleri karmaşık değerli QAM sembollerinden oluşsa bile, $x_{n,t} = [x_{n,t}^{(1)}, x_{n,t}^{(2)}, \dots, x_{n,t}^{(A_T)}]^T$ şeklinde ifade edilen ilgili vektörleri direkt olarak ikili arama uzayında optimize etmek mümkündür. $x_{n,t}$ sembol vektörünü ikili optimizasyona uygun hale getirmek için yapılması gereken tek şey, optimizasyon işlemi boyunca söz konusu vektörün ikili eşdeğerini kullanmaktır. Diğer bir ifade ile, QAM sembol vektörlerini ikili arama uzayında DCSA algoritması tarafından optimize edilmeye uygun hale getirmek için ikili bit dizileri olarak temsil edebiliriz. $x_{n,t}$ vektöründeki her bir QAM sembolü tarafından taşınan bit sayısı $\log_2^Q = k$ olarak hesaplandığından dolayı, sembol vektörlerinin ikili eşdeğerleri $(k \cdot A_T)$ uzunluğunda olacaktır. Örneğin, modülasyon türü olarak 4-QAM kullanıldığı takdirde, her bir karmaşık değerli sembol $\log_2^4 = 2$ adet ikili bit taşıyacaktır. Bu durumda, her bir sembol vektöründeki toplam bit sayısı $(2 \cdot A_T)$ değerine eşit olacaktır. Bu çalışmada önerilen DCSA-ML isimli sembol algılama stratejisinde, karmaşık değerli QAM sembol dizileri ve bu dizilerin ikili bit dizisi versiyonları karga pozisyonları olarak aşağıdaki gibi tanımlanmaktadır:

$$x_z^{(i)} = [x_z^{(1)}, x_z^{(2)}, \dots, x_z^{(A_T)}] \quad , \quad z = 1, 2, \dots, Z \quad (22)$$

$$b_z^{(j)} = [b_z^{(1)}, b_z^{(2)}, \dots, b_z^{(k \cdot A_T)}] \quad , \quad z = 1, 2, \dots, Z \quad (23)$$

Burada $b_z^{(j)} \in \{0,1\}$ 'dir. DCSA-ML stratejisi aşağıda adım adım anlatılmıştır:

Adım 1: İlk olarak, başlangıç karga popülasyonu oluşturulur. Bunun için Denklem (23)'te tanımlanan ikili bit dizisi formunda Z adet rastgele pozisyonlarda çözüm vektörleri üretilir. Rasgele bir şekilde üretilmiş olan söz konusu başlangıç pozisyonları, aynı zamanda hafıza vektörlerine de kaydedilir. Aşağıda z 'inci karganın hafıza vektörü tanımlanmıştır:

$$m_z^{(j)} = [m_z^{(1)}, m_z^{(2)}, \dots, m_z^{(k \cdot A_T)}] \quad , \quad z = 1, 2, \dots, Z \quad (24)$$

Adım 2: Başlangıç popülasyon üyelerinin rastgele üretilmesinin ardından, her birinin uygunluk değeri hesaplanır. $b_z^{(j)}$ şeklinde ikili QAM sembol vektörü olarak tanımlanan herhangi bir popülasyon üyesinin uygunluk değerini hesaplamadan önce, ilgili sembol vektörünün $x_z^{(i)}$ şeklinde tanımlanan karmaşık eşdeğerine dönüştürülmesi gerekmektedir. Söz konusu dönüşüm gerçekleştirildikten sonra, ilgili popülasyon üyesini aşağıda verilen uygunluk fonksiyonunda yerine yazıp uygunluk değerini hesaplayabiliriz:

$$f_z = \text{fit}(x_z^{(i)}) = \|y_{n,t} - \hat{H}_{n,t} \cdot x_z^{(i)}\|^2 \quad , \quad z = 1, 2, \dots, Z ; \quad i = 1, 2, \dots, A_T \quad (25)$$

Burada f_z , karga sürüsündeki z 'inci popülasyon üyesinin uygunluk değerine karşılık gelmektedir.

Adım 3: Popülasyondaki her bir karga için $r_z \in [0,1]$ şeklinde tanımlanan rastgele bir sayı üretilir ve üretilen bu sayı daha sonra AP parametresi ile karşılaştırılır.

Adım 4: $r_z \geq AP$ olması halinde, z 'inci kargadan bir adet komşu çözüm üretmek için, ilgili kargaya ait pozisyon vektörüne aşağıdaki gibi bit çevirme işlemi uygulanır:

$$b_z^{(j),\text{yeni}} = \text{flip}(b_z^{(j)})_{c_z} \quad (26)$$

Üretilen $b_z^{(j),\text{yeni}}$ çözüm vektörü, ikili bit dizisi halinden karmaşık değerli formuna karşılık gelen $x_z^{(i),\text{yeni}}$ vektörüne dönüştürüldükten sonra, aşağıdaki fonksiyon vasıtasıyla uygunluk değeri hesaplanır:

$$f_z^{\text{yeni}} = \text{fit}(x_z^{(i),\text{yeni}}) = \|y_{n,t} - \hat{H}_{n,t} \cdot x_z^{(i),\text{yeni}}\|^2 \quad (27)$$

Burada f_z^{yeni} , z 'inci kargadan üretilmiş olan yeni çözümün uygunluk değerini ifade etmektedir. Uygunluk hesabının ardından, sırasıyla hem $b_z^{(j)}$ ve $b_z^{(j),\text{yeni}}$ arasında, hem de $m_z^{(j)}$ ve $b_z^{(j),\text{yeni}}$ arasında uygunluk kalitesi bakımından aç gözlü seçim işlemi gerçekleştirilir. Daha sonra, bit çevirme işleminin optimizasyon işlemi süresince her bir popülasyon üyesi için döngüsel bir şekilde ilerlemesini sağlamak amacıyla, bit çevirme indisi olarak bilinen c_z parametresi aşağıdaki işlemlerden geçirilir:

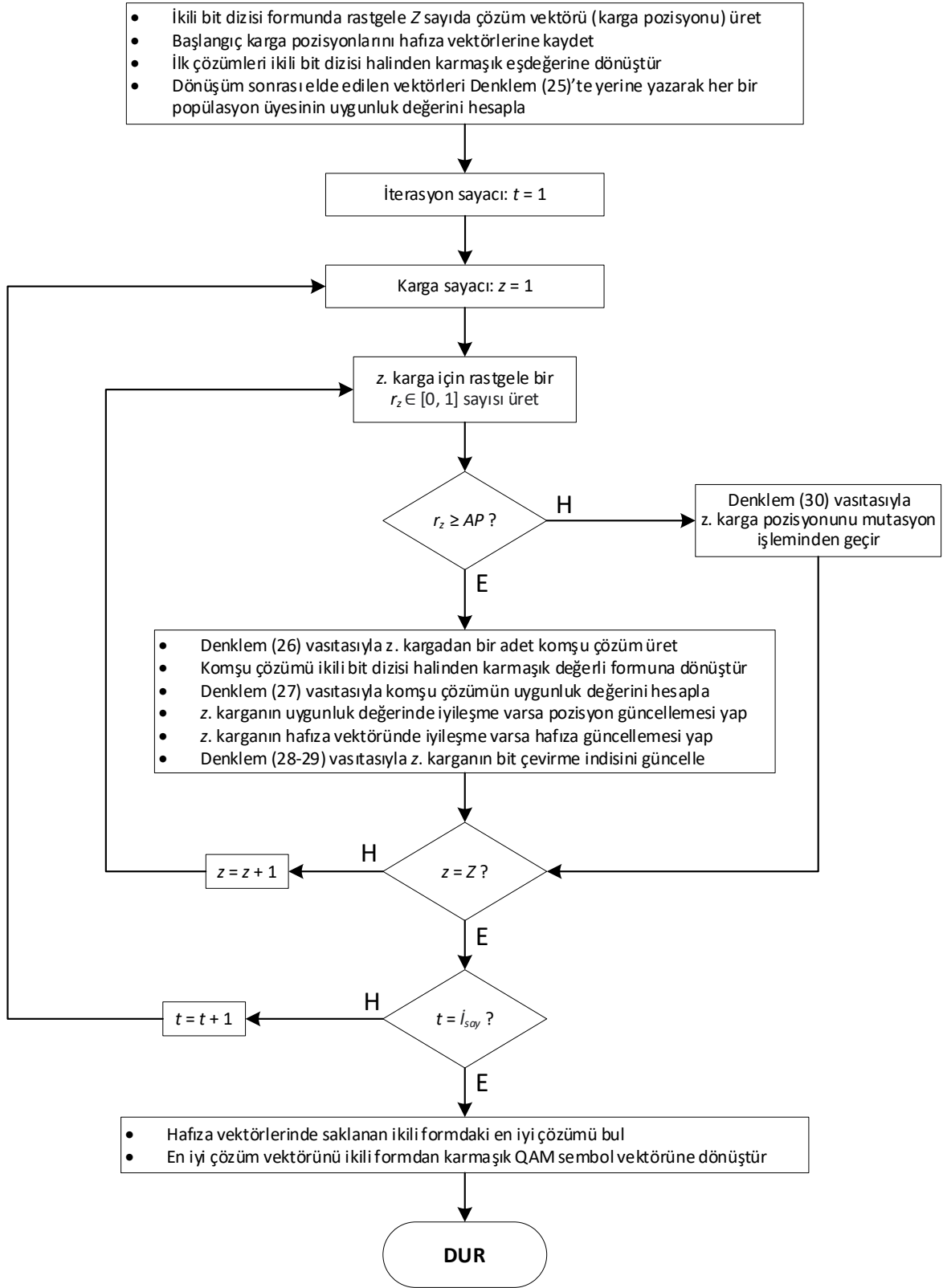
$$c_z = c_z + 1 \quad (28)$$

$$c_z = \text{mod}(c_z - 1, k \cdot A_T) + 1 \quad (29)$$

Adım 5: $r_z < AP$ ise, z 'inci karga pozisyonuna aşağıdaki mutasyon operatörü uygulanır:

$$b_z^{(j),\text{mutant}} = \text{mut}_{1 \leq j \leq k \cdot A_T}(b_z^{(j)}, NB) \quad (30)$$

Denklem (30)'da verilen mutasyon operatörü, $1 \leq j \leq k \cdot A_T$ aralığında rastgele belirlenen NB adet ikili biti 0'dan 1'e ya da 1'den sıfıra dönüştürmektedir.



Şekil 1. DCSA-ML prosedürünün akış şeması.

Adım 6: Adım 3 – Adım 5 arası işlemler her bir popülasyon üyesi için tekrarlanarak ilk iterasyon tamamlanır. İlgili adımlar arasındaki işlemlerin Z kere tekrar edilmesinin, DCSA-ML yönteminin tek bir iterasyonuna karşılık geldiği hesaba katılarak, önceden belirlenmiş olan maksimum iterasyon sayısına (t_{say}) ulaşıncaya kadar optimizasyon işlemine devam edilir. Durdurma kriteri sağlandıktan sonra, hafıza vektörlerinde saklanan en düşük uygunluk değerine sahip karga pozisyonu, ikili formdaki optimum sembol vektörü olarak tayin edilir. Sembol algılama işlemini tamamlamak için, ikili bitlerden oluşan söz konusu sembol vektörü, eşdeğeri olan karmaşık QAM sembol dizisine dönüştürülür. Şekil 1’de DCSA-ML prosedürünün akış diyagramı verilmiştir.

6. Simülasyon Sonuçları

Bu bölümde, önerilen DCSA-ML ve ele alınan diğer sembol algılama stratejilerinin MIMO-FBMC/OQAM alıcısında QAM sembol dizilerini elde etmedeki başarıları test edilmiştir. Bu amaçla, söz konusu yöntemlerin 4×4 , 6×6 ve 8×8 MIMO konfigürasyonlarındaki bit hata oranı başarımları karşılaştırılmıştır. Bunun yanı sıra, ilgili sembol algılama stratejilerinin optimum çözüme yakınsama yetenekleri, her birinin yakınsama eğrisi elde edilerek birbirleriyle kıyaslanmıştır. Simülasyonlarda kullanılan sistem parametreleri Tablo 1’de verilmiştir.

Tablo 1. Simülasyonlar için belirlenen parametre değerleri.

Filtre türü	PHYDYAS
FBMC sembollerinin sayısı	$T = 30$
QAM yıldız kümesi boyutu	$Q = 4$
Alt taşıyıcıların frekans aralığı	15 KHz
Örtüşme katsayısı	4
Anten yapısı	4×4 , 6×6 , 8×8
Taşıyıcı frekansı	2.5 GHz
Kanal türü	SUI-6
Alt taşıyıcı sayısı	$N = 64$
Kestirim hatası oranı	$e = 25\%$

Tablo 2’de, ML tabanlı stratejilerin araştırma karmaşıklığı (AK) üzerine bir analiz gerçekleştirilmiştir. Geleneksel ML yöntemi için araştırma karmaşıklığı tanımı, sembol algılama işlemi boyunca gerçekleştirilen Öklid hesabı sayısı dikkate alınarak yapılmaktadır. Metasezgisel yaklaşımlara dayanan modifiye edilmiş ML varyantlarının araştırma karmaşıklıklarında ise, her biri tek bir Öklid hesabına karşılık gelen uygunluk değerlendirmelerinin sayısı hesaba katılmaktadır. Buna göre, söz konusu ML varyantlarının araştırma karmaşıklığını tanımlamak için tek yapılması gereken, sembol optimizasyonu süresince gerçekleştirilen uygunluk değerlendirmelerinin toplam sayısını belirlemektir. Bu bağlamda, DCSA-ML stratejisinin optimizasyon periyodu boyunca kaç adet uygunluk değerlendirmesi gerçekleştirdiğini elde ederek ilgili yöntemin araştırma karmaşıklığını tanımlayabiliriz. DCSA-ML yönteminin her bir iterasyonunda, mevcut popülasyonda yer alan her bir karga için bir adet uygunluk değerlendirmesi gerçekleştirilmektedir. Buna göre, toplam karga sayısını maksimum iterasyon sayısı ile çarpacak olursak, DCSA-ML stratejisinin araştırma karmaşıklığını $AK = Z \cdot I_{\text{say}}$ şeklinde elde ederiz. disABC-ML yönteminin araştırma karmaşıklığı, aynı yol izlenerek $AK = R \cdot D_{\text{say}}$ şeklinde elde edilir. DBHS-ML yönteminde, her bir araştırmada tek bir uygunluk hesabı gerçekleştirildiğinden dolayı, ilgili yöntemin araştırma karmaşıklığı, Tablo 2’de gösterildiği gibi direk olarak optimizasyon süresi boyunca gerçekleştirilen toplam araştırma sayısına eşit olacaktır.

Sinyal iletimi için daha yüksek anten konfigürasyonu kullanımı, optimum sonuca yakınsayabilmek için çok daha yoğun bir şekilde taranması gereken daha geniş bir arama uzayına yol açmaktadır. Bu nedenle, daha yüksek bir MIMO konfigürasyonu, metasezgisel algoritmalara dayalı ML stratejilerinin her biri için hem daha fazla sayıda araştırma sayısı hem de fazladan popülasyon üyeleri gerektirmektedir. Bu gereksinimi karşılamak için, DCSA-ML yönteminin 4×4 , 6×6 ve 8×8 anten konfigürasyonlarındaki araştırma karmaşıklıkları sırasıyla 150, 375 ve 860 olarak belirlenirken, DBHS-ML ve disABC-ML yöntemlerinin aynı MIMO yapıları için araştırma karmaşıklıkları 220, 750 ve 1800 değerlerine eşitlenmiştir. Aynı şekilde, 4×4 ’ten 8×8 ’e kadar artırılan söz konusu anten yapıları için, DCSA-ML prosedüründeki karga popülasyonunun büyüklüğü 10, 15 ve 20 olarak ayarlanırken, DBHS-ML ve disABC-ML yöntemlerindeki popülasyon boyutları sırasıyla 20, 30 ve 40 şeklinde belirlenmiştir. Araştırma karmaşıklığı ile bit hata oranı başarımları arasında bir ödünleşim olduğu unutulmamalıdır. Önerilen DCSA-ML stratejisi için en uygun araştırma karmaşıklığı değerlerini belirlerken, bu ödünleşim dikkate alınmıştır. Örneğin, MIMO-FBMC/OQAM sisteminde tatmin edici BER sonuçlarına ulaşmak için DCSA-ML dedektörü tarafından ihtiyaç duyulan uygunluk hesabı sayıları, söz konusu ödünleşim göz önünde bulundurularak sırasıyla 150, 375 ve 860 olarak belirlenmiştir. Fazladan uygunluk değerlendirmeleri, DCSA-ML yönteminin BER performansında kayda değer bir iyileşme sağlamadığından dolayı, 150, 375 ve 860 olarak belirlenen söz konusu uygunluk değerlendirme sayılarındaki herhangi bir artış, önerilen yöntemin araştırma karmaşıklığında gereksiz bir büyümeye yol açacaktır. Metasezgisel tabanlı ML dedektörlerinde, uygunluk hesabı sayısının direkt olarak araştırma karmaşıklığını belirlediği unutulmamalıdır. Diğer taraftan DBHS-ML ve disABC-ML yöntemlerinin, optimizasyon süreci boyunca çok daha fazla sayıda uygunluk hesabı yapmasına müsaade edilerek çözüm uzayını daha çok araştırmalarına fırsat tanınmıştır. Rakiplerine göre dezavantajlı bir pozisyona düşürülmesine rağmen, önerilen DCSA-ML stratejisi, her

bir anten konfigürasyonu için optimuma en yakın BER seviyesine ulaşarak DBHS-ML ve disABC-ML dedektörlerini geride bırakmayı başarmıştır. Bu çalışmada ele alınan zeki optimizasyona dayalı ML dedektörlerinin diğer parametre değerleri Tablo 3'te gösterilmektedir.

Tablo 2. Makalede ele alınan ML tabanlı sembol dedektörleri için araştırma karmaşıklığı analizi.

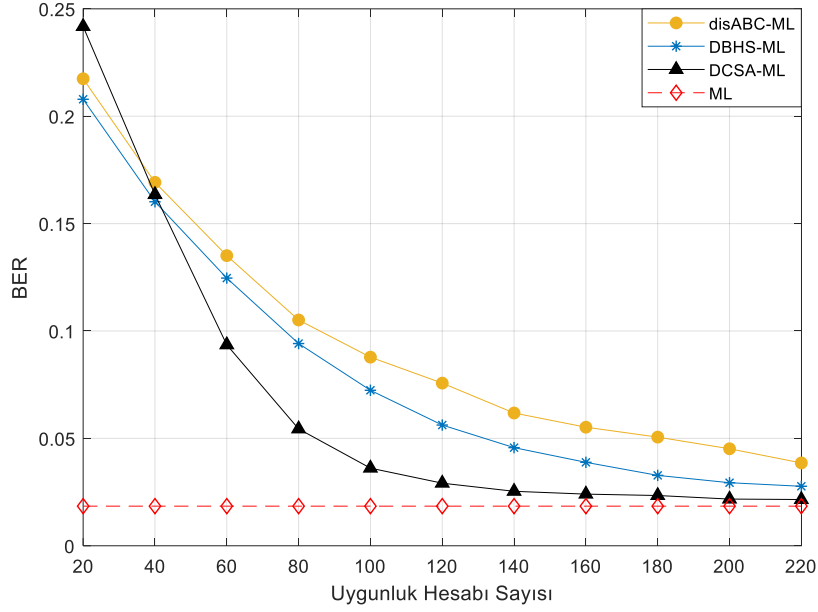
Sembol Algılama Prosedürleri		4×4	6×6	8×8
disABC-ML	Koloni boyutu	$R = 20$	$R = 30$	$R = 40$
	Döngü sayısı	$D_{say} = 11$	$D_{say} = 25$	$D_{say} = 45$
	Araştırma karmaşıklığı	$AK = R \cdot D_{say} = 220$	$AK = R \cdot D_{say} = 750$	$AK = R \cdot D_{say} = 1800$
	12 dB SNR için BER	0.03857	0.02942	0.01783
DBHS-ML	Popülasyon boyutu	$Pop = 20$	$Pop = 30$	$Pop = 40$
	Araştırma sayısı	$A_{say} = 220$	$A_{say} = 750$	$A_{say} = 1800$
	Araştırma karmaşıklığı	$AK = A_{say} = 220$	$AK = A_{say} = 750$	$AK = A_{say} = 1800$
	12 dB SNR için BER	0.02770	0.02109	0.01123
DCSA-ML	Karga sayısı	$Z = 10$	$Z = 15$	$Z = 20$
	İterasyon sayısı	$I_{say} = 15$	$I_{say} = 25$	$I_{say} = 43$
	Araştırma karmaşıklığı	$AK = Z \cdot I_{say} = 150$	$AK = Z \cdot I_{say} = 375$	$AK = Z \cdot I_{say} = 860$
	12 dB SNR için BER	0.02471	0.02049	0.006142
ML	Araştırma karmaşıklığı	$AK = Q^{Ar} = 4^4 = 256$	$AK = Q^{Ar} = 4^6 = 4096$	$AK = Q^{Ar} = 4^8 = 65536$
	12 dB SNR için BER	0.01845	0.01118	0.002702

Tablo 3. Zeki optimizasyon tabanlı sembol algılama stratejilerine ait parametre değerleri.

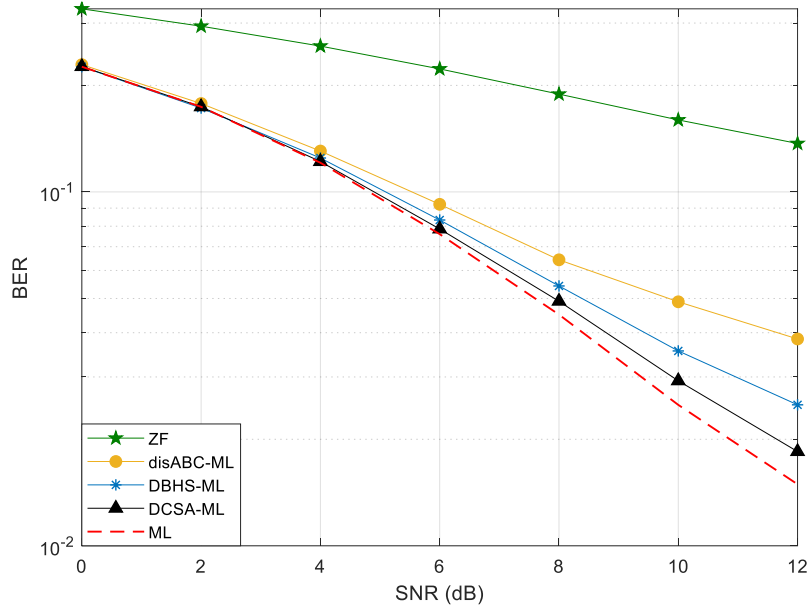
Strateji	Parametre	Değer
DCSA-ML	Farkındalık olasılığı	$AP = 0.05$
	Mutasyona uğratılacak bit sayısı	$NB = 3$
DBHS-ML	Ton ayarlama oranı	$PAR = 0.05$
	Harmoni hafızasını kullanma oranı	$HMCR = 0.6$
disABC-ML	Maksimum deneme sayısı	30

Şekil 2'de, çalışmada ele alınan zeki optimizasyona dayalı modifiye edilmiş ML stratejilerinin yakınsama kabiliyetleri, 4×4 anten konfigürasyonlu MIMO-FBMC/OQAM sisteminde teste tabi tutulmuştur. Söz konusu yakınsama analizinde, sistemin SNR değeri 12 dB'ye ayarlanmıştır. Şekilden de açıkça görüleceği üzere, DBHS-ML ve disABC-ML yöntemlerinin optimum BER seviyesine yaklaşımları için 220 adet uygunluk hesabı bile yeterli olmazken, önerilen DCSA-ML stratejisinin yakınsama eğrisi sadece 150 uygunluk değerlendirmesi sonucu neredeyse bu seviyeye örtüşmektedir. Örneğin DBHS-ML ve disABC-ML dedektörleri, sırasıyla 0.02770 ve 0.03857 BER değerlerine ulaşabilmek için 220 adet uygunluk değerlendirmesine ihtiyaç duymaktadırlar. Öte yandan, üst düzey yakınsama kabiliyeti sayesinde, DCSA-ML dedektörü aynı BER seviyelerine sadece 128 ve 97 uygunluk değerlendirme sayıları ile ulaşabilmektedir. Bunun haricinde, önerilen strateji ile 220 uygunluk hesabı boyunca optimizasyon işlemine devam etmenin gereksiz olduğu Şekil 2'den de açıkça görülmektedir. Bu durum göz önüne alınarak, DCSA-ML sembol dedektörünün maksimum uygunluk hesabı sayısı 150'ye ayarlanmıştır.

Şekil 3'te, önerilen DCSA-ML stratejisi, ZF ve ML gibi geleneksel yöntemler de dahil olmak üzere çalışmada ele alınan diğer sembol dedektörlerinin her biri ile 4×4 MIMO-FBMC/OQAM sistemindeki BER performansı bakımından karşılaştırılmıştır. Şekilden de kolaylıkla anlaşılacağı üzere DCSA-ML yöntemi, bit hata oranı seviyesi olarak alt sınıra en çok yaklaşan yöntem olmuştur. En yakın takipçisi olarak göze çarpan DBHS-ML dedektörüne sembol algılama işlemi sırasında 70 adet daha fazla uygunluk hesabı gerçekleştirme fırsatı verilmesine rağmen, önerilen DCSA-ML prosedürü söz konusu metasezgisel tabanlı rakibine göre daha düşük BER değerlerine ulaşmıştır.

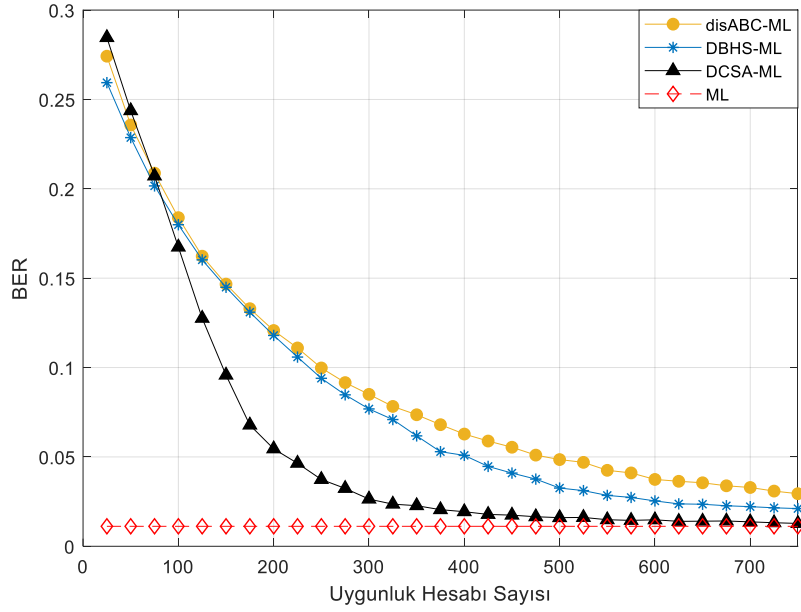


Şekil 2. Zeki optimizasyon tabanlı ML stratejilerinin 4x4 anten konfigürasyonu için yakınsaklık analizi.



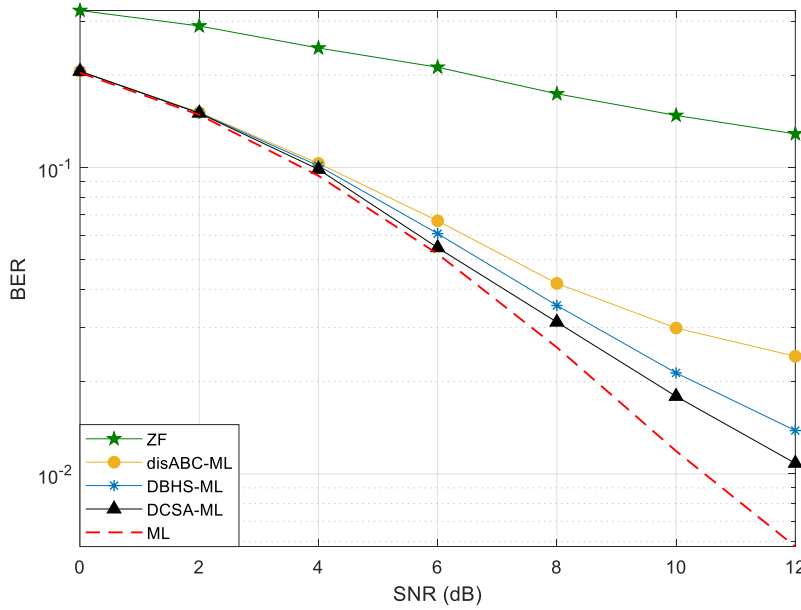
Şekil 3. Sembol algılama stratejilerinin 4x4 anten konfigürasyonu için bit hata oranı başarımları.

Sembol algılama stratejilerinin daha geniş bir araştırma uzayındaki yakınsama davranışlarını incelemek için, MIMO-FBMC/OQAM sisteminin anten konfigürasyonu 4x4'ten 6x6'ya yükseltilmiştir. Anten sayısındaki sözcük artışı ardından, metasezgisel algoritmaya dayanan her bir sembol dedektörünün yakınsama eğrisi Şekil 4'te elde edilmiştir. Daha büyük anten konfigürasyonlarında daha yüksek sayıda uygunluk değerlendirmesine duyulan ihtiyaç, Şekil 2 ve Şekil 4'te sırasıyla 4x4 ve 6x6 anten konfigürasyonları için elde edilen yakınsama eğrilerinden rahatlıkla anlaşılabilir. Bunun haricinde, Şekil 4'ten de açıkça görüleceği üzere, önerilen DCSA-ML stratejisi oldukça başarılı bir yakınsama performansı göstermektedir. Önerilen yöntemin DBHS-ML ve disABC-ML dedektörlerini tam olarak geride bırakması için sadece 84 adet uygunluk hesabı yeterli olmuştur. Bu noktadan itibaren, optimum çözüme hızla yakınsamaya devam ederek, diğer yöntemler ile arasındaki farkı daha da açmıştır. Şekil 4'ten de açıkça görüleceği üzere, 375. uygunluk değerlendirmesinden itibaren, DCSA-ML yönteminin yakınsama eğrisinde önemli bir gelişim gözlenmemektedir. Bu yüzden, gereksiz işlem yükünden kaçınmak amacıyla, DCSA-ML dedektörünün maksimum uygunluk hesabı sayısı, 6x6 MIMO yapısı için 375 olarak tayin edilmiştir.



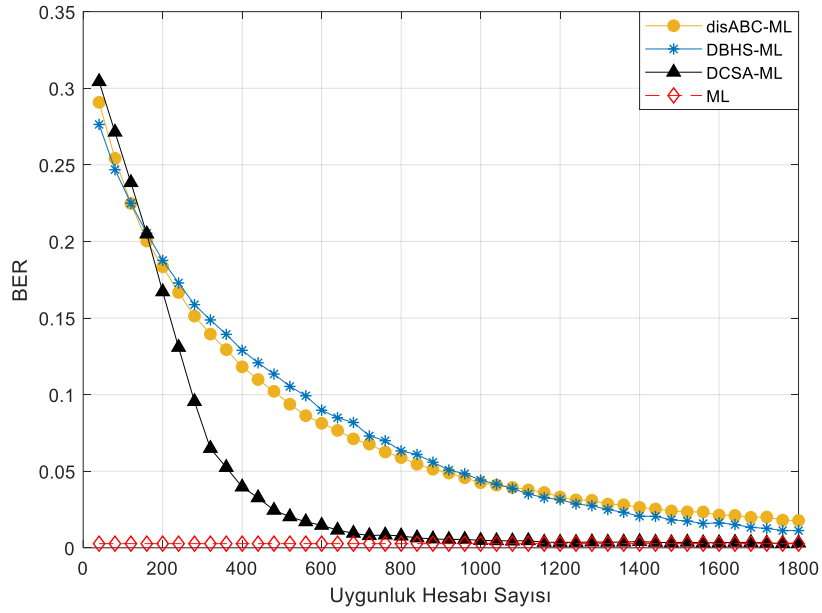
Şekil 4. Zeki optimizasyon tabanlı ML stratejilerinin 6x6 anten konfigürasyonu için yakınsaklık analizi.

Şekil 5'te, 4x4'ten 6x6'ya anten artışının DCSA-ML ve diğer karşılaştırma yöntemleri üzerindeki etkisini incelemek amacıyla, 6x6 MIMO-FBMC/OQAM sistemi için her bir sembol algılama stratejisinin bit hata oranı eğrisi elde edilmiştir. Verici ve alıcı anten sayılarındaki artış, önerilen DCSA-ML stratejisinin ZF, DBHS-ML ve disABC-ML dedektörlerine oranla optimuma daha yakın BER değerleri elde ederek üstünlüğünü korumasına engel olamamıştır. Örneğin, 10 dB SNR değerindeki BER sonuçları dikkate alınacak olursa, geleneksel ML yöntemine ait optimum BER seviyesi 1.19×10^{-2} değerine eşit olurken, önerilen DCSA-ML stratejisi 1.79×10^{-2} BER değerine ulaşarak optimuma yakın bir sonuç elde etmiştir. İlgili SNR değerinde DBHS-ML ve disABC-ML yöntemleri, sırasıyla 2.13×10^{-2} ve 3×10^{-2} seviyelerine ancak ulaşabilmişlerdir.



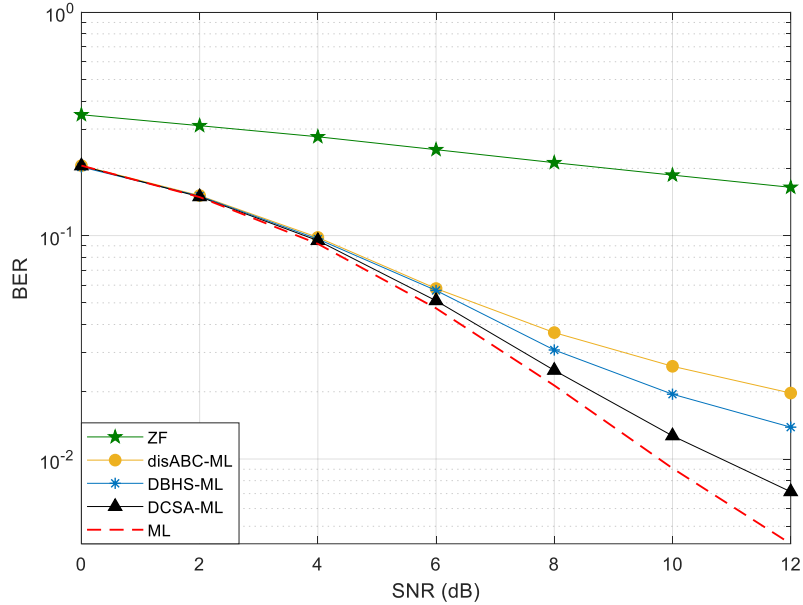
Şekil 5. Sembol algılama stratejilerinin 6x6 anten konfigürasyonu için bit hata oranı başarımları.

Şekil 6'da zeki optimizasyon tabanlı gelişmiş ML dedektörleri, 8x8 gibi daha da yükseltilmiş bir anten konfigürasyonuna sahip MIMO-FBMC/OQAM sistemindeki yakınsama hızlarına göre karşılaştırılmıştır. Şekilde görüldüğü gibi, önerilen DCSA-ML stratejisi üstün yakınsama kabiliyetini 8x8 anten konfigürasyonunda da göstermiştir. Önerilen sembol algılama yöntemi, DBHS-ML ve disABC-ML dedektörlerini yaklaşık olarak 170. uygunluk hesabında geçtikten sonra optimum çözüme doğru hız kesmeden yakınsamaya devam ederken diğer yöntemler, yetersiz yakınsama performanslarından dolayı optimum bit hata oranı seviyesinden oldukça uzakta kalmışlardır. DCSA-ML dedektörünün optimuma yakın bir çözüme ulaşması için yaklaşık 860 adet uygunluk hesabı yeterli gelmiştir, bu nedenle 860'dan 1800'e kadar 940 adet gereksiz uygunluk hesabından kaçınmak için, maksimum uygunluk değerlendirme sayısı 860 olarak belirlenmiştir.



Şekil 6. Zeki optimizasyon tabanlı ML stratejilerinin 8x8 anten konfigürasyonu için yakınsaklık analizi.

Anten konfigürasyonunu 6x6'dan 8x8'e yükselterek çalışmada ele alınan sembol algılama stratejileri için işleri daha da zorlaştırdıktan sonra, ilgili stratejilerin 8x8 MIMO-FBMC/OQAM sistemindeki bit hata oranı başarımları Şekil 7'de karşılaştırılmıştır. Söz konusu anten artışından dolayı araştırma uzayına meydana gelen önemli genişlemeye rağmen, önerilen DCSA-ML stratejisi, sahip olduğu özel yakınsama yeteneği sayesinde bit hata oranı performansı bakımından DBHS-ML ve disABC-ML metotları ile arasındaki mesafeyi daha da açmıştır. Öte yandan, diğer yöntemlerin yakınsama hızları, simülasyonlarda 940 adet daha fazla uygunluk hesabı yapmalarına izin verilmesine rağmen DCSA-ML dedektörünün BER sonuçlarına yaklaşabilmeleri için yeterli olmamıştır.



Şekil 7. Sembol algılama stratejilerinin 8x8 anten konfigürasyonu için bit hata oranı başarımları.

7. Sonuç

Bu çalışmada, karga arama algoritmasının oldukça nitelikli bir ayrık varyantı (DCSA) sembol optimizatörü olarak kullanılmak üzere klasik ML yöntemine entegre edilerek MIMO-FBMC/OQAM sistemi için DCSA-ML isimli yeni ve modern bir sembol algılama stratejisi geliştirilmiştir. Sonrasında bu strateji, MIMO-FBMC/OQAM sistemindeki bit hata oranı başarımları bakımından çeşitli anten konfigürasyonları için teste tabi tutulmuştur. Önerilen DCSA-ML stratejisinin ilgili sistemdeki bit hata oranı başarımlarını ölçerek sembol algılama performansını değerlendirirken, geleneksel ML ve ZF sembol dedektörlerinin yanı sıra, DBHS-ML ve disABC-ML isimli gelişmiş ML varyantları karşılaştırma amaçlı kullanılmıştır. Simülasyon sonuçları, önerilen DCSA-ML stratejisinin üstün performansını açık bir şekilde ortaya koymuştur. Çalışmada ele alınan sembol dedektörlerinin bit hata oranı analizlerinden de anlaşılacağı üzere önerilen strateji, her bir anten yapısı için optimuma en yakın BER seviyelerine ulaşarak, sadece

klasik ZF dedektöründen daha iyi sembol algılama performansı sergilemekle kalmayıp, kendisiyle aynı kategoride yer alan DBHS-ML ve disABC-ML isimli modern rakiplerini de geride bırakmıştır.

Kaynakça

- [1] Nissel, R., Schwarz, S., Rupp, M. 2017. Filter bank multicarrier modulation schemes for future mobile communications. *IEEE Journal on Selected Areas in Communications*, 34(10), 1768-1782.
- [2] Farhang-Boroujeny, B. 2014. Filter bank multicarrier modulation: A waveform candidate for 5G and beyond. *Advances in Electrical Engineering*, 2014(1), 1-25.
- [3] Cimini, L. J. 1985. Analysis and simulation of a digital mobile channel using orthogonal frequency division multiplexing. *IEEE Transactions on Communications*, 33(7), 665-675.
- [4] Peng, Y., Alexandropoulos, G. C., Li, Y. 2016. Pilot-assisted channel estimation with MSE-optimal thresholding for OFDM systems. *Transactions on Emerging Telecommunications Technologies*, 27(8), 1055-1070.
- [5] Zakaria, R., Le Ruyet, D. 2010. On maximum likelihood MIMO detection in QAM-FBMC systems. 21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 26-30 September, Istanbul, Turkey, 183-187.
- [6] Nissel, R., Blumenstein, J., Rupp, M. 2017. Block frequency spreading: A method for low-complexity MIMO in FBMC-OQAM. *IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 03-06 July, Sapporo, Japan, 1-5.
- [7] Spencer, Q. H., Swindlehurst, A. L., Haardt, M. 2004. Zero-forcing methods for downlink spatial multiplexing in multiuser MIMO channels. *IEEE Transactions on Signal Processing*, 52(2), 461-471.
- [8] Zhu, X., Murch, R. D. 2002. Performance analysis of maximum likelihood detection in a MIMO antenna system. *IEEE Transactions on Communications*, 50(2), 187-191.
- [9] Şimşir, Ş., Taşpınar, N. 2023. A powerful PTS strategy boosted by a novel discrete crow search algorithm for reducing the PAPR of UFMC signals. *Journal of Electrical Engineering*, 74(3), 197-209.
- [10] Askarzadeh, A. 2016. A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Computers & Structures*, 169, 1-12.
- [11] Kong, X., Gao, L., Ouyang, H., Li, S. 2015. A simplified binary harmony search algorithm for large scale 0-1 knapsack problems. *Expert Systems with Applications*, 42(12), 5337-5355.
- [12] Cheng, X., Liu, D., Feng, S., Pan, Q., Fang, H. 2018. PTS based on DisABC algorithm for PAPR reduction in OFDM systems. *Electronics Letters*, 54(6), 397-398.
- [13] Khan, A. A., Bashir, S., Naeem, M., Shah, S. I., Li, X. 2008. Symbol detection in spatial multiplexing system using particle swarm optimization meta-heuristics. *International Journal of Communication Systems*, 21(12), 1239-1257.
- [14] Seyman, M. N. 2022. Symbol detection based on back tracking search algorithm in MIMO-NOMA systems. *Computer Systems Science & Engineering*, 40(2), 795-804.
- [15] Li, L., Meng, W., Ju, S. 2016. A novel artificial bee colony detection algorithm for massive MIMO system. *Wireless Communications and Mobile Computing*, 16(17), 3139-3152.
- [16] Mandloi, M., Bhatia, V. 2016. A low-complexity hybrid algorithm based on particle swarm and ant colony optimization for large-MIMO detection. *Expert Systems with Applications*, 50, 66-74.
- [17] Seyman, M. N., Taşpınar, N. (2013). Symbol detection using the differential evolution algorithm in MIMO-OFDM systems. *Turkish Journal of Electrical Engineering and Computer Sciences*, 21(2), 373-380.
- [18] Wang, C., Au, E. K. S., Murch, R. D., Mow, W. H., Cheng, R. S., Lau, V. 2007. On the performance of the MIMO zero-forcing receiver in the presence of channel estimation error. *IEEE Transactions on Wireless Communications*, 6(3), 805-810.
- [19] Nguyen, T. T., Lampe, L. 2008. On partial transmit sequences for PAR reduction in OFDM systems. *IEEE Transactions on Wireless Communications*, 7(2), 746-755.