



Makale / Research Paper

**Ayrık PSO Algoritması ile Sehîm Kısıtı Altında İki Doğrultuda Çalışan Kirişli Döşemelerin Güvenilirlik Tabanlı Optimizasyonu**

Mehmet Kevser DERDİMAN<sup>1a\*</sup>

<sup>1</sup>Isparta Uygulamalı Bilimler Üniversitesi, Teknoloji Fakültesi, İnşaat Mühendisliği Bölümü, Isparta/TÜRKİYE  
[mehmetderdیمان@isparta.edu.tr](mailto:mehmetderdیمان@isparta.edu.tr)

Received/Geliş: 15.05.2021

Accepted/Kabul: 15.11.2021

**Öz:** Optimizasyon geliştirilen bilgi teknolojilerinin sonucu olarak mühendislik tasarımlarının en önemli konusu haline gelmiştir. Tasarım gereklilikleri ve koşullarını sağlayan çok sayıda alternatif arasından optimum değerlerin belirlenmesi yapı mühendisliği için de önemli bir problemdir. İki doğrultuda çalışan döşemelerde döşemenin sehîm sınırlarını aşmadan TS500 taşıma gücü kriterlerini sağlayacak çok sayıda alternatif çözümü mevcuttur. Bu alternatifler arasından en yararlı ve ekonomik olanın seçilmesi gereklidir. Eğer optimal tasarıma ilişkin bir kriter ortaya konulursa bu tasarımcı için önemli bir yol gösterici olacaktır. Meta sezgisel optimizasyon teknikleri arasında önemli bir yere sahip olan parçacık sürü optimizasyonu (PSO) algoritması popülasyon tabanlı arama algoritmalarından olup kullanımı oldukça yaygındır. Bu çalışmada PSO algoritması ayrık optimizasyon yapacak şekilde modifiye edilerek, iki doğrultuda çalışan ve değişen boyutlara sahip kirişli döşemelerin kısıtlar altında optimizasyon işlemi yapılmıştır. Optimizasyon işleminde TS500 şartları, taşıma gücü ve deplasman kriterleri kısıt olarak tanımlanmış, bu amaçla artan kısa kenar (2.2-8.6 m) ve değişen m ( $L_{long}/L_{short}$ ) değerleri ile farklı döşeme tipleri için modeller kurgulanmıştır. Seçilecek kesit ve donatının güvenilirlik tabanlı optimizasyonu yapılmıştır. Çalışmada toplam 5236 model analiz edilmiştir. Çözümleme sonucunda farklı döşemeler için optimal tasarım değerleri belirlenmiştir.

**Anahtar Kelimeler:** Optimizasyon, Parçacık sürü optimizasyonu, İki doğrultuda çalışan kirişli döşemeler.

**Reliability-Based Optimization of Two-Way RC Slabs with Beam Under Deflection Constraint Using Discrete PSO Algorithm**

**Abstract:** Optimization has become the most important subject of engineering designs as a result of developed information technologies. Determining optimal values among a large number of alternatives that provide design terms and conditions is also an important problem for Structural Engineering. In the design of two-way slabs, there are many alternatives that will meet the limit state criteria of TS500 without exceeding the deflection limits of the slab. Among these alternatives, it is necessary to choose the most useful and economical one. If a criterion for optimal design is put forward, this will be an important guide for the designer. The particle swarm optimization (PSO) algorithm, which occupies an important place among metaheuristic optimization techniques, is one of the population-based search algorithms and its use is quite common. As part of this study, the particle swarm algorithm was used. In this study, the PSO algorithm has been modified to perform reliability-based discrete optimization and the optimization of slabs with two-direction changing dimensions has been performed. In the optimization, TS500 conditions, limit state and displacement criteria are defined as constraints. For this purpose, models were created for different slab types with increasing short edge (2.2-8.6 m) and changing m ( $L_{long}/L_{short}$ ) values. The section and reinforcement to be selected are optimized by reliability-based discrete (discontinuous) optimization analysis. A total of 5236 models were analyzed in the study. Optimal design values for different slabs are determined as a result of the analysis.

**Keywords:** Optimization, Particle swarm optimization, two-way slab with beams.

*Bu makaleye atf yapmak için*

Derdiman, M.K., "Ayrık PSO Algoritması ile Sehîm Kısıtı Altında İki Doğrultuda Çalışan Kirişli Döşemelerin Güvenilirlik Tabanlı Optimizasyonu" El-Cezerî Fen ve Mühendislik Dergisi 2022, 9(1); 49-64.

*How to cite this article*

Derdiman, M.K., "Reliability-Based Optimization of Two-Way RC Slabs with Beam Under Deflection Constraint Using Discrete PSO Algorithm" El-Cezerî Journal of Science and Engineering, 2022, 9(1); 49-64.

ORCID ID: \*0000-0003-2359-3120

## 1. Giriş

Üretim, ürün veya hizmet tabanlı sistemler, sınırlı kaynakların bir sonucu olarak sürekli bir rekabet ile karşı karşıyadır. Ürün ve hizmet endüstrileri gibi ekonominin tüm sektörlerinde önemli kaynaklar kısıtlara tabidir. Bu kısıt kaynakların en ideal kullanım amacı stratejik araştırmalara yol açmıştır. Bu araştırmalar da, ürün ve hizmet tabanlı sistemlerde sınırlı kaynak sorununu çözmek için optimizasyon algoritmalarının gelişim sürecine yol açmıştır [1,2].

Betonarme dünyada en çok kullanılan yapı malzemesi olduğundan, bu konuda bilimsel bilgi son yıllarda büyük ölçüde artmıştır. Günümüzde, basit bir betonarme yapının tasarımı, birçok kısıt ve parametrelere bağımlı olarak gerçekleştirilir, bu nedenle bilgisayar kullanımı artık neredeyse zorunludur. Bu bağlamda, yapı mühendislerinden, basitleştirilmiş tasarım yöntemlerine olan talep, en azından betonarme elemanlarının ön tasarımı için artmıştır [3]. Yapısal optimizasyon alanında önemli sayıda araştırmalar yapılmış olup, büyük çoğunluğu akademik veya küçük problemlerle ilgilidir. Onlarca yıldır bu optimizasyon teknolojisi, yapı mühendislerinin pratik uygulamalarında pek yer bulamamış fakat, son on yılda, optimizasyon teknolojilerinin yapısal mühendislik uygulamalarında kayda değer gelişmeler olmuştur [2].

Optimizasyon sürecinde, ulaşılması amaçlanan hedefler ve karşılanması zorunlu kısıtlamalar vardır. Belirlenen amaç, her problem türü için farklı bir fonksiyondur. Kısıtlar ise optimize edilmek istenen ürünün değişkenleri arasındaki bazı zorunlu oransal veya sınırlı ilişkilerden kaynaklanmakta olup bu da her problem için farklı tanımlanması gereken eşitlikler ve/veya eşitsizliklerden kaynaklanan koşullardır.

Gerçek dünya uygulamaları ne olursa olsun, bir optimizasyon problemi genel bir formda formüle etmek mümkündür. Amacı belirgin tüm optimizasyon problemleri genel olarak, doğrusal olmayan bir şekilde kısıtlı bir optimizasyon problemi olarak ifade edilebilir.

$$\begin{aligned} \text{Maksimum} - \text{Minimum} : \quad & f(x), x = (x_1, x_2, \dots, x_D)^T \in \mathbb{R}^D, \\ \text{Kısıt Şartları:} \quad & g_j(x) \leq 0 \quad (j = 1, 2, \dots, m), \\ & h_k(x) = 0 \quad (k = 1, 2, \dots, n), \end{aligned} \quad (1)$$

Burada  $f(x)$ ,  $g_j(x)$  ve  $h_k(x)$  optimizasyon probleminin skaler fonksiyonlarıdır. Burada  $x$  değişkeni sürekli, ayrık veya bunların bir karışımı olabilir. Vektör  $x = (x_1, x_2, \dots, x_D)^T$ , tasarım veya karar değişkenleri olup,  $D$  boyutlu uzayda değişen değerlerdir ve karar vektörü olarak adlandırılır [4].

Optimizasyon sürecinde matematiksel yaklaşım veya meta sezgisel yaklaşım uygulanabilir.

Matematiksel diğer adı ile klasik optimizasyonda doğrusal programlama, doğrusal olmayan programlama (kuadratik programlama ve geometrik programlama), dinamik programlama, integer programlama, stokastik programlama, lagrange çarpanları metodu dahil olmak üzere geniş bir optimizasyon teknikleri sınıfı vardır [5]. Her yerde ve uygulamada optimizasyon vardır. Ancak farklı bakış açıları farklı anlamlara sahip olabilir. Bazı optimizasyon problemleri temel matematik araçları ile kolayca çözümlenebilir, örneğin  $x \in \mathbb{R}$  olmak üzere  $f(x) = x^4 + 2x^2 + 1$  gibi bir fonksiyonun minimum değeri gradyan tabanlı yaklaşım ile fonksiyonun türevi sıfıra eşitlenerek,  $x = 0$  çözümü elde edilir [6]. Buna karşılık bazı optimizasyon problemleri oldukça zorlu olabilir, amaç fonksiyon çok değişkenli ve değişkenler doğrusal olmayan ilişkilere sahip olduğunda problemi temel matematik yöntemlerle çözümlenemeyen olabilir. Bu durumda

yine problemin boyutuna ve doğrusallık durumlarına bağlı olarak yinelemeli hesaplar içeren farklı algoritmalara sahip yöntemler kullanılmalıdır.

Genel olarak, optimizasyon problemleri son derece doğrusal olmayan ve çok modludur; bu geleneksel algoritmalar genellikle doğrusal olmayan problemlerle başa çıkmak için yeterlidir, bir çözüm bulunabilir, ama yine de yerel optimal çözümler olma eğilimindedir. Karmaşık kısıtlamalarla son derece doğrusal olmayan problemlerde küresel optimal çözümler bulmak için etkili yöntemler yoktur. Bu gibi durumlarda, karmaşık doğadan ilham alan optimizasyon algoritmaları pratikte umut verici alternatifler olma eğilimindedir. Son zamanlarda, bilişim teknolojilerinin gelişimi ile bu tekniklerin hem uygulanması yaygınlaşmış, hem de hızlı bir şekilde önemli gelişmeler olmuştur [6].

Doğadan esinli algoritmaların yaygın kullanılanları genetik algoritma, parçacık sürüsü optimizasyon algoritması, tavlama benzetimi algoritması, harmoni arama algoritması ve karınca kolonisi optimizasyon algoritmasıdır, daha nadir kullanılanları ise yarasa algoritması, guguklu arama algoritması, diferansiyel evrim algoritması, ateşböceği algoritması, çiçek tozlaşma algoritması, yusufçuk algoritması ve benzeri çok sayıda algoritmadır.

Doğadan esinli algoritmalar, özellikle de sürü zekasına dayananlar, gerçek dünyadaki uygulamalarda çeşitli optimizasyon problemlerini çözmek için başarıyla uygulanmıştır ve bu nedenle popülariteleri son yıllarda önemli ölçüde artmıştır. Bu optimizasyon algoritmalarının uygulamaları, biyoloji, kimya alanlarından mühendislik tasarımlarına, iş ve makine kullanım planlanmasından araç rota planlamasına olmak üzere çok çeşitlidir. Sonuç olarak, son birkaç yılda bu alanlarda yayınlanan birkaç bin yeni araştırma makalesi ile önemli ilerleme kaydedilmiştir [7].

Doğadan ilham alan optimizasyon algoritmaları, biyolojik evrim, sürü davranışı ve fiziksel ve kimyasal süreçler ilkelerinden geliştirilen meta sezgisel algoritmalarlardır. Bu optimizasyon algoritmaları, algoritmalara zekayı dahil ettikleri için doğa-esinli hesaplamalı zeka teknikleridir. Bu algoritmalar üzerine yapılan araştırmalar sayesinde son yirmi yılda önemli sıçramalar ile büyümüştür. İlk atılım 1960'larda John Holland ve Michigan Üniversitesi'ndeki meslektaşları tarafından evrimsel genetik algoritmanın (GA) öncü gelişimi ile gerçekleşti. O zamandan beri, GA'nın birçok çeşidi ve melezleri de dahil olmak üzere çeşitli evrimsel algoritmalar ortaya çıkmıştır. Evrimsel algoritmalar biyolojik evrime dayanır ve GA bu kategorideki klasik örneklerden biridir. Sürü zekasına dayanan algoritmalar ise kuş sürüsü, karınca sürüsü, balık sürüsü, fil sürüsü ve benzeri gibi doğadaki sürülerin davranışlarından esinlenen algoritma kategorisidir. Sezgisel ile birlikte arama aracı popülasyonlarının kullanılması, karmaşık mühendislik tasarım problemlerinin çözümleri üzerinde derin bir etkiye sahiptir. Doğadan ilham alan algoritmalar, hızlı bir şekilde çözümlere kolayca ulaşmada etkili yöntemlerdir [5].

Sezgisel algoritmalar, yeni çözümler üretmede bir deneme-yanılma yaklaşımı kullanırken, meta sezgisel algoritmalar, bellek, çözüm geçmişi ve diğer 'öğrenme' stratejisi biçimlerinin kullanımıyla daha üst düzey bir sezgisel yaklaşım sunarlar. Geleneksel algoritmaların aksine, meta sezgisel algoritmalar esas olarak global arama için tasarlanmıştır ve birçok avantajlara sahiptir:

- Global çözümü bulma olasılığı daha yüksektir.
- Problemleri belirli bir bilgi olmadan kara kutu olarak ele alırlar, böylece daha geniş bir problem yelpazesini çözebilirler.
- Meta sezgisel algoritmalar genellikle gradyansız yöntemlerdir ve herhangi bir türev bilgi kullanmazlar ve bu nedenle çok karmaşık doğrusal olmayan problemlerle ve süreksizlik problemleriyle başa çıkabilirler [7].

## 1.1. PSO Algoritması

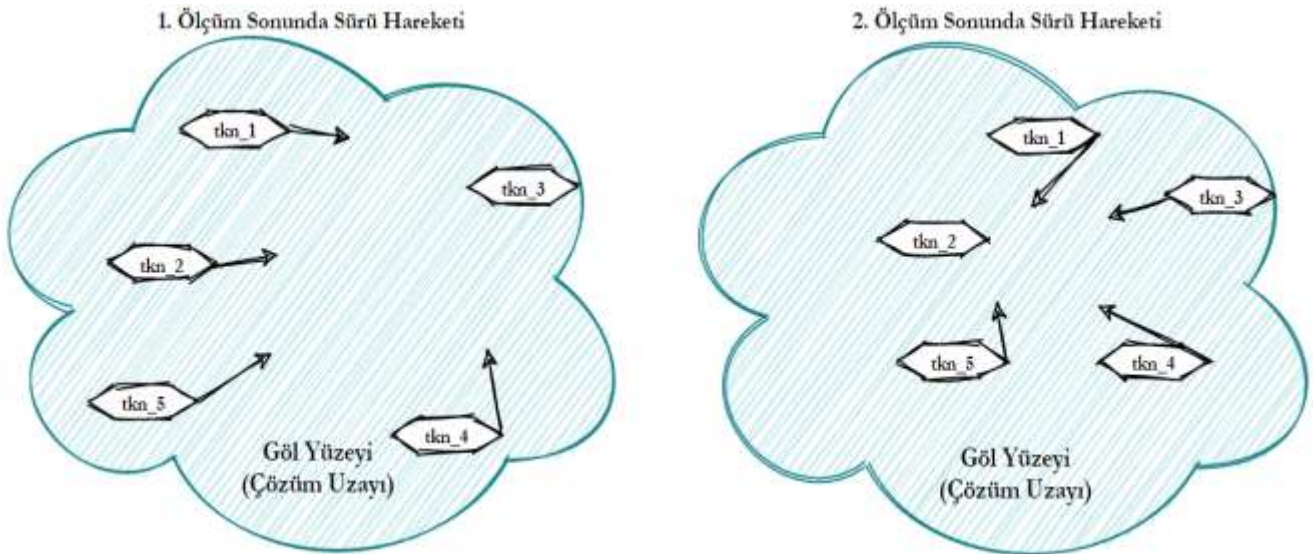
Parçacık sürüsü optimizasyonu (PSO) 1995 yılında Kennedy ve Eberhart tarafından geliştirilmiştir [8], kuş ve balık sürülerinin sosyal davranışlarını simüle eden denklemler kullanır. PSO algoritması koloni sürüsü istihbarat paradigması üzerine kuruludur. James Kennedy ve Russell Aberhart hayvanların sosyal davranışlarını tanımlamak için bir model geliştirmeye çalışıyorlardı, geliştirdikleri modelin optimizasyon yapabildiğini fark ettiler. Karınca koloni ve parçacık sürü algoritmaları, sürü zekasının ilk örnekleridir.

PSO'daki sürünün operasyonel arama yapan akıllı bir optimizasyon sürüsü olduğu söylenebilir. PSO bilim ve mühendisliğin çeşitli alanlarında başarıyla uygulanarak çeşitli optimizasyon problemlerini çözmek için en kullanışlı ve en popüler algoritmalarından biri haline gelmiştir, PSO çok basit ancak güçlü bir algoritmadır.

Parçacık sürüsü optimizasyonu (PSO), sürekli ve ayrık optimizasyon problemlerini çözmek için popülasyona dayalı stokastik bir yaklaşımdır. Burada, parçacıklar olarak adlandırılan ve yazılımda vektör ile ifade edilen aday çözüm değerleri, bir optimizasyon probleminin arama alanı içinde belirli bir hızda değişir. Her parçacık, kuş akısının davranışsal modellerinden esinlenen kurallara göre hızını değiştirerek arama alanında daha iyi pozisyonlar arar [9].

PSO'da çözüme ulaşmak için aramaya katılan her bir sürü ferdine parçacık, parçacıkların bulunduğu popülasyona ise sürü adı verilir. Parçacıkların (çözüm vektörünün) amaç fonksiyon içinde yerine konması ile uygunluk değerleri belirlenir. Uygunluk değeri en iyi olan sürünün en iyisi olarak adlandırılır. Parçacıkların her konum için amaç fonksiyonu hesaplanırken parçacığın kendine ait en uygun değeri de parçacık bilgisi içinde saklanır ve bu da parçacığın en iyisi olarak adlandırılır. Parçacıklar yeni yön ve hızı bu iki bilgi ile belirlenir.

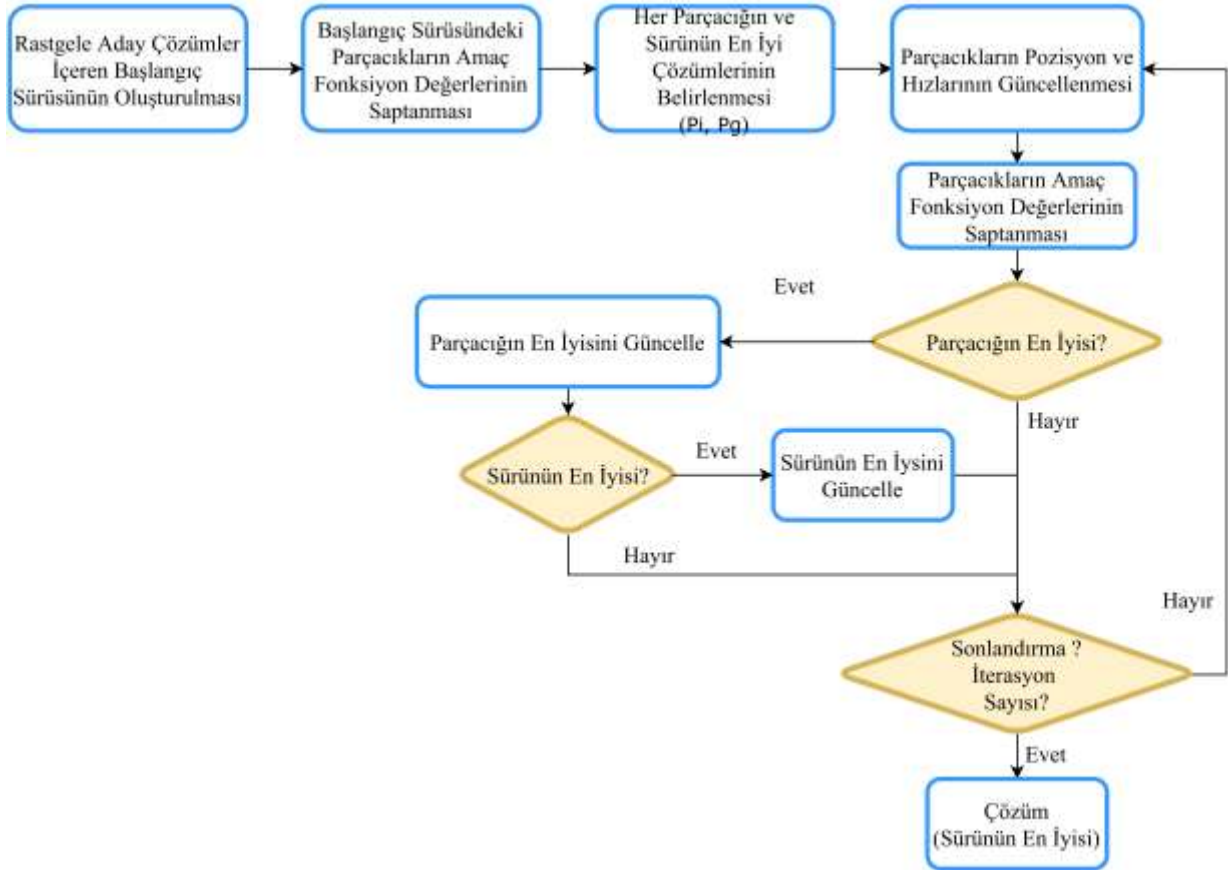
PSO'daki sürü hareketi bir göl analogisi ile anlatılabilir. Örneğin bir göl üzerinde dağılmış tekneler tarafından gölün en derin noktası tespit edilmeye çalışılıyor olsun. Eğer gölün yüzeyinde bir nokta orijin kabul edilirse, üç boyutlu koordinat sisteminde maksimum derinlik aranan noktayı ifade edecektir. Göl yüzeyinde dağılmış  $tkn_1$   $tkn_2$ ,  $tkn_3$   $tkn_4$  ve  $tkn_5$  ölçüm yapan tekneler olduğunu varsayalım. Birinci ölçüm sonunda sürünün en iyi değeri  $tkn_3$  tarafından alınmış ise diğer tekneler doğru ve hızlarını  $tkn_3$  e yönlendirecek ve en iyi değerden uzaklığı oranında iyiye yaklaşma hızına sahip olacaktır.



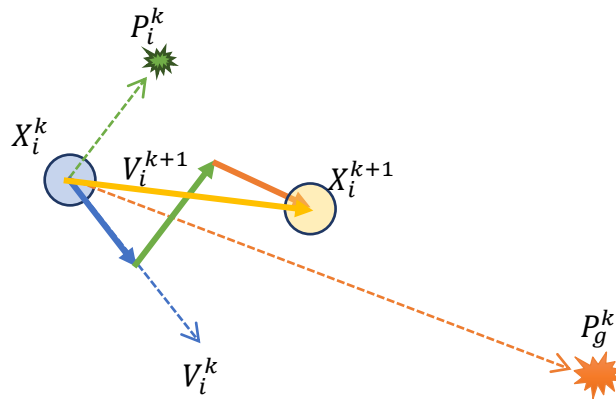
Şekil 1. Sürü-Parçacık hareketliliğinin göl analogisi ile gösterimi

Sonraki ölçümde parçacıklar yeni pozisyonlarındaki ölçümleri tekrar kontrol edip kendi en iyi değerini ve sürünü en iyi değerini değerlendirerek yeni yön ve hızlara sahip olacaklardır. Bu yinelemeli olarak devam ettiğinde ve yeter sayıda parçacık mevcut ise çözüm uzayını temsil eden tüm göl yüzeyi araştırılmış olacaktır.

Şekil 1’de yukarıda anlatılan parçacık hareketliliği gösterilmiştir. Burada tekne sayısı popülasyon sayısını (nüfusu), her ölçüm ve değerlendirme iterasyon içi işlemleri, göl yüzeyi çözüm uzayını ve global olarak bulunan en iyi değer çözümü ifade etmektedir. PSO’nun genel algoritması Şekil 2’de verilmiştir.



Şekil 2. Parçacık sürü optimizasyonu algoritması



Şekil 3. Parçacığın k ve k+1’nci döngüdeki pozisyon ve hızının vektörel olarak hesaplanması ve güncellenmesi

Parçacıklar arama alanında yer değiştirir ve her birinin konumu, her yinelemede tüm parçacıkların en iyi konumlarına ve kendi en iyi konumuna göre güncellenir. Şekil 3’de gösterilen konum ve hız güncellemesinin hesabı Denklem (2),(3)’de verilmiştir [10].

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k) \quad (2)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (3)$$

Burada  $X_i^k$  ve  $V_i^k$  parçacığın geçerli pozisyon ve hızını,  $X_i^{k+1}$  ve  $V_i^{k+1}$  parçacığın güncellenmiş pozisyon ve hızını,  $\omega$  atalet katsayısını,  $c_1$  ve  $c_2$  parçacığın kendi en iyisine (bilişsel bileşen) ve sürünün en iyisine (sosyal bileşen) doğru ivme değerini,  $r_1$  ve  $r_2$  birden küçük sıfırdan büyük rastgele sayı değerini,  $P_i^k$  ve  $P_g^k$  ise parçacık ve sürüye ait en iyi pozisyon değerlerini,  $c_1 r_1 (P_i^k - X_i^k)$  bilişsel bileşen katkısını ve  $c_2 r_2 (P_g^k - X_i^k)$  sosyal bileşen katkısını ifade etmektedir. PSO temel algoritması  $\omega$  atalet katsayısının her iterasyonda azaltılmasını öngörmektedir. Dolayısıyla iterasyonun her döngüsünde sönüm katsayısı  $\omega_{dam}$  ile çarpılarak azaltılmaktadır. Örneğin atalet katsayısının 1 ve sönümün 0.95 olduğu bir durum için ikinci döngüde  $\omega = \omega * \omega_{dam}$  işlemi sonucu atalet katsayısı 0.95 olacaktır.

## 1.2. Ayrık Optimizasyon

Optimizasyon algoritmaları, amaç fonksiyon değişkenlerinin sürekli değerlerinde arama yapmak için tasarlanmışlardır. Bununla birlikte, değişkenlerin süreksiz değerler olma zorunluluğu (yalnızca 10’ un katları olma şartı gibi), problemi sürekli optimizasyon olma durumundan çıkartır. Örneğin giriş gövde genişliğinin veya giriş yüksekliğinin belirli değerlerle artarak (5’in katları veya 10’un katları gibi) değişme durumu, giriş yüksekliğinin veya gövde genişliğinin değişken olarak dahil olduğu problemi ayrık optimizasyon problemine dönüştürür. Örneğin kullanılacak beton sınıfının hesaplarda ele alınacak dayanım değerleri için de aynı durum söz konusudur. Mühendislik problemlerinin çoğunda bu durum mevcuttur ve optimizasyon problemi karşımıza ayrık optimizasyon problemi olarak çıkar. Ayrık optimizasyonda çoğunlukla kombinatorik çözüm mevcuttur, örneğin donatı çapı için  $\phi 8, \phi 10, \dots$  gibi. Bu da çözüm algoritmasında değişikliği gerektirecektir.

## 1.3. Güvenilirlik Tabanlı Optimizasyon

Mekanik sistemlerde güvenilirlik, otomobil, havacılık, uzay, inşaat mühendisliği ve savunma endüstrilerinde tasarımın odağında yer alır [11].

Bu temel güvenilirlik ilkesi olarak bir yapıda aranan en önemli özellik, yapının öngörülen yüklerin olası en elverişsiz etkime durumunda dahi geçmeden ayakta kalabilmesi ve servis yükleri altında yapı elemanlarında aşırı deformasyon, çatlama ve titreşim oluşmamasıdır [12]. Bu durumda tasarımdaki ilgili eleman dayanımlarının yük etkisinden büyük veya eşit, deplasman miktarının ve çatlak genişliğinin ise yapı kullanım amacına göre belirlenecek kısıtlar içinde kalması şartının sağlanmasını gerektirir.

Malzeme davranışı üzerine kabuller tasarlanan eleman dayanım hesabında önemli bir etkiye sahiptir. Dolayısıyla malzemelere ait mekanik özellikler dayanım hesabını etkileyen önemli değişkenlerdir. Özellikle betonarmede malzemenin mekanik özelliklerinin zamana bağlı değişkenliği veya üretim aşamasındaki belirsizlikler nedeni ile deterministik olarak kesin bir değer belirlemek mümkün değildir. Ayrıca yük etkisinin belirlenmesinde kullanılacak olan yüklerin de, sabit tek bir değer alınması olanaksızdır. Dolayısıyla hem malzeme özelliklerine hem yük değerlerine ilişkin değerlerin belirlenmesi ancak istatistiksel yaklaşımlarla mümkündür.

İstatistiksel arařtırmalar çerçevesinde TS498 [13] yük deęerleri için sınır şartlarını ve TS500 [14] de dayanım ve malzeme şartları ve bunlara ait güvenlik katsayılarını vermiřtir.

Denklem (4)'de TS500'deki "tařıma gücü sınır durumu" şartı verilmiřtir. Tasarımdaki elemanlar için Denklem (5)'de belirtilen biçimde, azaltılmıř malzeme dayanımları (tasarım dayanımları) kullanılarak hesaplanan tařıma gücü deęerlerinin, artırılmıř tasarım yükü ile hesaplanan iç kuvvet deęerlerinden hiçbir zaman küçük olmadığı kanıtlanmalıdır.

$$R_d \geq F_d \quad (4)$$

Buradaki  $R_d$ : dayanımı ve  $F_d$ : yük etkisini göstermek üzere Denklem (5)'deki gibi hesaplanır.

$$R_d = R_k/\gamma_m ; F_d = \gamma_Q F_Q + \gamma_G F_G \quad (5)$$

Buradaki  $\gamma_m$  malzeme katsayısını,  $\gamma_Q, \gamma_G$  yük katsayılarını,  $R_k$  karakteristik dayanımı ve  $F_Q, F_G$  karakteristik yük etkilerini göstermektedir.

Optimizasyonda bu şartlar belirleyici bir kısıt olarak tanımlanması durumunda güvenilirlik esaslı bir optimal tasarım gerçeleřtirilmiř olacaktır. Güvenilirlik esaslı bir optimizasyon için bu Denklem (6)'deki řekilde matematiksel olarak ifade edilebilir.

$$\text{Güvenilirlik Kısıtları: } \begin{cases} (g_R)_i(x) \leq (g_F^*)_i \\ i = 1, 2, \dots, j \end{cases} \quad (6)$$

Kullanım yükü etkisindeki yapıda veya elemanda yer deęiřtirme veya sehim belirli sınırları ařmamalıdır. Ařırı sehim taşıyıcı olmayan elemanlarda hasara neden olur ve eęer varsa, hassas makinelerin iřlevini bozar.

Tařıma gücü ilkelerine göre tasarlanmıř, yani dayanım açasından güvenli donatılmıř bir kiriřin veya döřemenin ne kadar sehim yapacaęının belirlenmesi ve izin verilen sınırı ařıp ařmadığının kontrol edilmesi gerekir. Son yıllarda, artan beton kalitesi ve dayanımı (C30/37, C40/50 gibi) sonucu mühendisler kiriř, döřeme ve kolonların kesitlerini küçültme eęilimindedirler. Bunun sonucu olarak ařırı sehim oluřmaktadır. Sehim hesabı yaklařıktır, kesin deęeri hesaplamak imkansızdır [15].

Literatür taramasında betonarme döřemelerin optimizasyonu üzerine kısıtlı sayıda çalıřmanın yapıldığı görülmüřtür. Aldwaik ve Adeli [16] çalıřmalarında düzensiz yüksek yapılarında betonarme döřemelerinin maliyet optimizasyonu üzerine, Ghandi ve dię. [17] "guguk kuřu (cuckoo) optimizasyon algoritması (COA)" modeli ile tek doęrultuda ve iki doęrultuda çalıřan betonarme döřemelerin optimizasyonu üzerine, Garcia ve dię. [18] tek doęrultuda çalıřan betonarme döřemelerin yařam döngüsü analizi ile optimizasyonu üzerine, Ahmadi-Nedushan ve Varae [19] tek doęrultuda çalıřan döřemelerin PSO ile maliyet optimizasyonu üzerine yaptıkları çalıřmalar ön plana çıkmaktadır.

Bu çalıřmada iki doęrultuda çalıřan kiriřli döřemelerin güvenilirlik tabanlı ayrık optimizasyon yapılarak, yüküne, tipine ve  $L_u/L_k$  oranına ve  $L_k$  boyuna baęlı optimal kesit yükseklięi belirlenmeye çalıřılmıřtır. Bu amaçla PSO algoritması modifiye edilerek ayrık parçacık sürü optimizasyon algoritması tasarlanmıř, TS500 tařıma gücü yöntemi, sınır şartları ve deplasman kriterleri dikkate alınarak kombinatorik çözüm uzayı içinden optimal çözüm kümesi (kesit deęerleri) arařtırılmıřtır. Bu iřlem yapılırken amaç fonksiyon, aęırlığı deęil, fiyatı minimize edecek řekilde tasarlanmıřtır.

## 2. Materyal ve Metot

Optimizasyonda PSO algoritması, güvenilirlik koşullarını ve kısıtları içeren fonksiyonların tamamı Python programlama dilinde kodlanarak çalışılmıştır. Kodlamada ayrık optimizasyonun gerçekleştirilmesi için parçacık hızı ve ivmesine de bağlı olarak ortaya çıkan değişken değerleri ilgili değişkenin alabileceği değerler kümesinden en yakın olana yuvarlanarak sürekli değil ayrık değerlerin elde edilmesi sağlanmıştır. Yuvarlama için Python matematiksel işlemleri kullanılarak optimizasyon hızının yavaşlamasının önüne geçilmiştir. Şekil 4’de parçacık hızını, parçacık değişken değerini ve ayrık optimizasyon değişken değerlerinin elde edilmesini sağlayan ilgili program kodunun “Spyder ide” ara yüzündeki görüntüsü bulunmaktadır.

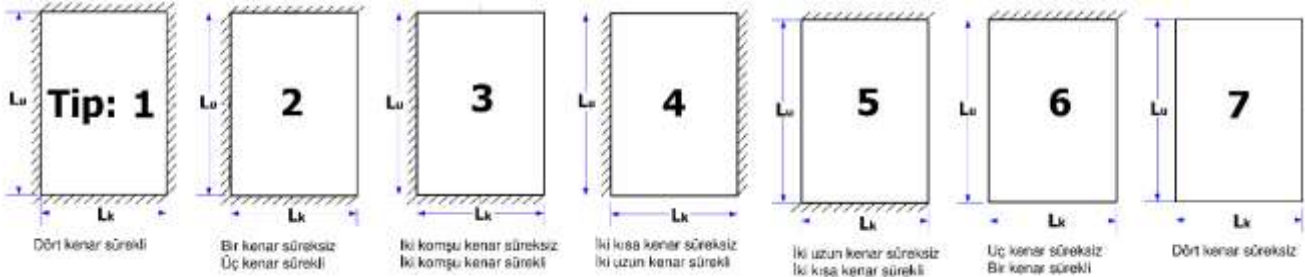
```
# Parçacık hızları düzenleniyor:
particles[i]['Vlcty'] = w * particles[i]['Vlcty'] \
+ c1 * np.random.uniform(0,1,nVar) * (particles[i]['bestVars'] - particles[i]['Vars']) \
+ c2 * np.random.uniform(0,1,nVar) * (eniyei['bestVars'] - particles[i]['Vars'])

# Parçacık pozisyonu hız eklenerek güncelleniyor:
particles[i]['Vars'] += particles[i]['Vlcty']

# Parçacıkların değişkenleri kendileri için mümkün değerlere yuvarlanıyor
particles[i]['Vars'][0] = particles[i]['Vars'][0]//10*10
particles[i]['Vars'][1] = particles[i]['Vars'][1]//2*2
particles[i]['Vars'][2] = particles[i]['Vars'][2]//10*10
particles[i]['Vars'][3] = particles[i]['Vars'][3]//2*2
particles[i]['Vars'][4] = particles[i]['Vars'][4]//10*10
particles[i]['Vars'][5] = particles[i]['Vars'][5]//8*8
particles[i]['Vars'][6] = particles[i]['Vars'][6]//10*10
particles[i]['Vars'][7] = particles[i]['Vars'][7]//8*8
particles[i]['Vars'][8] = particles[i]['Vars'][8]//10*10
```

Şekil 4. Ayrık optimizasyon değişken değerlerinin elde edilmesini sağlayan kod parçasının Spyder ara yüzündeki görüntüsü

Şekil 5’de optimizasyonu çalışılan döşeme tipleri gösterilmiştir. Optimizasyonda PSO algoritması, güvenilirlik koşullarını ve kısıtları içeren fonksiyonların tamamı Python programlama dilinde kodlanarak çalışılmıştır.



Şekil 5. Optimizasyonu çalışılan TS500'e göre döşeme tipleri

Optimizasyon 9 boyutlu olarak tasarlanmıştır. Dolayısıyla değişken sayısı  $n = 9$ 'dur ve bu değişkenler, döşeme kalınlığı  $x_1$ , kısa doğrultu açıklık donatı çapı  $x_2$ , kısa doğrultu açıklık donatı aralığı  $x_3$ , uzun doğrultu açıklık donatısı çapı  $x_4$ , uzun doğrultu açıklık donatısı aralığı  $x_5$ , kısa doğrultu mesnet ek donatı çapı  $x_6$ , kısa doğrultu mesnet ek donatı aralığı  $x_7$ , uzun doğrultu mesnet ek donatısı çapı  $x_8$  ve uzun doğrultu mesnet ek donatısı aralığı  $x_9$  şeklinde alınmıştır. Python'da dizi elemanları sıfırdan başladığı için kodlama ortamında değişkenlerimiz sırasıyla  $x[0]$ ,  $x[1]$ ,  $x[2]$ ,  $x[3]$ ,  $x[4]$ ,  $x[5]$ ,  $x[6]$ ,  $x[7]$  ve  $x[8]$  şeklindedir.

PSO algoritma performansı algoritmada yer alan kontrol parametrelerinden önemli derecede etkilenir. Bunlar, problemin boyutu, popülasyon miktarı (parçacık sayısı), eylemsizlik katsayısı ( $\omega$ ), eylemsizlik sönüm katsayısı ( $\omega_{dam}$ ), hızlanma katsayıları ( $c_1, c_2$ ), ve bilişsel ve sosyal bileşenlerin katkı oranları olan rastgele sayılar ( $r_1, r_2$ ) ve döngü (iterasyon) sayısıdır [20]. Bu parametrelerden



problem boyutu optimizasyon modeline bağımlıdır ve bu problem için ilgili parametreler Tablo 1’de verilmiştir.

Tüm optimizasyon modelleri için alt sınır olarak kullanılan beton sınıfı C25/30 alınmıştır. Betonun denetimli olduğu varsayımı ile malzeme katsayısı  $\gamma_{mc} = 1.5$  ile hesap yapılmıştır. Betonarme çeliğinin hesaplarda karakteristik dayanımı  $f_{yk} = 420 \text{ N/mm}^2$  ve malzeme katsayısı  $\gamma_{ms} = 1.15$  alınmıştır. Paspayı tüm modellerde  $c_o = (15 + \phi/2)$  mm alınmıştır.

Hareketli yük karakteristik değeri 1.5, 2, 3.5 ve 5 kN/m<sup>2</sup> [13] ve ölü yük karakteristik değeri ise döşeme ağırlığı, 5 cm tesviye ağırlığı, 2 cm seramik kaplama ağırlığı ve 2 cm tavan sıvası ağırlığı toplanarak elde edilmiş olup, burada döşeme kalınlığı her döngüsel işlem için ve her parçacık için değişken olduğundan plak kalınlığına ( $x_1$ ’e ) bağımlı bir fonksiyon olup sürekli değişen değere sahiptir.

Kısa kenar  $L_k$  değeri 2.2 m den başlayıp 0.4 m artım ile 8.6 m’ye kadar değişen değerler ile analiz gerçekleştirilmiştir.

Optimizasyonda tercih edilen parametre değerleri Tablo 1’de, optimizasyona ait değişkenlerin alt ve üst sınırları ise Tablo 2 ‘de verilmiştir.

**Tablo 1.** PSO algoritma parametreleri

Parametre ismi	Değeri
Döngü (iterasyon) sayısı:	150
Popülasyon nüfusu (parçacık sayısı):	350
Eylemsizlik katsayısı ( $\omega$ ): $w$	0.999
Eylemsizlik sönüm katsayısı ( $\omega_{dam}$ ):	0.97
Birey ivme (hızlanma) katsayısı ( $c_1$ ):	1.497
Sürü ivme (hızlanma) katsayısı ( $c_2$ ):	1.497

**Tablo 2.** Optimizasyona ait değişkenlerin alt ve üst sınırları

Değişkenler	Alt Sınır	Üst Sınır	Değişim Aralığı
Döşeme kalınlığı ( $h_f$ ) : $x_1$ ( $mm$ )	80	500	10
Kısa doğrultu açıklık donatı çapı: $x_2$ ( $mm$ )	8	12	10
Kısa doğrultu açıklık donatı aralığı $x_3$ ( $mm$ )	50	200	2
Uzun doğrultu açıklık donatısı çapı $x_4$ ( $mm$ )	8	12	10
Uzun doğrultu açıklık donatısı aralığı $x_5$ ( $mm$ )	50	250	2
Kısa doğrultu mesnet ek donatı çapı $x_6$ ( $mm$ )	8	12	2
Kısa doğrultu mesnet ek donatı aralığı $x_7$ ( $mm$ )	0	1000	10
Uzun doğrultu mesnet ek donatısı çapı $x_8$ ( $mm$ )	8	12	2
Uzun doğrultu mesnet ek donatısı aralığı $x_9$ ( $mm$ )	0	1000	10

Analizler TS500 ‘ün [14] taşıma gücü varsayım ve yöntemleri ile hesap yapılmıştır.

$$Optimizasyon: \begin{cases} \min_{x_i \in [x_i^L, x_i^U]} f(x_1, x_2, \dots, x_n) \\ \text{Güvenilirlik Kısıtları: } \begin{cases} (g_R)_i(x) - (g_F^*)_i \leq 0 \\ i = 1, 2, \dots, m \end{cases} \\ \text{Diğer Kısıtlar: } \begin{cases} g_i(x) \leq 0 \\ i = 1, 2, \dots, j \end{cases} \end{cases} \quad (7)$$

Bu optimizasyon probleminin genel matematiksel ifadesi Denklem (7)'da verilmiştir.

Amaç fonksiyonunun ( $f(x_1, x_2, \dots, x_n)$ ) doğru şekilde ifade edilmesi, hedeflenen çözümün verimli bir şekilde seçilmesini sağlayacaktır. Amaç fonksiyonun yanlış ifade edilmesi, anlamsız çözümlere neden olabilir [21]. Bu çalışmada amaç fonksiyonu ağırlığı değil, fiyatı minimize edecek şekilde tasarlanmıştır. Beton ve betonarme çeliğinin ikisi de  $m^3$  birim fiyatı dikkate alınarak, döşemenin birim genişlik için amaç fonksiyonun Denklem (9)'deki gibidir.

$$pen\_fon(x) = kısıtlar(x, g(x)) \quad (8)$$

$$f(x_1, \dots, x_9) = BF * \frac{x_1}{10^3} + \left[ Asmx(x_2, x_3) + Asmy(x_4, x_5) + \frac{Asex(x_6, x_7)}{4} + \frac{Asey(x_8, x_9)}{4} \right] * \frac{CF}{10^6} + pen\_fon(x) \quad (9)$$

Burada mesnet ek donatıları tüm döşeme yüzeyinde olmayıp, dörtte birlik kısmında yer alacağı için bu donatı kesitlerinin dörtte biri amaç fonksiyonunun da yer almıştır. Amaç fonksiyonun değişkenleri, kısıtlara uymaması durumunda Denklem (8)'de verilen penaltı fonksiyonu ( $pen\_fon(x)$ ) uygulaması yapılmıştır. Penaltı fonksiyonu da her bir kısıtlama için ayrı olmakla birlikte basit olarak, kısıtlamaları aşan her değişken için büyük bir değer üretecek şekilde tasarlanmıştır. Toplam 13 kısıt tanımlanmıştır, bunlar sırasıyla:

- Minimum döşeme kalınlığı kısıtlaması,
- Minimum kısa doğrultu açıklık donatı oranı kısıtlaması,
- Minimum kısa doğrultu açıklık donatısı aralığı kısıtlaması (burada iki kısıtlama var),
- Minimum iki doğrultudaki toplam donatı oranı kısıtlaması,
- Minimum uzun doğrultu açıklık donatı oranı kısıtlaması,
- Minimum uzun doğrultu açıklık donatısı aralığı kısıtlaması (burada da iki kısıtlama var),
- İki adet açıklıkta ve iki adet mesnette güvenilirlik kısıtlaması (taşıma gücü yeterliliği),
- Son olarak da deplasman (sehim) kısıtlamasıdır.

Aşağıda uygulanan güvenilirlik kısıtlarından biri örnek olarak verilmiştir. Kısa doğrultuda açıklık taşıma gücü fonksiyonu  $M_{r_{mx}}$  Denklem (10)'de, ilgili güvenilirlik kısıtlaması Denklem (11)'de verilmiştir. Denklem (11)'de yer alan  $M_d$  tasarım yükü ve TS500-Çizelge 11.1 katsayıları kullanılarak hesaplanmıştır.

$$M_{r_{mx}}(x_1, x_2, x_3, f_{ck}, f_{yk}) = 0.85 f_{cd} \left( \frac{1000}{x_3} * \frac{x_2^2 \pi}{4} \right) c(x) k_1 \left( x_1 - c_o - \frac{c(x) k_1}{2} \right) \quad (10)$$

$$g_5(x) = M_d(x_1) - M_r(x_1, x_2, x_3, x_4, f_{ck}, f_{yk}) \leq 0 \quad (11)$$

Deplasman hesabında dört tarafından kirşlere oturan dikdörtgen plaklar için katsayılar kullanılmıştır. Tüm kullanım yüklerinden (G+Q) oluşan ani sehim değeri ( $\delta_i$ ), Tablo 3'de verilen katsayılar aracılığıyla Denklem (12) kullanılarak hesaplanmıştır. Ani sehime bağlı olarak da TS500'e göre zamana bağlı sehim ( $\delta_{ig}\lambda$ ), hesaplanmıştır [14].

$$\delta_i = \frac{PL_k^4}{12kE_c I_{ef}} \quad (12)$$

$$I_{ef} = \left( \frac{M_{cr}}{M_{max}} \right)^3 I_c + \left[ 1 - \left( \frac{M_{cr}}{M_{max}} \right)^3 \right] I_{cr} \quad (13)$$








$$M_{cr} = 2.5 f_{ctd} \frac{I_c}{y} \quad (14)$$

$$\delta_t = \delta_i + \delta_{ig} \lambda \quad (15)$$

$$\lambda = \frac{\gamma_t}{1 + 50\rho'} \quad (16)$$

$$\delta_{ig} = \delta_i \frac{\sum \text{KalıcıYükler}}{\sum \text{TümYükler}} \quad (17)$$

**Tablo 3.** Dört tarafından kirişlere oturan dikdörtgen plaklar için ani sehim katsayıları ( $k$ ) [15]

$m = L_u/L_k$	1.0	1.10	1.20	1.30	1.40	1.50	1.60	1.70	1.80	1.90	2.0
1. 	65.79	55.25	48.31	43.48	40.32	37.88	36.1	34.84	34.01	33.33	32.89
2. 	53.19	42.37	35.21	30.4	26.95	24.45	22.62	21.23	20.2	19.31	18.55
3. 	39.68	33.11	28.74	25.71	23.47	21.79	20.66	19.69	18.9	18.28	17.79
4. 	43.48	39.84	37.45	35.71	34.6	33.67	33.11	32.57	32.36	32.15	31.95
5. 	43.48	33	26.11	21.41	18.08	15.65	13.85	12.47	11.4	10.54	9.87
6. 	29.94	23.7	19.53	16.61	14.51	12.94	11.74	10.8	10.06	9.47	8.99
7. 	20.53	17.12	14.75	13.04	11.76	10.79	10.03	9.43	8.94	8.55	8.23

Denklem (12) de yer alan Efektif atalet momenti ve kesitin çatlama momenti Denklem (13) ve (14)'de verilmiştir.

Betonarme yapılarda sünme ve büzülme etkisi ile oluşan zamana bağlı ek sehimlerin hesabı, genel olarak TS500-Madde 3.3.4 de verilen değişkenlere göre hesaplanmalıdır, zamana bağlı sehimleri de içeren toplam sehim ise Denklem (15)'ten hesaplanır [14].

Denklem (15) ile  $\lambda$  değeri yine TS500-Çizelge 13.2'den alınan kalıcı yük süre katsayısı,  $\gamma_t = 2$  alınarak hesaplanmıştır. Kısıtlamadaki sınır değer ise  $L_k/240$  (TS500-Çizelge 13.3) alınmıştır.

### 3. Bulgular ve Tartışma

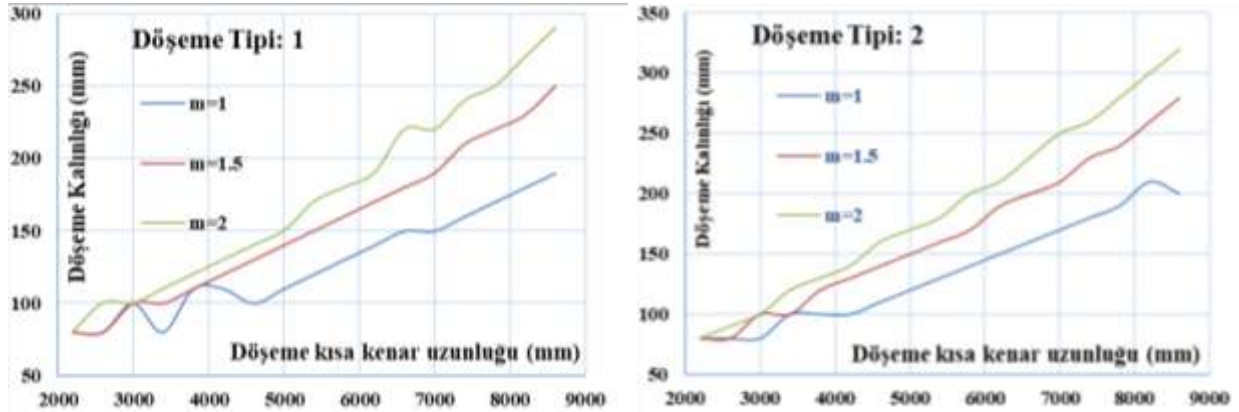
Bu optimizasyon işleminde 7 farklı döşeme tipi (Şekil ) ve her tipteki 11 farklı  $m = L_u/L_k$  değeri için analiz yapılmıştır. Tablo 4. 7'nolu döşeme tipi ve  $m = \{1, 1.5, 2\}$  değerleri için döşeme optimizasyonu sonucu optimal döşeme kalınlığı ( $h$ ) ile donatı çap ve aralıkları'de örnek olması açısından 7'nolu döşeme tipi ve  $m = \{1, 1.5, 2\}$ ,  $L_k = \{3000, 5000, 7000, 8600\}$  değerleri için döşeme optimizasyonu sonucu optimal döşeme kalınlığı ( $h$ ) ile donatı çap ve aralıkları verilmiştir. Burada donatı çap tercihinin de optimal tasarıma etkisi olduğu görülmektedir. Örneğin  $L_x = 7000$  mm ve  $m = 1.5$  için  $h = 280$  mm, x yönü açıklık donatı aralığı 130 mm ve  $\phi = 12$  mm, y yönü açıklık donatı aralığı 130 mm ve  $\phi = 10$  mm ile optimal tasarım tercihleri çözüm olarak elde edilmiştir.

Tablo 4'deki döşemedeki toplam deplasman ile sınır deplasman değerleri incelendiğinde yaklaşık  $L_k = 5-5.5$  m'yi aşan değerlerde döşeme kalınlığını belirleyici unsur taşıma gücü kriterleri değil deplasman kriterleri olduğu görülmektedir.

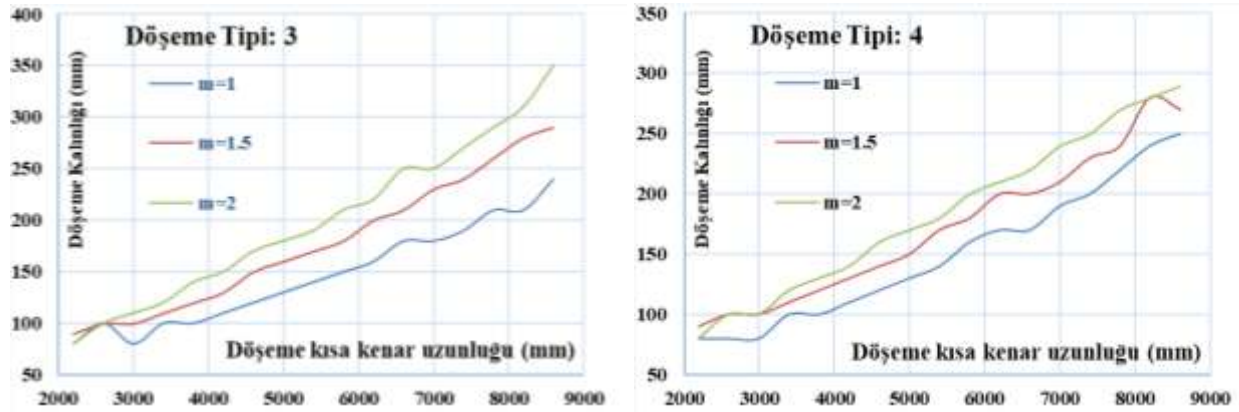
Şekil 5 ve Şekil 6 grafikleri hareketli yükün en kritik olduğu  $5 \text{ kN/m}^2$  olduğu durumdaki analizler için verilmiştir. Grafiklerde de görüldüğü üzere optimal döşeme kalınlığı arasında doğrusal varsayılabilir bir ilişki vardır. Optimizasyonda bazı donatı tercihleri doğrusal eğilimde bir miktar dalgalanmaya neden olduğu görülmüştür. Ayrıca döşeme tipi ve  $m=Lu/Lk$  değeri eğimin artmasına neden olmaktadır.

**Tablo 4.** 7' nolu döşeme tipi ve  $m = \{1, 1.5, 2\}$  değerleri için döşeme optimizasyonu sonucu optimal döşeme kalınlığı (h) ile donatı çap ve aralıkları

Model No	Lk (mm)	m	q (kN/m <sup>2</sup> )	h (mm)	Xm Aralığı (mm)	Xm $\phi$ (mm)	Ym Aralığı (mm)	Ym $\phi$ (mm)	Toplam Deplasman (mm)	Deplasman Sınırı (mm)
5052	3000	1.0	5.0	100	150	8	150	8	3.08	12.50
5057	5000	1.0	5.0	150	110	8	160	10	14.35	20.83
5062	7000	1.0	5.0	220	120	10	110	10	24.59	29.17
5066	8600	1.0	5.0	280	60	8	90	10	35.02	35.83
5137	3000	1.5	5.0	110	160	8	160	8	6.21	12.50
5142	5000	1.5	5.0	190	90	8	130	8	15.91	20.83
5147	7000	1.5	5.0	280	130	12	130	10	28.77	29.17
5151	8600	1.5	5.0	370	70	10	150	12	34.30	35.83
5222	3000	2.0	5.0	120	150	8	180	8	5.71	12.50
5227	5000	2.0	5.0	200	190	12	130	8	19.35	20.83
5232	7000	2.0	5.0	310	90	10	210	12	25.87	29.17
5236	8600	2.0	5.0	420	70	10	160	12	27.02	35.83



**Şekil 5.** Döşeme tipi 1, 2 ve  $m = \{1, 1.5, 2\}$  değerleri altında değişen Lk değerleri için optimal döşeme kalınlıkları grafiği



**Şekil 6.** Döşeme tipi 3, 4 ve  $m = \{1, 1.5, 2\}$  değerleri altında değişen Lk değerleri için optimal döşeme kalınlıkları grafiği

**Tablo 5.** Döşeme optimizasyonu sonucu optimal döşeme kalınlığı (h) değerleri (burada 5236 modelden bazıları verilmiştir)

Lk	Döşeme Tipi	q=1.5 kN/m <sup>2</sup>			q=2 kN/m <sup>2</sup>			q=3.5 kN/m <sup>2</sup>			q=5 kN/m <sup>2</sup>		
		m=1	m=1.5	m=2	m=1	m=1.5	m=2	m=1	m=1.5	m=2	m=1	m=1.5	m=2
3000	1	80	80	100	80	80	90	80	80	100	100	100	100
	2	80	90	100	80	90	100	80	100	100	80	100	100
	3	80	100	110	100	100	110	80	100	110	80	100	110
	4	80	100	100	80	100	100	80	100	100	80	100	100
	5	90	100	120	80	100	120	80	100	120	80	100	120
	6	100	110	120	90	110	120	100	110	120	110	120	120
	7	100	110	120	100	110	120	100	110	120	100	110	120
3800	1	100	110	120	100	110	120	100	110	120	110	110	120
	2	90	120	130	100	120	130	100	120	130	100	120	130
	3	100	120	140	100	120	140	100	120	140	100	120	140
	4	100	120	130	100	120	130	100	120	130	100	120	130
	5	100	130	140	100	130	140	100	130	140	100	130	140
	6	110	130	150	110	130	150	110	130	150	110	130	150
	7	110	140	160	110	140	160	110	140	160	110	140	160
4600	1	100	130	140	100	130	140	100	130	140	100	130	140
	2	110	140	160	110	140	160	110	140	160	110	140	160
	3	120	150	170	120	150	170	120	150	170	120	150	170
	4	120	140	160	120	140	160	120	140	160	120	140	160
	5	120	150	170	120	150	170	120	150	170	120	150	170
	6	130	160	180	130	160	180	130	160	180	130	160	180
	7	140	170	190	140	170	190	140	170	190	140	170	190
5400	1	120	150	170	120	150	170	120	150	170	120	150	170
	2	130	160	180	130	160	180	130	160	180	130	160	180
	3	140	170	190	140	170	190	140	170	190	140	170	190
	4	140	170	180	140	170	180	140	170	180	140	170	180
	5	140	180	200	140	180	200	140	180	200	140	190	210
	6	150	190	210	150	190	210	150	190	210	150	190	210
	7	160	200	220	160	200	220	160	200	220	160	210	220
6200	1	140	170	190	140	170	190	140	170	190	140	170	190
	2	150	190	210	150	190	210	150	190	210	150	190	210
	3	160	200	220	160	200	220	160	200	220	160	200	220
	4	160	190	210	160	190	210	160	190	210	170	200	210
	5	160	200	230	160	210	230	160	210	240	160	230	250
	6	170	210	240	170	210	240	170	210	240	170	230	250
	7	180	220	250	180	220	250	180	230	250	190	240	260
7000	1	150	190	210	150	190	210	150	190	220	150	190	220
	2	170	210	240	170	210	240	170	220	240	170	210	250
	3	180	220	250	180	220	250	180	220	250	180	230	250
	4	180	210	240	180	210	240	190	210	240	190	210	240
	5	180	230	260	180	240	260	180	250	280	190	260	310
	6	190	240	270	190	240	270	190	240	270	200	260	290
	7	200	250	280	200	250	280	210	270	290	220	280	310
7800	1	170	210	240	170	210	240	170	210	240	170	220	250
	2	190	230	260	190	230	260	190	230	260	190	240	280
	3	200	250	280	200	250	280	200	250	280	210	260	290
	4	200	240	260	200	240	260	200	240	260	220	240	270
	5	200	260	290	200	260	300	200	280	320	210	300	340
	6	210	270	300	210	270	300	250	280	320	230	310	330
	7	230	280	320	230	290	320	240	310	340	250	330	360
8600	1	190	230	260	190	230	260	190	230	270	190	250	290
	2	200	260	290	200	260	290	200	260	300	200	280	320
	3	220	270	310	220	270	310	220	280	310	240	290	350
	4	220	260	290	230	270	290	220	270	290	250	270	290
	5	220	300	340	220	300	350	220	320	390	240	340	390
	6	240	290	340	250	300	340	250	320	360	250	340	380
	7	250	320	350	250	330	360	270	350	390	280	370	420

Tablo 5’de 5236 optimizasyonu çalışılan modellerden bazıları ( $m = [1, 1.5, 2]$ ) seçilerek farklı yükleme durumları ve döşeme tipleri için optimal döşeme kalınlığı (h) değerleri verilmiştir. Tablo

6’da ise tüm döşeme tipleri için m değerlerine göre ortalama Lk/h değerleri verilmiştir. Bu tabloda verilen Lk/h, minimum döşeme kalınlığı kriteri değil, deplasman kontrolü gerektirmeyen güvenilirlik tabanlı optimal tasarım kriteridir. Tablo 7’de ise, 7 Numaralı döşeme tipi (nadiren karşılaşılan dört kenarı süreksiz döşeme) için en kritik Lk/h değerleri verilmiştir.

**Tablo 6.** Tüm döşeme tipleri için m değerlerine göre ortalama Lk/h değerleri

Lk (mm)	m											Ortalama Lk/h
	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0	
3000	34.7	34.0	31.7	31.2	30.3	30.0	29.2	28.7	28.0	27.8	27.6	30.3
3400	35.6	34.2	33.1	32.9	31.5	31.1	29.9	29.3	29.0	27.8	27.2	31.1
3800	36.8	35.5	34.7	33.0	32.3	30.7	30.0	29.8	28.8	27.9	27.6	31.6
4200	38.1	36.5	35.4	33.1	32.5	31.1	30.5	29.6	28.5	28.2	28.0	32.0
4600	38.7	36.0	35.6	33.0	32.8	31.1	30.6	29.5	28.9	28.3	27.7	32.0
5000	38.8	36.0	35.8	33.4	32.3	31.6	30.3	29.7	28.9	28.4	28.0	32.1
5400	38.9	36.2	35.2	33.5	32.4	31.1	30.4	29.7	28.9	28.4	28.1	32.1
5800	38.7	35.8	35.2	33.4	32.5	31.4	30.5	29.7	28.9	28.3	27.6	32.0
6200	38.8	36.5	34.9	33.6	32.1	30.9	30.4	29.3	29.1	28.0	28.0	32.0
6600	38.8	36.7	34.6	33.3	32.3	31.1	30.6	29.3	28.7	28.3	27.5	31.9
7000	38.9	36.6	34.6	33.3	32.3	31.0	29.9	29.4	28.5	28.1	27.5	31.8
7400	38.8	36.5	34.4	32.9	31.9	30.5	29.8	29.2	28.4	27.8	27.4	31.6
7800	38.3	36.0	34.0	32.6	31.6	30.5	29.4	29.0	28.2	27.7	27.2	31.3
8200	38.4	36.0	34.1	32.2	31.4	30.3	29.3	28.6	28.1	27.4	27.0	31.2
8600	38.3	35.6	33.7	32.1	30.6	30.0	29.0	28.3	27.7	27.1	26.5	30.8
<b>Ortalama Lk/h→</b>	<b>37.1</b>	<b>35.1</b>	<b>33.8</b>	<b>32.4</b>	<b>31.6</b>	<b>30.5</b>	<b>29.7</b>	<b>29.0</b>	<b>28.3</b>	<b>27.8</b>	<b>27.3</b>	<b>31.1</b>

**Tablo 7.** 7 Numaralı döşeme tipi (dört kenarı süreksiz) için en kritik Lk/h değerleri

Lk (mm)	m											Minimum Lk/h
	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0	
3000	27.3	30.0	30.0	30.0	27.3	25.0	27.3	25.0	25.0	25.0	25.0	25.0
3400	34.0	30.9	30.9	28.3	28.3	28.3	26.2	26.2	24.3	24.3	24.3	24.3
3800	34.5	31.7	31.7	29.2	29.2	27.1	27.1	25.3	25.3	25.3	23.8	23.8
4200	32.3	32.3	30.0	30.0	28.0	28.0	26.3	26.3	24.7	24.7	24.7	24.7
4600	32.9	32.9	30.7	28.8	28.8	27.1	27.1	25.6	25.6	24.2	24.2	24.2
5000	33.3	31.3	29.4	29.4	27.8	26.3	26.3	25.0	25.0	25.0	25.0	25.0
5400	33.8	31.8	30.0	28.4	27.0	25.7	25.7	25.7	24.5	24.5	24.5	24.5
5800	34.1	29.0	29.0	27.6	26.4	26.4	25.2	25.2	24.2	24.2	24.2	24.2
6200	32.6	31.0	28.2	27.0	25.8	25.8	23.8	23.8	23.8	23.8	23.8	23.8
6600	33.0	30.0	28.7	26.4	26.4	25.4	24.4	23.6	23.6	23.6	22.8	22.8
7000	31.8	29.2	28.0	26.9	25.9	25.0	24.1	23.3	23.3	22.6	22.6	22.6
7400	32.2	29.6	27.4	26.4	25.5	23.1	23.9	23.1	22.4	22.4	22.4	22.4
7800	31.2	28.9	26.9	26.0	24.4	23.6	22.9	22.9	22.3	22.3	21.7	21.7
8200	30.4	28.3	26.5	25.6	24.1	23.4	22.2	21.6	22.2	21.6	21.6	21.6
8600	30.7	27.7	26.1	25.3	23.2	23.2	22.6	22.1	21.5	21.0	20.5	20.5
<b>Genel Minimum</b>	<b>26.0</b>	<b>23.6</b>	<b>26.0</b>	<b>25.3</b>	<b>23.2</b>	<b>23.1</b>	<b>22.2</b>	<b>21.6</b>	<b>21.5</b>	<b>21.0</b>	<b>20.5</b>	<b>20.5</b>

#### 4. Sonuç ve Öneriler

Bu optimizasyon çalışmasında ayrık PSO algoritması ile farklı boyutlarda, farklı hareketli yükler altında ve farklı döşeme tipleri için döşeme kalınlığı ve donatı tercihlerine ait en uygun değerler

güvenilirlik ilkelerini, deplasman kısıtını ve diğer TS500 kurallarını sağlayacak şekilde saptanmaya çalışılmıştır. Şu sonuçlara varılmıştır:

- Optimizasyon sonucuna göre belirli boyuta kadar TS500 minimum döşeme kalınlığı hem optimal döşeme kalınlığı hem de deplasman kriterlerini sağlayan kalınlık olarak elde edilmiştir.
- Hareketli yükün  $1.5 \text{ kN/m}^2$  (çatı kat döşemesi gibi) ve  $2 \text{ kN/m}^2$  olduğu durumda  $L_k=7 \text{ m}$  'ye kadar minimum döşeme kalınlığı optimal kalınlık olmaktadır.  $L_k=7 \text{ m}$  ve üzeri döşemelerde ise deplasman kontrolü gerektirmeyecek optimal döşeme kalınlığının yaklaşık  $h=L_k/29$  seçilmesi uygundur. Bu değer 7 numaralı döşeme tipi haricindeki döşeme tiplerinin ortalamasıdır.
- Hareketli yükün  $3.5 \text{ kN/m}^2$  olduğu durumda  $L_k=6 \text{ m}$  'ye kadar minimum döşeme kalınlığı optimal kalınlık olmaktadır.  $L_k=6 \text{ m}$  ve üzeri döşemelerde ise deplasman kontrolü gerektirmeyecek optimal döşeme kalınlığının yaklaşık  $h=L_k/26.5$  seçilmesi uygundur. Bu değer 7 numaralı döşeme tipi haricindeki döşeme tiplerinin ortalamasıdır.
- Hareketli yükün  $5 \text{ kN/m}^2$  olduğu durumda  $L_k=5 \text{ m}$ 'ye kadar minimum döşeme kalınlığı optimal kalınlık olmaktadır.  $L_k=5-7 \text{ m}$  döşemelerde ise deplasman kontrolü gerektirmeyecek optimal döşeme kalınlığının yaklaşık  $h=L_k/26.5$ ,  $L_k=7-8.5 \text{ m}$  döşemelerde ise deplasman kontrolü gerektirmeyecek optimal döşeme kalınlığının yaklaşık  $h=L_k/25$  seçilmesi uygundur. Bu değer 7 numaralı döşeme tipi haricindeki döşeme tiplerinin ortalamasıdır.
- Donatı tercihleri aşamasında, farklı donatı çapları ile çalışmak daha ekonomik çözümleri sağlayabilmektedir.
- Kısıtlar altında ayrık (sürekli) parçacık sürü optimizasyon algoritmasının 9 boyutlu döşeme optimizasyon probleminde başarılı şekilde kullanılabileceği gözlenmiştir.

### Yazarların Katkıları

MKD çalışma kapsamında gerçekleştirilen deneysel prosedürlerin ve aygıt tasarımlarının tümünü gerçekleştirmiştir. Makalenin son halini okumuş ve onaylamıştır.

### Çıkar Çatışması

Yazarlar, çıkar çatışması olmadığını beyan eder.

### Kaynaklar

- [1]. Okwu, M. O., Tartibu, L. K., Metaheuristic Optimization : Nature-Inspired Algorithms Swarm and Computational Intelligence, Theory and Applications, Switzerland, 2021.
- [2]. Aldwaik, M., Adeli, H., Advances in optimization of highrise building structures, Structural and Multidisciplinary Optimization, 2014, 50: 899–919.
- [3]. Santos, J., Henriques, A. A., Span-to-depth ratio limits for RC continuous beams and slabs based on MC2010 and EC2 ductility and deflection requirements, Engineering Structures, 2021, 228, 111565.
- [4]. Yang, X. S., Introduction to Algorithms for Data Mining and Machine Learning, Elsevier Inc., 2019.
- [5]. Vasuki, A., Nature-Inspired Optimization Algorithms, Taylor & Francis, Oxford, 2020.
- [6]. Yang, X. S., He, X. S., Mathematical Foundations of Algorithms, Springer, Switzerland, 2019.
- [7]. Yang, X. S., Mathematical Analysis of Nature-Inspired Algorithms, in: X.S. Yang (Ed.), Nature-Inspired Algorithms and Applied Optimization, Springer, London, 2018.
- [8]. Eberhart, R., Kennedy, J., New optimizer using particle swarm theory, MHS'95 Proceedings of the International Symposium on Micro Machine and Human Science, 1995, 39–43.

- [9]. Dorigo, M., Oca, M. A. M., Engelbrecht, A., Particle swarm optimization, Scholarpedia, 2008, 3:1486.
- [10]. Li, L., Liu, F., Group Search Optimization for Applications in Structural Design, Springer International Publishing, Berlin, 2011.
- [11]. ElHami, A., Radi, B., Uncertainty and optimization in structural mechanics, John Wiley & Sons, Inc., London, 2013.
- [12]. Ersoy, U., Özcebe, G., Canbay, E., Betonarme : Davranış ve Hesap İlkeleri, 9th ed., İstanbul, 2019.
- [13]. TSE, TS 498 Yapi Elemanlarının Boyutlandırılmasında Alınacak Yüklerin Hesap Değerleri, Ankara, 1997.
- [14]. TSE, TS 500 Betonarme Yapıların Tasarım ve Yapım Kuralları, Ankara, 2000.
- [15]. Topçu A., Betonarme I-II, Eskişehir Osmangazi Üniversitesi, Eskişehir, 2019. [http://mmf2.ogu.edu.tr/atopcu/index\\_dosyalar/Dersler](http://mmf2.ogu.edu.tr/atopcu/index_dosyalar/Dersler).
- [16]. Aldwaik, M., Adeli, H., Cost optimization of reinforced concrete flat slabs of arbitrary configuration in irregular highrise building structures, Structural and Multidisciplinary Optimization. 2016, 54: 151–164.
- [17]. Ghandi E., Shokrollahi N., Nasrolahi M., Optimum Cost Design of Reinforced Concrete Slabs Using Cuckoo Search Optimization Algorithm, Engineering Applications of Artificial Intelligence. 2017, 7: 539–564.
- [18]. Fraile-Garcia, E., Ferreiro-Cabello, J., Martinez-Camara, E., Jimenez-Macias, E., Optimization based on life cycle analysis for reinforced concrete structures with one-way slabs, Engineering Structures. 2016, 109: 126–138.
- [19]. Ahmadi-Nedushan, B., Varae, H., Minimum cost design of concrete slabs using particle swarm optimization with time varying acceleration coefficients, World Applied Sciences Journal. 2011, 13: 2484–2494.
- [20]. Karaboğa, D., Yapay Zeka Optimizasyon Algoritmaları, 3rd ed., Nobel, Ankara, 2014.
- [21]. Yang, X. S., Nature-Inspired Optimization Algorithms, Elsevier, London, 2014.