



## Comparison of multi-objective and single-objective approaches in feasibility enhanced particle swarm optimization

Mehmet Sinan Hasanoğlu<sup>1\*</sup>, Melik Dölen<sup>2</sup>

<sup>1</sup>Mechanical Engineering Department, Middle East Technical University, Ankara, 06800, Turkey

<sup>2</sup>TUBITAK SAGE, Ankara, 06261, Turkey

### Highlights:

- Multi-objective and single-objective feasibility enhanced particle swarm optimization
- Pareto optimality based multi-objective optimization approach
- Fixed weight linearly aggregated objective functions

### Keywords:

- Multi-objective optimization
- Constrained optimization
- Particle swarm optimization
- Pareto front
- fixed weight linearly aggregated objective function

### Article Info:

Research Article  
Received: 27.06.2019  
Accepted: 21.09.2019

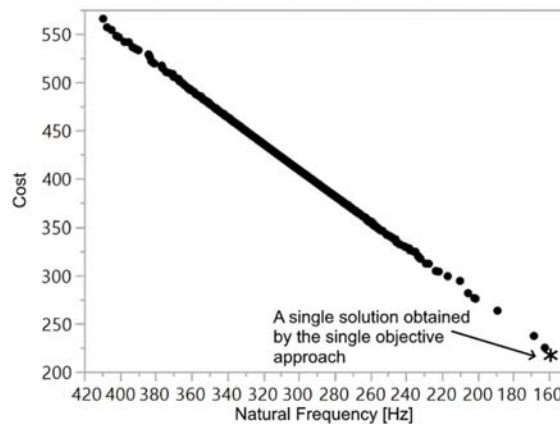
### DOI:

10.17341/gazimmfd.437579

### Correspondence:

Author: Mehmet Sinan Hasanoğlu  
e-mail: sinan.hasanoglu@tubitak.gov.tr  
phone: +90 312 590 93 04

### Graphical/Tabular Abstract



**Figure A.** Solutions obtained by the multi-objective approach provides design alternatives in the solution domain whereas the single objective approach gives a single solution for the given criteria

**Purpose:** Comparison of a Pareto based multi-objective feasibility enhanced particle swarm optimization approach with fixed weight linearly aggregated single-objective optimization using a variant of the same algorithm. Evaluation of the advantages presented by the multi-objective approach.

### Theory and Methods:

Solutions for three separate multi-objective constrained problems were obtained using two different approaches: Pareto based multi-objective feasibility enhanced particle swarm optimization Single-objective feasibility enhanced particle swarm optimization using fixed weight linearly aggregated single-objective variants of the problems.

### Results:

Comparisons involving three problems (two of which were highly constrained) revealed that optimizations performed using the multi-objective approach resulted in solutions that were also suitable for all single-objective criteria. Results obtained by a single multi-objective optimization run were found to be at least as good as results obtained by separate aggregated-objective runs for each aggregated criterion.

### Conclusion:

In Pareto optimality based optimization algorithms a Pareto front consisting of many non-dominated solutions are found in each run instead of a single solution. Therefore: The designer can choose from these non-dominated solutions with a clear trade-off between known values of all objectives. This approach allows performing the trade-off without requiring rerunning the optimization with new objective weights. If efficient Pareto optimality based multi-objective optimization algorithms are available, it is unnecessary to employ single-objective approaches by combining the objectives with presumably biased weights assigned with insufficient information related with the problem.



## Olurluğu artırılmış parçacık sürü optimizasyonunda çok amaçlı ve tek amaçlı yaklaşımların kıyaslanması

Mehmet Sinan Hasanoğlu<sup>1\*</sup>, Melik Dölen<sup>2</sup>

<sup>1</sup>Orta Doğu Teknik Üniversitesi, Mühendislik Fakültesi, Makina Mühendisliği Bölümü, Üniversiteler Mahallesi, Dumlupınar Bulvarı No:1 06800 Çankaya Ankara, Türkiye

<sup>2</sup>TÜBİTAK, SAGE, Pk. 16 Mamak 06261 Ankara, Türkiye

### Ö N E Ç İ K A N L A R

- Çok amaçlı ve tek amaçlı olurluğu artırılmış parçacık sürü optimizasyonu
- Pareto tabanlı çok amaçlı eniyileme yaklaşımı
- Sabit ağırlıklı doğrusal fonksiyonlar ile birleştirilmiş amaç fonksiyonları

### Makale Bilgileri

Araştırma Makalesi

Geliş: 27.06.2019

Kabul: 21.09.2019

DOI:

10.17341/gazimmfd.437579

Anahtar Kelimeler:

Çok amaçlı eniyileme,  
kısıtlı eniyileme,  
parçacık sürü optimizasyonu,  
pareto cephesi,  
sabit ağırlıklı doğrusal amaç  
fonksiyonu

### ÖZET

Bu çalışmada çok amaçlı ve kısıtlı eniyileme problemleri için Pareto esasına dayalı çok amaçlı olurluğu artırılmış parçacık sürü optimizasyonu (MOFEPSO) elde edilen çözümler ile bu problemlerin sabit ağırlıklı doğrusal fonksiyonlarla tek amaca indirgenen halleri için tek amaçlı eniyileme yaklaşımıyla bulunan çözümler karşılaştırılmıştır. İki aşırı kısıtlı olmak üzere toplam üç farklı problem için gerçekleştirilen karşılaştırmalarda amaç fonksiyonlarına önceden bir ağırlık verilmeksizin çok amaçlı bir yaklaşımla gerçekleştirilen eniyilemenin incelenen tüm tek amaçlı kısıtlar için de uygun sonuçlar verebildiği görülmüştür. Çok amaçlı yaklaşımla amaç fonksiyonlarına sonradan (eniyileme sonrasında) ağırlık verilebilir ve eniyileme koşulunun tekrarlanmasına gerek olmadan ağırlıklar değiştirilebilir.

## Comparison of multi-objective and single-objective approaches in feasibility enhanced particle swarm optimization

### H I G H L I G H T S

- Multi-objective and single-objective feasibility enhanced particle swarm optimization
- Pareto optimality based multi-objective optimization approach
- Fixed weight linearly aggregated objective functions

### Article Info

Research Article

Received: 27.06.2019

Accepted: 21.09.2019

DOI:

10.17341/gazimmfd.437579

Keywords:

Multi-objective optimization,  
constrained optimization,  
particle swarm optimization,  
pareto front,  
fixed weight linearly  
aggregated objective  
function

### ABSTRACT

In this study, solutions for multi-objective constrained problems and their fixed weight linearly aggregated single-objective variants were obtained using the Pareto based multi-objective feasibility enhanced particle swarm optimization and single-objective approaches respectively. Comparisons involving three problems (two of which were highly constrained) revealed that optimizations performed using the multi-objective approach resulted in solutions that were also suitable for all single-objective criteria. With the multi-objective approach, objectives can be weighted after the optimization run and trade-offs can be performed without repeating the run.

\*Sorumlu Yazar/Corresponding Author: sinan.hasanoğlu@tubitak.gov.tr, dolen@metu.edu.tr / Tel: +90 312 590 93 04

## 1. GİRİŞ (INTRODUCTION)

Eniyileme (optimizasyon) bilim ve mühendislik dallarının hemen hepsinde sıkça kullanılır. Eniyileme, teorik olarak, bir problem için en iyi (ya da mümkün olan en iyi) çözümü bulma olarak tanımlanabilir. Matematiksel yaklaşımlar ideal çözümün özellikleri ve bu çözüme ulaşmak için uygulanan yöntemlere odaklanırken mühendislik yaklaşımları işlem gücü ve zaman kısıtlarından dolayı yaklaşık çözüm yöntemlerini de kullanır. Eniyileme problemi bir ya da daha fazla fonksiyonun en düşük değer(ler)inin (veya özel durumlarda en yüksek değerinin) bulunması olarak ifade edilir. Bu fonksiyonlara amaç fonksiyonları adı verilir. Tek bir amaç fonksiyonu bulunan problemler “tek amaçlı” olarak sınıflandırılırken, birden fazla amaç fonksiyonu olan problemlere ise “çok amaçlı” eniyileme problemleri denir. Çok amaçlı eniyileme problemleri sabit ağırlıklı doğrusal fonksiyonlar yardımıyla tek amaçlı problemlere dönüştürülerek tek amaçlı eniyileme yaklaşımlarıyla çözülebilirler. Ancak, bu durumda amaç fonksiyonlarının eniyileme öncesinde önem sırasının belirlenerek ağırlıkların tanımlanması gerekir. Çok amaçlı yaklaşımlarda ise buna gerek kalmadan tüm amaç fonksiyonları aynı anda eniyilenebilir. Ne var ki bu durumda da eniyilenen fonksiyonların tamamı tam olarak aynı noktada en iyi değerlerini almıyorsa tek bir en iyi noktadan bahsetmek mümkün değildir. Dolayısıyla tek amaçlı eniyilemede geçerli olan skaler “en iyi” tanımı çok amaçlı yaklaşımlar için geçerli değildir. Bunun yerine içinden ödünleşme ile bir çözümün seçilebileceği “Pareto en iyi” çözümler kümesi (Pareto kümesi veya Pareto cephesi olarak da anılır) kullanılır [1]. Pareto kümesi diğer hiçbir çözüm tarafından baskın gelinemeyen (basılın olmayan) çözümlerden oluşur. Bir çözüm diğer bir çözüme ancak ve ancak tüm amaç fonksiyonu değerleri diğeri kadar iyi ve en az bir amaç fonksiyonu diğeri ondan daha iyi ise baskındır.

Pratikte tasarım problemlerinin önemli bir kısmı çok amaçlı olduğu gibi aynı zamanda kısıtlıdır. Kısıtlı problemler karar vektörünü oluşturan tasarım parametreleri üzerinde bazı kısıtların tanımlandığı problemlerdir. Özellikle aşırı kısıtlı problemlerde kısıtlara uyan bir çözüm bulmak en iyi çözümü bulmak kadar zor olabilir. Evrimsel algoritmalar, kısıtlı ve kısıtsız eniyileme problemlerinin çözümünde yaygın olarak kullanılmaktadır. Çok amaçlı evrimsel algoritmalar ile ilgili kapsamlı bir derleme Zhou ve diğeri [1] tarafından kaleme alınmıştır. Çok amaçlı yaklaşımlarda kısıtların ele alınması ile ilgili pek çok yöntem bulunmaktadır. Bu yöntemler ile ilgili Coello [2] ve Mezura-Montes ve Coello [3] tarafından derlenen çalışmalara başvurulabilir.

Bu çalışmada, kısıtlı problemlerde çok amaçlı ve tek amaçlı yaklaşımların karşılaştırılması için parçacık sürüsü optimizasyonuna (PSO) dayanan FEPSO [4] algoritması ile bu algoritmanın çok amaçlı problemler için geliştirilmiş hali olan MOFEPSO [5] algoritması kullanılmıştır. Tek amaçlı ve çok amaçlı yaklaşımların kıyaslandığı bu çalışmada

FEPSO tabanlı algoritmalar hem tek amaçlı hem çok amaçlı problemleri ele alabilen sürülerinin bulunması ve bu tip problemler için başarımlarının gösterilmiş olması nedeniyle tercih edilmiştir. PSO, Eberhart ve Kennedy [6] tarafından geliştirilip Bratton ve Kennedy [7] tarafından standart hale getirilen bir evrimsel eniyileme algoritmasıdır. Çok amaçlı problemlere ilk defa Ray, Tai ve Seow [8] tarafından uygulanan PSO, pek çok tek amaçlı ve çok amaçlı problem için tatmin edici çözümlerin görece az bir hesaplama maliyeti ile bulunmasında kullanılmıştır [9–12]. Her ne kadar ilk hali kısıtsız problemlerde kullanılmak üzere geliştirilmiş olsa da PSO kısıtlı problemlerin çözümünde kullanılmak üzere uyarlanmıştır. PSO'nun kısıtlı problemlerde kullanımı genellikle Deb yaklaşımı [13] uygulanarak gerçekleştirilmişse de literatürde pek çok farklı yaklaşım bulunmaktadır [14].

PSO'da parçacıklar çok boyutlu karar uzayında hareket ederler. Parçacıklar konumlarını deneyimlerine (önceden deneyimledikleri konumlar içinden en iyi olana) ve sürüdeki diğer parçacıkların durumlarına göre değiştirirler [15]. Parçacık hareketi, her yinelemede (iterasyonda) parçacığın mevcut konumuna, hızına, deneyimlediği en iyi konuma ve sürü liderinin konumuna bağlı olarak gerçekleşir [15–17].

Pareto tabanlı çok amaçlı yaklaşımların diğer yaklaşımlarla kıyaslandığı bazı diğer çalışmalar da bulunmaktadır. Zitzler ve Thiele [18] tarafından Pareto tabanlı NSGA (İng. Non-dominated Sorting Genetic Algorithm) [19] algoritması ile sabit ağırlıklı fonksiyonlarla gerçekleştirilen çözümler kıyaslanmış ve incelenen problemler için Pareto tabanlı yaklaşımın üstünlüğü gösterilmiştir. Coello [20] tarafından sabit ağırlıklı fonksiyonlar ile kullanılan tek amaçlı yaklaşımların en önemli eksikliği amaçların ölçeklenmesi için kullanılacak ağırlıkların problem hakkında yeterince bilgi sahibi olunmaksızın belirlenmesi ve çözümün bu ağırlıklara bağımlı olması olarak belirtilmiştir. Ishibuchi vd. [21] ise tek amaçlı ve çok amaçlı eniyileme algoritmalarının kıyaslanması ile ilgili yöntemleri tartışmış ve gerçekleştirdikleri hesaplamalı deneylerde çok amaçlı yaklaşımlar ile daha iyi sonuçlar alındığını bildirmişlerdir. Bu çalışmada hem tek amaçlı problemlerde hem de çok amaçlı problemlerde başarımlı bilinen algoritmalar ile bu iki yaklaşımın kıyaslanması ve tek amaca indirgenerek gerçekleştirilen eniyilemelerde kullanılan ağırlıklar bakımından istenen sonucun elde edilmediğinin ortaya çıkarılması amaçlanmıştır. Bunun yanında belirlenen ağırlıkların tek amaçlı yaklaşımla gerçekleştirilen eniyilemelerde beklenen sonucu verip vermediği de araştırılmıştır.

Çalışmanın geri kalan bölümleri şu şekilde yapılandırılmıştır: Bu girişin ardından; ikinci kısmında MOFEPSO algoritması anlatılmaktadır. Üçüncü kısımda çok amaçlı ve tek amaçlı yaklaşımların karşılaştırılmasından elde edilen bulgulara ve tartışmaya yer verilirken; son kısımda da çalışmadan elde edilen sonuçlara yer verilmiştir.

**2. ÇOKLU AMAÇ İÇEREN OLURLUĞU GELİŞTİRİLMİŞ PARÇACIK SÜRÜ OPTİMİZASYONU (MULTI-OBJECTIVE FEASIBILITY ENHANCED PARTICLE SWARM OPTIMIZATION)**

Çoklu amaca (veya bir başka değişle değerlendirme ölçütüne) sahip eniyileme problemleri en genel haliyle (Eş. 1, Eş. 2 ve Eş. 3)

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) \equiv [f_1(\mathbf{x}) \quad f_2(\mathbf{x}) \quad \dots \quad f_k(\mathbf{x})]^T \quad (1)$$

$$g_m(\mathbf{x}) \leq 0, \quad m \in \{1, \dots, M\} \quad (2)$$

$$h_p(\mathbf{x}) = 0, \quad p \in \{1, \dots, P\} \quad (3)$$

şeklinde tanımlanır [22]. Eş. 2 ve Eş. 3'te verilen kısıtları ihlâl etmeden; Eş. 1'deki amaç fonksiyonu vektörünün en düşük değerini (veya özel durumlarda en yüksek değerini) sağlayacak bir  $\mathbf{x}$  vektörünün (tasarım parametresi uzayında) bulunması problemin özünü oluşturur. Burada  $K$  amaç fonksiyonu sayısını;  $\mathbf{x} = [x_1 \quad x_2 \quad \dots \quad x_N]^T \in \mathbb{R}^{N \times 1}$  karar vektörünü;  $M$  ve  $P$  sırasıyla eşitsizlik ve eşitlik kısıtlarının sayısını ifade etmektedir. Karar değişkenleri için sınırlar (Eş. 4)

$$x_n^L \leq x_n \leq x_n^U, \quad n \in \{1, \dots, N\} \quad (4)$$

şeklinde. Bu makalede ele alınan MOFEPSO isimli teknik sadece eşitsizlik şekline yazılabilen kısıtları kullanabilmektedir. Dikkat edilirse, Eş. 3'te verilen eşitlik kısıtlarının tümünü (Eş. 5)

$$|h_p(\mathbf{x})| - \varepsilon \leq 0 \quad (5)$$

şeklinde Eş. 2'deki gibi eşitsizlere dönüştürmek mümkündür. Burada  $\varepsilon$  kullanıcı tarafından seçilmiş (negatif olmayan) bir reel sayıyı belirtmektedir.

Hasanoğlu ve Dölen [5] tarafından önerilen MOFEPSO yöntemi şematik olarak Şekil 1'de gösterilmiştir. Detaylandırmak gerekirse, bu teknikte eniyileme süreci konum ve hız matrislerine rastgele ilk değerler vererek başlar. (Eş. 6, Eş. 7)

$$\mathbf{X} = [x_{i,n}] \in \mathbb{R}^{I \times N} \quad (6)$$

$$\mathbf{V} = [v_{i,n}] \in \mathbb{R}^{I \times N} \quad (7)$$

Burada  $i \in \{1, \dots, I\}$  ve  $n \in \{1, \dots, N\}$  olup;  $I$  sürüdeki parçacık sayısını,  $N$  ise karar değişkeni sayısını temsil etmektedir. Konum matrisi  $\mathbf{X}$ 'in her satırı (Eş. 8)

$$\mathbf{x}_{i,*} = [x_{i,1} \quad x_{i,2} \quad \dots \quad x_{i,N}] \quad (8)$$

şeklinde ifade edilebilir. Parçacık konumlarına ilk değer verilmesinin ardından (Eş. 9)

$$\mathbf{C} = [c_{i,m}] \in \mathbb{R}^{I \times M} \quad (9)$$

tanımlı kısıt matrisi (Eş. 10)

$$\mathbf{c}_{i,*} := [g_1(\mathbf{x}_{i,*}^T) \quad g_2(\mathbf{x}_{i,*}^T) \quad \dots \quad g_M(\mathbf{x}_{i,*}^T)] \quad (10)$$

gibi hesaplanır.

MOFEPSO genel (tüm sürü için) ve özel baskın çözüm kümeleri kullanır. Bu kümeler, tüm sürünün ulaştığı çözümler içinden ve her parçacığın bulduğu (özel/bireysel) çözümler içinden diğer çözümlerce baskın gelinemeyenlerin bir araya getirildiği kümelerdir. Genel baskın çözüm kümesinin elemanları karar ve amaç vektörü ikililerinden oluşmaktadır. Bu küme (Eş. 11)

$$\mathbf{B} = \{ \mathbf{b}_\phi = (\mathbf{b}_\phi^x, \mathbf{b}_\phi^y) \mid \phi \in \{1, \dots, \Phi\} \} \quad (11)$$

şeklinde tanımlanır. Burada  $\Phi$  kümedeki baskın çözüm sayısını yansıtmaktadır. Benzer şekilde, sürüdeki her parçacığın kendi baskın çözüm kümesi de mevcuttur: (Eş. 12)

$$\mathbf{D}_i = \{ \mathbf{d}_{i,\omega} = (\mathbf{d}_{i,\omega}^x, \mathbf{d}_{i,\omega}^y) \mid \omega \in \{1, \dots, \Omega_i\} \} \quad (12)$$

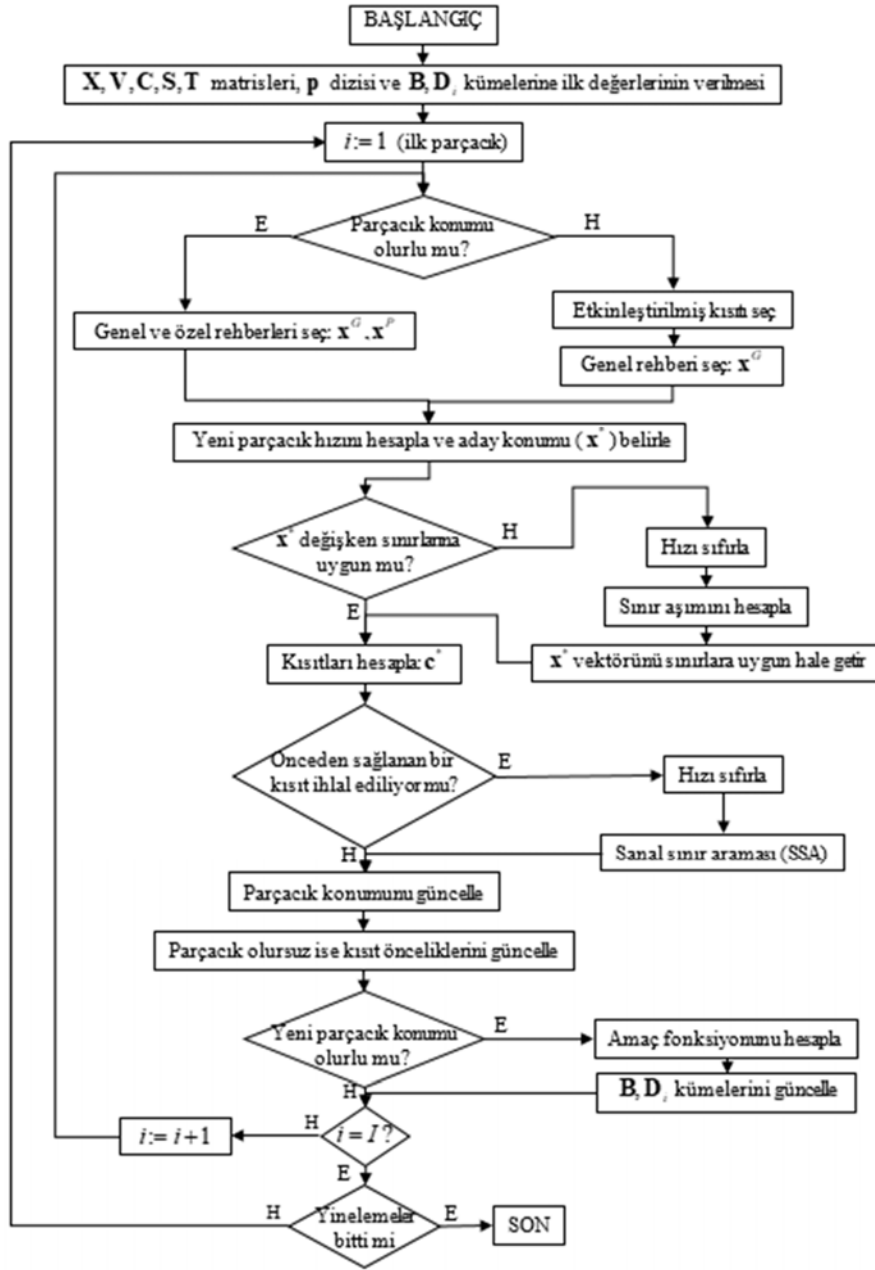
Burada  $\Omega_i$ ,  $i$  numaralı parçacığın baskın çözüm kümesinin eleman sayısıdır. Eğer  $i$  numaralı parçacık *olabilir*<sup>1</sup> bir çözümü yansıtmıyorsa,  $\mathbf{D}_i = \emptyset$  olur. Eğer sürü hiç olabilir bir çözüm bulamadıysa  $\mathbf{B} = \emptyset$  halini alır. Dikkat edilirse,  $\mathbf{b}_\gamma$  çözümü  $\mathbf{b}_\lambda$  çözümüne göre (Eş. 13)

$$(\mathbf{b}_\gamma^y \neq \mathbf{b}_\lambda^y) \wedge (\forall k \in \{1, \dots, K\} : b_{\gamma,k}^y \leq b_{\lambda,k}^y) \quad (13)$$

koşulunu *ancak ve ancak* şartıyla sağlanması halinde baskındır. Parçacık konumlarının ilk değerlerinin verilmesi ve kısıt matrisinin hesaplanmasından sonra, olabilir görülen konumlar için amaç vektörleri hesaplanır. Buna göre baskın çözüm kümeleri güncellenir. MOFEPSO'da olabilir görülmeyen (veya bir başka deyişle *olursuz*) bireyler/noktalar için amaç fonksiyonu hesaplanmaz.

MOFEPSO'da olursuz parçacıkların hareketi etkinleştirilmiş kısıttan (EK'dan) etkilenir. EK'nın seçimi ileriki bölümlerde irdelenmektedir. MOFEPSO'da olursuz parçacıklar sadece EK'nın duyarlı olduğu boyutlarda hareket eder. Bu nedenle, kısıtlar için bir duyarlılık matrisi tanımlanır ve ilk değerleri verilir: Eş. 14)

<sup>1</sup> "Olabilir" bir çözüm tüm kısıtları sağlayan bir karar vektörünü ifade etmektedir.



Şekil 1. MOFEPSO akış şeması (MOFEPSO flowchart)

$$S = [s_{m,n}] \in \mathbb{B}^{M \times N} \quad (14)$$

Burada  $\mathbb{B}$  ikili-düzendeki sayı (Boolean) kümesidir ve (Eş. 15)

$$s_{m,n} = \begin{cases} 1, & \text{kısıt } m, n \text{ değişkenine duyarlıdır} \\ 0, & \text{aksi halde} \end{cases} \quad (15)$$

EK'nın seçimi aynı zamanda kısıt önceliklerine ve her bir kısıtın ilgili parçacık için kaç defa EK olarak seçilmiş olmasına bağlıdır. Başlangıçta değer verilmiş *kısıt*

*öncelikleri* her parçacık hareketi sonrasında güncellenmektedir. Bu nedenle  $T = [t_{i,m}] \in \mathbb{R}^{I \times M}$  bir sıfır matrisi olarak,  $p = [p_1, p_2, \dots, p_M]^T \in \mathbb{R}^{M \times 1}$  ise her kısıt için ihlal eden parçacık sayılarıyla başlatılır. Yani, (Eş. 16)

$$p_m = \sum_{i=1}^I z_{i,m}, \quad z_{i,m} = \begin{cases} 1, & c_{i,m} > 0 \\ 0, & \text{aksi halde} \end{cases} \quad (16)$$

$p_m$ ,  $m$  numaralı kısıtın önceliğidir.  $c_{i,m} > 0$ , bu kısıtın ihlal edildiğini ifade eder.

MOFEPSO algoritmasında olurlu (olabilir) ve olursuz parçacıklar birbirinden farklı davranışlar sergiler. Aşağıdaki iki kısım bu davranışları ayrı ayrı incelemektedir.

2.1. Olursuz Parçacık Davranışı (Infeasible Particle Behavior)

Olursuz parçacığın hareketi her yinelemede her parçacık için seçilen EK'ya büyük ölçüde dayalıdır. Olursuz parçacık en az seçilmiş kısıtlar içinden önceliği en yüksek olanı EK olarak (ki aşağıdaki ifadede  $a$  indisiyle gösterilmektedir) seçer. Bu seçimden sonra,  $T$  matrisi güncellenir: (Eş. 17)

$$t_{i,a} := t_{i,a} + 1 \tag{17}$$

Bu seçim yaklaşımı herhangi bir yüksek öncelikli kısıta aşırı yoğunlaşmayı önleyerek, sürünün çeşitliliğinin korunmasını sağlar. EK'nın seçilmesinin ardından, parçacığın hareketine öncülük üzere bir genel rehber çözüm seçilir. Bu rehber EK'yi sağlayan parçacıkların içinden rastgele seçilir.

Bu parçacığın konumuna  $\mathbf{x}^G$  dersek, parçacık hızı aşağıda görüldüğü gibi hesaplanır: (Eş. 18)

$$v_{i,n} := s_{a,n} [Wv_{i,n} + C_1 r_{1,n} (x_n^G - x_{i,n})] \tag{18}$$

Burada  $W$  atalet çarpanını,  $C_1$  sosyal hızlanma katsayısını,  $r_{1,n} \in [0,1] \subset \square$  düzgün olasılık dağılımlı bir rastgele sayıyı temsil etmektedir.  $s_{a,n}$  terimi EK'nın duyarlı olmadığı boyutlardaki hız vektörü bileşenlerini sıfırlayarak, parçacığın bu boyutlardaki hareketini engeller. Hız vektörünün hesaplanmasından sonra, parçacık hareketi Bölüm 2.3'te detaylandırıldığı gibi gerçekleşir. Parçacık hareketi olurlu ve olursuz parçacıklar için ortaktır.

2.2. Olurlu Parçacık Davranışı (Feasible Particle Behavior)

Olurlu parçacıklar sosyal olarak genel baskın çözüm kümesindeki konumlara, deneyimsel olaraksa kendi baskın çözüm kümelerindeki konumlara öykünürler. Bu kümelerden rastgele seçilen konumlar ( $\mathbf{x}^G$  ve  $\mathbf{x}^P$ ) kullanılarak parçacık hızı hesaplanır: (Eş. 19)

$$v_{i,n} := Wv_{i,n} + C_1 r_{1,n} (x_n^G - x_{i,n}) + C_2 r_{2,n} (x_n^P - x_{i,n}) \tag{19}$$

Burada  $C_2$  deneyimsel hızlanma katsayısını,  $r_{2,n} \in [0,1] \subset \square$  düzgün olasılık dağılımlı bir rastgele sayıyı ifade etmektedir.

2.3. Parçacık Hareketi (Particle Flight)

Hesaplanan parçacık hızı kullanılarak bir aday konum hesaplanır: (Eş. 20)

$$x_n^* := x_{i,n} + v_{i,n} \tag{20}$$

Parçacığın bu aday konuma ilerleyebilmesi için aşağıdaki iki kistası sağlaması gerekir:

- $\mathbf{x}^*$  hiçbir karar değişkeni sınırını ihlal etmemelidir: (Eş. 21)

$$(\nexists n) (x_n^* < x_n^L \vee x_n^* > x_n^U) \tag{21}$$

- $\mathbf{x}^*$  parçacığın mevcut konumunda sağladığı hiçbir kısıtı ihlal etmemelidir: (Eş. 22)

$$(\nexists m) (c_{i,m} \leq 0 \wedge c_m^* > 0) \tag{22}$$

Sınırları ihlal eden aday konumlar, ihlal ettikleri en yakın sınırın üstüne çekilirler: (Eş. 23)

$$x_n^* := x_{i,n} + \frac{x_n^* - x_{i,n}}{1 + w} \tag{23}$$

Burada  $w$  sınır aşım oranıdır ve (Eş. 24, Eş. 25 ve Eş. 26)

$$w = \max(w^L, w^U) \tag{24}$$

$$w^L = \max_n \left( \frac{x_n^L - x_n^*}{x_{i,n} - x_n^L} \right) \tag{25}$$

$$w^U = \max_n \left( \frac{x_n^* - x_n^U}{x_n^U - x_{i,n}} \right) \tag{26}$$

şeklinde tanımlanır. Mevcut parçacık konumunda sağlanan herhangi bir kısıtın ihlal edilmesi halinde ise **Sabit Kesen Yöntem**'ine benzer bir ilkeyle çalışan **Sanal Sınır Araması** (SSA) yöntemi ile sanal olurluk denen bu kuralın ihlal edilmediği (sanal olurlu) bir konum saptanır. SSA, sanal olurluktan sanal olamazlığa (olursuza) geçişin gerçekleştiği sınırı bulmak için Sanal Olurlu ( $\mathbf{x}^+$ ) ve Sanal Olursuz ( $\mathbf{x}^-$ ) adı verilen iki nokta arasında yinelemeli bir arama gerçekleştirir. Başlangıçta bu iki nokta, parçacığın mevcut konumu ve sanal olursuz aday konumu olarak belirlenir: (Eş. 27 ve Eş. 28)

$$\mathbf{x}^+ := \mathbf{x}_i \quad \mathbf{c}^+ := \mathbf{c}_i \tag{27}$$

$$\mathbf{x}^- := \mathbf{x}^* \quad \mathbf{c}^- := \mathbf{c}^* \tag{28}$$

Her yinelemede bir ihlal oranı ( $q$ ) hesaplanır: (Eş. 29)

$$q = \max_{m \in Y} \left( \frac{c_m^-}{-c_m^+} \right) \tag{29}$$

Burada  $Y$  mevcut parçacık konumunun sağladığı kısıtlar kümesidir ve (Eş. 30)

$$\mathbf{Y} = \left\{ m \in \square_{>0}^{\leq M} \mid c_{i,m} \leq 0 \right\} \quad (30)$$

olarak tanımlanır. İhlal oranı kullanılarak, yeni bir sınama konumu ve bu sınama konumuna karşılık gelen kısıt vektörü (Eş. 31 ve Eş. 32)

$$\mathbf{x}^r := \mathbf{x}^+ + \frac{\mathbf{x}^- - \mathbf{x}^+}{1+q} \quad (31)$$

$$\mathbf{c}^r := [g_1(\mathbf{x}^r) \quad g_2(\mathbf{x}^r) \quad \dots \quad g_M(\mathbf{x}^r)]^T \quad (32)$$

hesaplanır. Eğer bu sınama konumu sanal olurluk koşullarını (Eş. 33)

$$(\forall m) (c_{i,m} \leq 0 \wedge c_m^r > 0), \quad (33)$$

sağlıyorsa, SSA'nın Sanal Olurlu noktasının yerini alır: (Eş. 34)

$$\mathbf{x}^+ := \mathbf{x}^r \quad \mathbf{c}^+ := \mathbf{c}^r \quad (34)$$

Aksi taktirde (Eş. 35)

$$\mathbf{x}^- := \mathbf{x}^r \quad \mathbf{c}^- := \mathbf{c}^r \quad (35)$$

halini alır. Bu yinelemeler SSA sonlandırma kistası sağlanana kadar devam eder: (Eş. 36)

$$(\forall n) \left( \frac{|x_n^r - x_n^e|}{x_n^U - x_n^L} \leq x^{tol} \right) \quad (36)$$

Şöyle ki (Eş. 37)

$$\mathbf{x}^e := \begin{cases} \mathbf{x}^+, & q \geq 1 \\ \mathbf{x}^-, & q < 1 \end{cases} \quad (37)$$

Yinelemeler tamamlandığında aday konum sanal olurlu nokta olarak güncellenir: (Eş. 38)

$$\mathbf{x}^* := \mathbf{x}^+ \quad \mathbf{c}^* := \mathbf{c}^+ \quad (38)$$

Eş. 21 veya 22 ile tanımlanan kistasların karşılanamaması nedeniyle aday konumu karar değişkeni sınırları içine çekildiye veya SSA uygulandıysa parçacık hızı sıfırlanır:  $v_{i,n} := 0$ .

Karar değişkeni sınırlarını ihlal etmeyen sanal olurlu aday parçacığın yeni konumu olarak atanır:  $\mathbf{x}_i := \mathbf{x}^*$  ve  $\mathbf{c}_i := \mathbf{c}^*$ . Aynı zamanda eğer aday konum olurlu ise amaç vektörü hesaplanarak  $[\mathbf{y}^* := \mathbf{f}(\mathbf{x}^*)]$  genel ve özel baskın çözüm kümeleri güncellenir. Eğer baskın çözüm kümesindeki (özel ve genel baskın çözüm kümeleri için ayrı ayrı) hiçbir çözüm

$\mathbf{y}^*$  vektörüne baskın gelmezse,  $\mathbf{y}^*$  vektörü kümeye eklenir ve  $\mathbf{y}^*$  vektörünün baskın olduğu tüm çözümler bu kümeden çıkarılır.

MOFEPSO algoritmasının MATLAB uygulamasına “<https://www.mathworks.com/matlabcentral/fileexchange/68990>” adresinden ulaşılabilir.

### 3. SONUÇLAR VE TARTIŞMALAR (RESULTS AND DISCUSSIONS)

Çok amaçlı ve tek amaçlı yaklaşımların kıyaslanması için SRN [23], RC-MOG (İng. reduced coonstraints multi-objective gear train) [5], ve MV-MOG (İng. mixed valued multi-objective gear train) [5] problemleri kullanılmıştır. SRN problemi (Eş. 39-Eş. 44)

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) \equiv [f_1(\mathbf{x}) \quad f_2(\mathbf{x})]^T \quad (39)$$

$$\mathbf{g}_m(\mathbf{x}) \leq 0, \quad m \in \square_{>0}^{\leq 2} \quad (40)$$

$$f_1(\mathbf{x}) = (x_1 - 2)^2 + (x_2 - 1)^2 + 2 \quad (41)$$

$$f_2(\mathbf{x}) = 9x_1 - (x_2 - 1)^2 \quad (42)$$

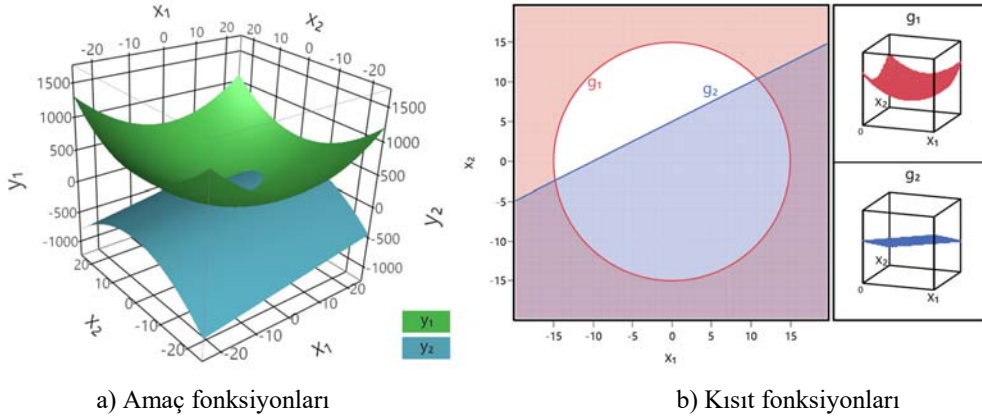
$$g_1(\mathbf{x}) = x_1^2 + x_2^2 - 225 \quad (43)$$

$$g_2(\mathbf{x}) = x_1 - 2x_2 + 10 \quad (44)$$

şeklinde tanımlı test problemidir (Şekil 2). RC-MOG ve MV-MOG problemleri ise aslında Pomrehn ve Papalambros [24] tarafından tasarlanan aşırı kısıtlı bir çok kademeli dişli kutusu eniyileme probleminin özel halleridir. Problemin çok amaç içeren ilk hali Dölen, Kaplan ve Seireg [25] tarafından tanımlanmış; Hasanoğlu ve Dölen [5] tarafından ise azaltılmış kısıtlı çok amaç içeren çok kademeli dişli kutusu problemi (RC-MOG) ve karışık değerli çok amaç içeren çok kademeli dişli kutusu problemi (MV-MOG) olmak üzere iki yeni hali daha tanımlanmıştır. Bu problemler tüm dişlilerin hacimlerinin toplamı ile orantılı olan toplam ağırlığın ve toplam dişli kutusu hacminin en aza indirilmesi (minimize edilmesi) olarak tanımlı iki amaçla, dört kademeli bir dişli kutusunun tasarlanmasını kapsar. Problemin tasarım parametreleri her kademede büyük dişlilerin konumları, ilk kademede pinyon dişlinin konumu (diğer kademelerde pinyon dişli bir önceki kademenin büyük dişlisiyle aynı konumdadır), dişlilerin diş sayıları ve her kademenin dişli kalınlığıdır. Her iki problem de (Eş. 45-Eş. 48)

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) \equiv [f_1(\mathbf{x}) \quad f_2(\mathbf{x})]^T \quad (45)$$

$$\mathbf{g}_m(\mathbf{x}) \leq 0, \quad m \in \square_{>0}^{\leq 86} \quad (46)$$



**Şekil 2.** SRN probleminin amaç ve kısıt fonksiyonları (Objective- and constraint functions of the SRN problem)

$$f_1(\mathbf{x}) = \pi \sum_{s=1}^4 b_s (C_s)^2 \frac{(N_s^p)^2 + (N_s^g)^2}{(N_s^p + N_s^g)^2} \quad (47)$$

$$f_2(\mathbf{x}) = L_x L_y \sum_{s=1}^4 b_s \quad (48)$$

olarak ifade edilebilir.

RC-MOG, MV-MOG ve SRN problemleri MOFEPSO algoritması kullanılarak çok amaçlı ve tek amaçlı yaklaşımlarla çözülmüştür (herbiri için 50 defa). Tek amaçlı yaklaşımla gerçekleştirilen çözümlerde iki amaç fonksiyonuna sahip problemler doğrusal bir fonksiyon ve üç farklı ağırlık kombinasyonu kullanılarak tek amaç fonksiyonuna sahip problemlere dönüştürülmüştür: (Eş. 49, Eş. 50 ve Eş. 51))

$$y^1 = 0,5y_1 + 0,5y_2 \quad (49)$$

$$y^2 = 0,6y_1 + 0,4y_2 \quad (50)$$

$$y^3 = 0,7y_1 + 0,3y_2 \quad (51)$$

Farklı ağırlıklar ile elde edilecek sonuçları da değerlendirebilmek amacıyla ağırlık kombinasyonları her iki amaç fonksiyonunun eşit ağırlıklandırıldığı ve buna ilaveten  $y_1$  amaç fonksiyonuna görece daha yüksek ağırlıkların atandığı kombinasyonlar seçilmiştir. Dolayısıyla tek amaçlı yaklaşım üç farklı ağırlık kombinasyonu için ayrı ayrı işletilmiştir. Dikkat edilirse, MOFEPSO algoritmasının tek amaç içeren problemleri de çözebileceği görülür. Şöyle ki algoritmanın tek amaçla kullanıldığı durumdaki davranışı tek amaçlı hali olan FEPSO [4] ile aynıdır. Bu durumda genel ve özel baskın kümelerde sadece amaç fonksiyonunun bulunmuş en iyi değerine sahip tek bir konum (veya aynı amaç fonksiyonu değerine sahip birkaç konum) bulunur. Bu nedenle çok amaçlı yaklaşımla gerçekleştirilen koşullar yanında birden fazla çözüm elde edilebilirken tek amaçlı yaklaşımla gerçekleştirilen koşullarda sadece bir çözüm elde

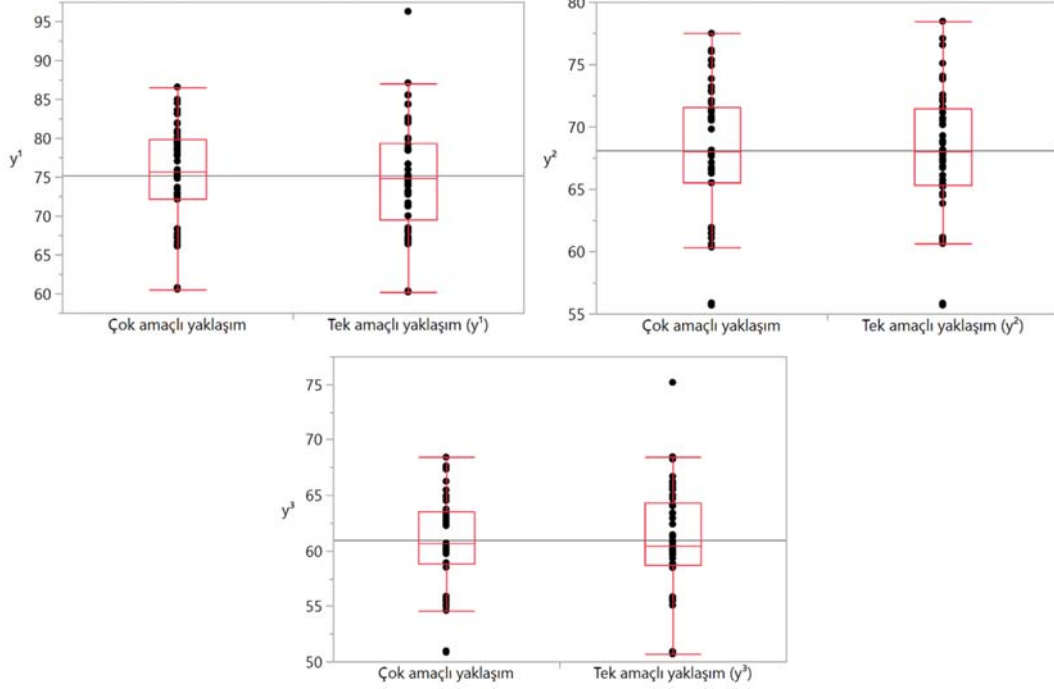
edilebilmektedir. Çok amaçlı yaklaşımda amaç fonksiyonları için önceden bir ağırlık atanması da gerekmediği gibi çözümler arasında gerçekleştirilecek getiri-götürü değerlendirmesi eniyileme sonrasına bırakılabilir. Bu çalışmada çok amaçlı ve tek amaçlı yaklaşımların karşılaştırılabilmesi için çok amaçlı koşullarda elde edilen çözümlerin tümü için Eş. 49, 50 ve 51 ile tanımlanan tek amaçlı kısıtlar da hesaplanmıştır. Çok amaçlı koşullar bu kısıtlardan biri kullanılarak gerçekleştirilen tek amaçlı koşullarla karşılaştırılırken çok amaçlı koşulların her birinde bulunan çözümlerden bu kısıt bakımından en iyi olanları dikkate alınmıştır. Tüm koşullarda MOFEPSO parametreleri  $W = 0,9 \rightarrow 0,4$ ,  $C_1 = 2$ ,  $C_2 = 2$  ve  $x^{tol} = 0,01$  olarak alınmıştır. RC-MOG ve MV-MOG problemleri için gerçekleştirilen çözümlerde parçacık sayısı 200, yineleme sayısı 500 olarak; SRN problemi için gerçekleştirilen çözümlerde ise parçacık sayısı ve yineleme sayısı 100 olarak belirlenmiştir.

RC-MOG ve MV-MOG için gerçekleştirilen tüm koşullar tek amaçlı kısıtlar bakımından incelendiğinde  $y^1$  ve  $y^2$  kısıtlarının normal dağılıma sahip olduğu,  $y^3$  kısıtlarının ise normal dağılmadığı gözlenmiştir (Shapiro-Wilk W Testi [26] kullanılmıştır). Bu nedenle  $y^1$  ve  $y^2$  kısıtları ile ilgili karşılaştırmalarda Student t-test,  $y^3$  kısıtları ile ilgili karşılaştırmalarda ise Wilcoxon işaretli sıra testi [27] uygulanmıştır. SRN çözümlerinde ise kısıtların hiçbirinin normal dağılıma sahip olmadığı görüldüğünden, bu çözümlerin karşılaştırılmasında da Wilcoxon işaretli sıra testi kullanılmıştır. RC-MOG problemi için bulunan çözümler tek amaçlı kriterler bakımından kıyaslandığında çok amaçlı yaklaşımın tek amaçlı yaklaşımlardan herhangi birine üstünlüğü gösterilemediği gibi tek amaçlı yaklaşımların da çok amaçlı yaklaşımdan üstün olmadığı görülmüştür (Şekil 3). MV-MOG problemi için bulunan çözümler değerlendirildiğinde çok amaçlı yaklaşımın  $y^2$  ve  $y^3$  kısıtları için daha iyi olduğu görülmüştür (sırasıyla  $p=0,0434$ ,  $p=0,0196$ ) (Şekil 4). Eniyileme koşulları sırasında RC-MOG ve MV-MOG problemleri için bulunan çözümlerden seçilen iki karar vektörü Şekil 5'te dışı kutusu

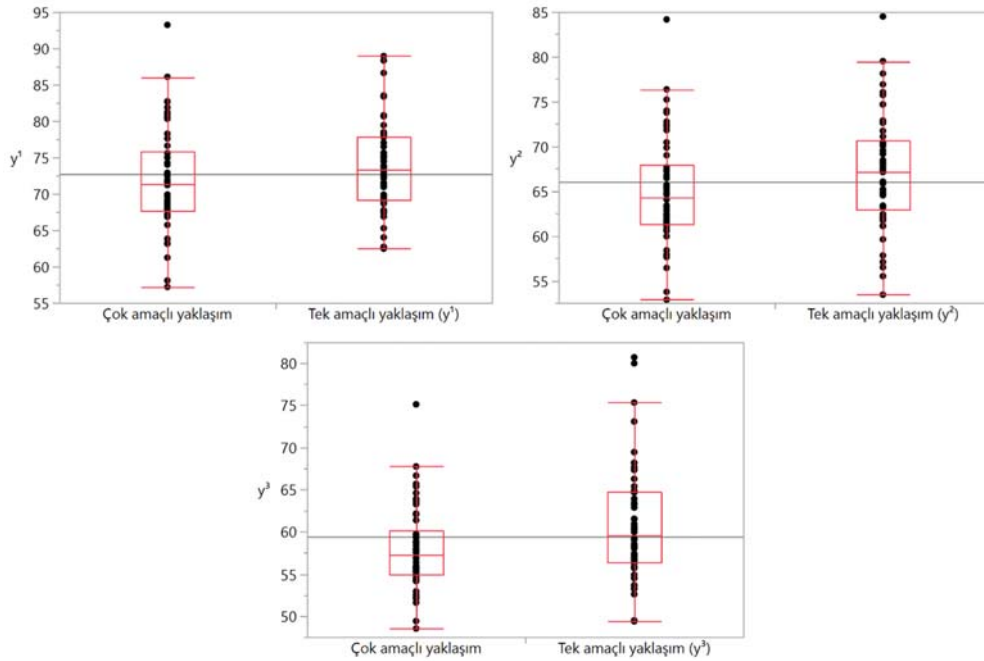


sınırları ile birlikte gösterilmiştir. Benzer şekilde, SRN problemi için çok amaçlı yaklaşımla bulunan çözümlerin içinde, tek amaçlı yaklaşımla bulunan çözümler kadar iyi çözümlerin bulunduğu görülmüştür. SRN problemi için de

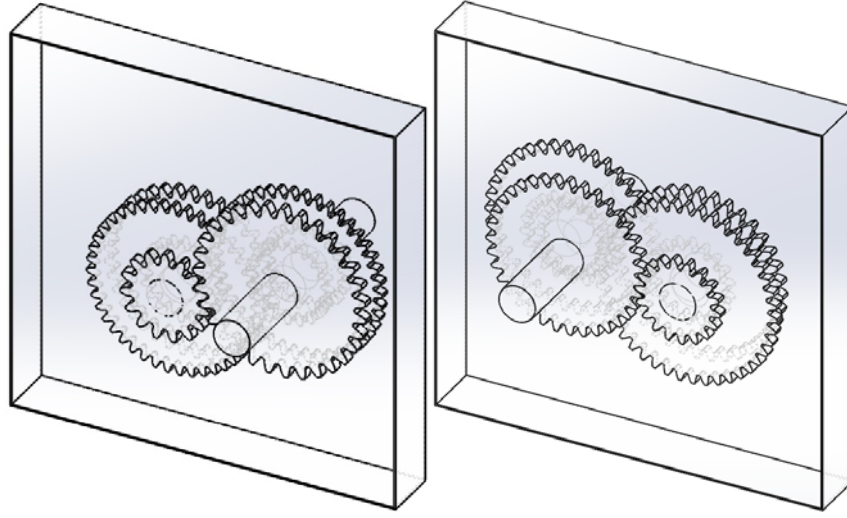
seçilen bir çözüm Şekil 6'da gösterilmiştir. İstatistiksel olarak anlamlı sonuçların bulunmuş olması gerçekleştirilen koşu sayılarının (50'şer defa) da yeterli olduğu yönünde yorumlanabilir.



**Şekil 3.** RC-MOG problemi için çok amaçlı yaklaşım ile elde edilen çözümlerin sırasıyla  $y^1$ ,  $y^2$  ve  $y^3$  kistaslarının en aza indirildiği (minimize edildiği) tek amaçlı yaklaşım çözümleriyle kıyası (A comparison of solutions of the RC-MOG problem obtained using the multi-objective approach with solutions obtained by minimization of  $y^1$ ,  $y^2$  and  $y^3$  criteria using the single-objective approach)

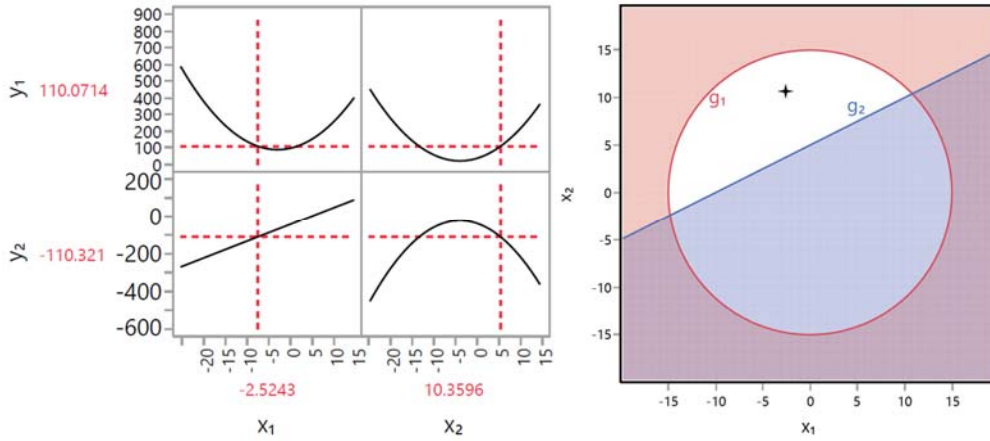


**Şekil 4.** MV-MOG problemi için çok amaçlı yaklaşım ile elde edilen çözümlerin sırasıyla  $y^1$ ,  $y^2$  ve  $y^3$  kistaslarının en aza indirildiği (minimize edildiği) tek amaçlı yaklaşım çözümleriyle kıyası (A comparison of solutions of the MV-MOG problem obtained using the multi-objective approach with solutions obtained by minimization of  $y^1$ ,  $y^2$ , and  $y^3$  criteria using the single-objective approach)



a) RC-MOG Çözümü b) MV-MOG Çözümü

Şekil 5. Seçkin RC-MOG ve MV-MOG çözümleri a) RC-MOG  $y_1=36.25$   $y_2=85.23$  b) MV-MOG  $y_1=35.14$   $y_2=82.85$   
(Selected RC-MOG and MV-MOG solutions)

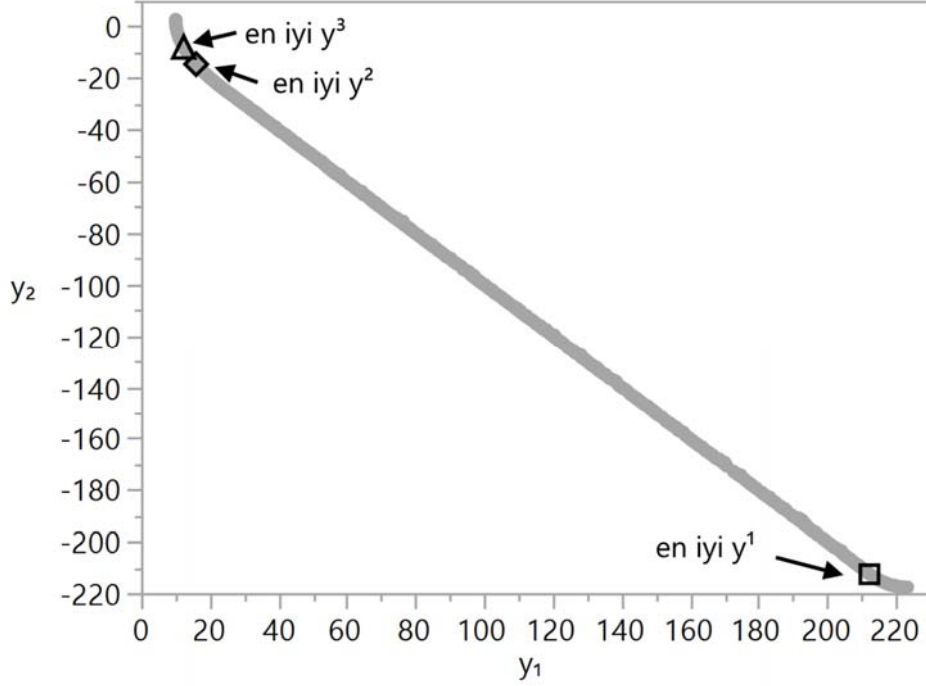


a) Amaç fonksiyonu değerleri b) Çözümün kısıt düzlemindeki konumu

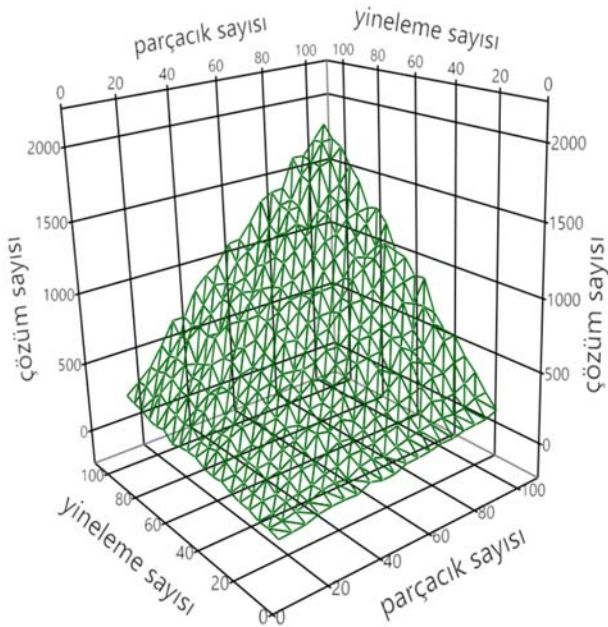
Şekil 6. Seçkin bir SRN çözüm noktası (A selected SRN solution)

Bu sonuçlar ışığında, MOFEPSO ile amaç fonksiyonlarına önceden bir ağırlık verilmeksizin çok amaçlı bir yaklaşımla gerçekleştirilen eniyilemenin incelenen tüm tek amaçlı kısıtlar için de uygun sonuçlar verebildiği söylenebilir. Diğer Pareto tabanlı eniyileme algoritmalarında olduğu gibi MOFEPSO da eniyileme koşusu sonunda bir Pareto cephesi üretir. Böylece, tasarımcı çözümler arasındaki kararı tüm baskın çözümleri değerlendirerek verebilir. Bu yaklaşımla amaç fonksiyonlarına sonradan (eniyileme sonrasında) ağırlık verilebilir ve eniyileme koşusunun tekrarlanmasına gerek olmadan ağırlıklar değiştirilebilir. Örnek olarak Şekil 7’de MOFEPSO ile çok amaçlı olarak gerçekleştirilen bir koşu sonucunda elde edilen baskın çözümlerin tamamı ve bu çözümlerin içinden  $y^1$ ,  $y^2$  ve  $y^3$  kısıtları bakımından en iyi olduğu belirlenenler gösterilmiştir. Burada da görüldüğü gibi Pareto tabanlı bir çok amaçlı yaklaşımla tek bir koşuda elde edilen sonuçların her biri aslında farklı bir doğrusal sabit ağırlıkla birleştirilmiş amaç fonksiyonuyla gerçekleştirilen

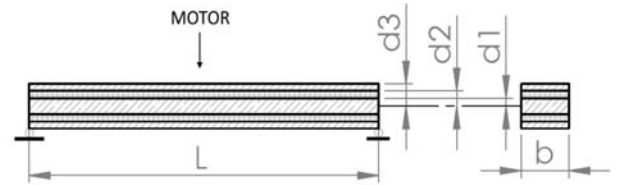
tek amaçlı eniyileme koşusuna karşılık gelmektedir. Dolayısıyla, her ne kadar çok amaçlı yaklaşım özel seçim yöntemleri gibi ilave karmaşıklık içeren adımları barındırda da birçok tek amaçlı koşuda elde edilebilecek sonuçların tek bir çok amaçlı koşunun sonucunda elde edilebilmesi sebebiyle hesaplama karmaşıklığı bakımından önemli bir kazanç sağlar. Çok amaçlı bir koşuda elde edilecek çözüm sayısı probleme bağlıdır. Şekil 8’de SRN problemi için farklı parçacık sayısı ve yineleme sayısı ile gerçekleştirilen koşularda elde edilmiş çözümlerin sayıları gösterilmiştir. Bu husus gerçek bir tasarım problemi olan titreşen platform problemi ile de açıklanabilir [28]. Bu problemde titreşen bir motoru taşıyacak olan platformun maliyetinin minimize edilmesinin yanında doğal frekansının maksimize edilmesi hedeflenmektedir. Sandviç yapıda iç katman, ara katmanlar ve dış katmanlardan oluşan platformun katmanları için üç farklı malzeme seçilebilir. Seçilebilecek malzemelerin özellikleri Tablo 1’de verilmiştir.



**Şekil 7.** SRN problemi için çok amaçlı yaklaşım ile bir koşuda elde edilen baskın çözümler (Pareto cephesi – 1879 nokta) ve bu çözümlerin içinden  $y^1$ ,  $y^2$  ve  $y^3$  kısıtaları bakımından en iyi (en küçük değere sahip) olanları (Non-dominated solutions (Pareto Front – 1879 points) for the SRN problem obtained in a single multi-objective optimization run and solutions that are best in terms of  $y^1$ ,  $y^2$ , and  $y^3$  criteria)



**Şekil 8.** SRN problemi için gerçekleştirilen çok amaçlı koşullarda parçacık sayısı ve yineleme sayısına bağlı olarak elde edilen Pareto cephesindeki çözüm sayısının değişimi. (Number of solutions in the Pareto front versus particle number and number of iterations for the multi-objective optimization of the SRN problem)



**Şekil 9.** Titreşen platform problemi. (Vibrating platform problem)

**Tablo 1.** Titreşen platformun katmanları için seçilebilecek malzemelerin özellikleri. (Properties of materials that can be selected for the layers of the vibrating platform)

	Malzeme A	Malzeme B	Malzeme C
$\rho$ (kg/m <sup>3</sup> )	100,0	2770,0	7780
E (GPa)	1.6	70	200
c (\$/m <sup>3</sup> )	500,0	1500	800

Problemin karar vektörü katmanların her biri için malzeme seçimi  $m_1, m_2, m_3 \in \{A, B, C\}$  ile birlikte geometrik parametreleri içerir:  $0,05 \leq d_1 \leq 0,5$ ,  $0,2 \leq d_2 \leq 0,5$ ,  $0,2 \leq d_3 \leq 0,6$ ,  $0,35 \leq b \leq 0,5$  ve  $3 \leq L \leq 6$ . Problemin amacı ifade Eş. 52’de verilen doğal frekans maksimize edilirken ifade Eş. 53’te yer alan maliyetin (J) minimize edilmesidir: (Eş. 52, Eş. 53)

$$f_n = \frac{\pi}{2L^2} \sqrt{\frac{EI}{\mu}} \quad (52)$$

$$J = 2bL[c_1d_1 + c_2(d_2 - d_1) + c_3(d_3 - d_2)] \quad (53)$$

Problemin kısıtları ise aşağıdaki eşitsizliklerle tanımlanmıştır: (Eş. 54)

$$\begin{aligned} g_1 &= \mu L - 2800 \leq 0 \\ g_2 &= d_1 - d_2 \leq 0 \\ g_3 &= d_2 - d_1 - 0,15 \leq 0 \\ g_4 &= d_2 - d_3 \leq 0 \\ g_5 &= d_3 - d_2 - 0,01 \leq 0 \end{aligned} \quad (54)$$

Bu ifadelerde yer alan  $EI$  ve  $\mu$  ise (Eş. 55, Eş. 56)

$$EI = \frac{2b}{3} [E_1d_1^3 + E_2(d_2^3 - d_1^3) + E_3(d_3^3 - d_2^3)] \quad (55)$$

$$\mu = 2b[\rho_1d_1 + \rho_2(d_2 - d_1) + \rho_3(d_3 - d_2)] \quad (56)$$

şeklinde yazılabilir.

Problemin iki amaç fonksiyonu bulunduğundan çok amaçlı optimizasyon yaklaşımlarıyla doğrudan çözülebilirken tek amaçlı yaklaşımlarla çözülebilmesi için tek bir amaç fonksiyonuna indirgenmesi gereklidir. Tek amaçlı yaklaşımla problemi ele alacak bir tasarımcı her iki amaç fonksiyonuna da eşit ağırlık atayarak aşağıdaki amaç fonksiyonunu minimize edebilir: (Eş. 57)

$$y = 0,5J - 0,5f_n \quad (57)$$

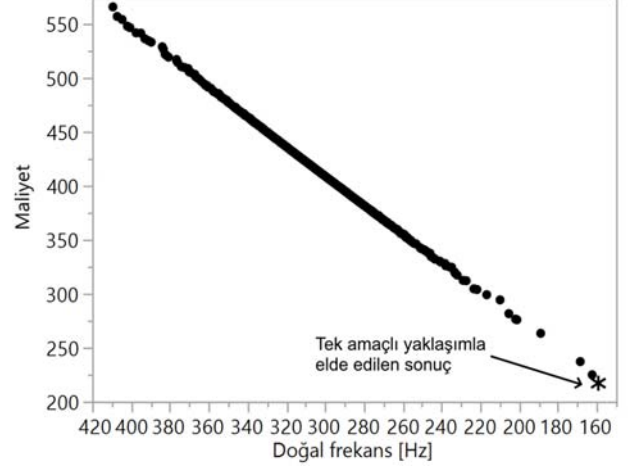
Yukarıda da belirtildiği gibi tek amaçlı yaklaşımla gerçekleştirilen bir optimizasyon koşulunda tek bir sonuç elde edilebilirken çok amaçlı yaklaşımla gerçekleştirilen bir koşulda bir Pareto cephesi elde edilir. Şekil 10'da tek amaçlı yaklaşımla ifade Eş. 57 kullanılarak elde edilen bir sonuç ile çok amaçlı bir MOFEPSO koşulunda elde edilen çözüm kümesi birlikte gösterilmiştir.

Şekil 10'da görülen sonuçlar ışığında aşağıdaki yargılara varılabilir:

Tek amaçlı yaklaşım kullanan bir tasarımcı amaç fonksiyonlarına eşit ağırlıklar atadığı halde aslında bu fonksiyonlar arasındaki ölçek farkından dolayı maliyeti önceleyen bir optimizasyon gerçekleştirmiştir. Amaç fonksiyonlarının tasarım uzayı içerisinde alabileceği değerler bilinmeksizin gerçekleştirilen ağırlık atamaları beklenen sonucu çoğunlukla veremez.

Çok amaçlı yaklaşım kullanan bir tasarımcı tasarım uzayının içerisinde hangi doğal frekans ve maliyet çiftlerinin gerçekleştirilebileceğini ortaya çıkarmanın ötesinde doğal

frekans ile maliyet arasında bir ilişki olduğunu ve doğal frekansı artırmak istediğinde maliyetten ne kadar ödün vermesi gerektiğini de optimizasyon çıktısı olarak elde edebilir.



**Şekil 10.** Titreşen platform problemi için çok amaçlı ve tek amaçlı birer koşulun sonuçları.

(Solutions obtained in a single multi-objective and single-objective optimization run for the vibrating platform problem)

Bu örnekte, tek amaçlı yaklaşımla optimizasyon yapan tasarımcının sadece 160 Hz doğal frekansa sahip olan bir motor taşıma platformu elde edebildiği görülmektedir. Bilindiği üzere, motorun tahrik frekansının ( $f_e$ ) taşıyıcı platformunun genel olarak altında olması istenir. Böylece, motorun; taşıyıcı sistem rezonans frekansının uzağında işletilmesi hedeflenir. Genel kabul görmüş pratik bir kural olarak,  $f_n/f_e \geq 2$  alınabilir. Bu tasarımda motorun tahrik frekansının 0-80 Hz bandında seçilmesi uygun olacaktır. Diğer taraftan, (başka bir uygulama çerçevesinde) motor tipinin ve yükünün değişimine bağlı olarak, motorun tahrik frekans bandının 0-120 Hz'e çıktığı kabul edilsin. Bu durumda, mevcut platform tasarımı artık uygun olmayacaktır ve (240 Hz gibi) daha yüksek bir doğal frekansa sahip yeni bir tasarımın yapılması gerekecektir. Bu durumda çok-amaçlı yaklaşım kullanılarak, yeni bir "optimal" tasarımın ortaya konmasına ihtiyaç olmayacağı açıktır: Pareto cephesi çözümleri içinde yer alan ve 240 Hz'lik bir doğal frekansa karşılık gelen (mevcut) çözümlerden birinin seçilmesi yeterli olacaktır. Diğer taraftan, tek-amaçlı yaklaşımda, amaç fonksiyonundaki ağırlıklar (deneme-yanılma yoluyla) değiştirilerek, istenen doğal frekansı sağlayan bir çözümün "aranması" gerekecektir. Gerçekten böyle bir yaklaşım izlendiğinde 8 değişik deneme sonucunda istenen doğal frekansa yaklaşan bir tasarım bulunmuştur:  $f_n = 240,4$  Hz;  $J = 329,3$  USD;  $y = 0,1 J - 0,9 f_n$ . Böylesi sistematik olmayan yaklaşımlar – özellikle amaç fonksiyonu hesabının çok zahmetli (pahalı) olduğu durumlarda – önemli bütçesel sorunlara ve zaman kayıplarına yol açabilmektedir. Oysa, çok amaçlı yaklaşım uygulayan tasarımcının hem farklı tasarım seçeneklerini değerlendirebilmesi hem de bu seçeneklerin amaç uzayındaki etkilerini görebilmesine olanak sağlamaktadır. Çok amaçlı yaklaşım amaç

fonksiyonlarının gerçekleştirilebilir değerleri ve aralarındaki ilişki hakkında sağladığı bilgi sayesinde tasarımcıya bilinçli bir ödünleşim gerçekleştirme imkânı sağlar.

#### 4. SONUÇLAR (CONCLUSIONS)

Çok amaçlı RC-MOG, MV-MOG ve SRN problemleri MOFEPSO algoritması kullanılarak ellisek defa çözülmüştür. Bu çözümler ile kıyaslama yapabilmek amacıyla söz konusu problemlerin her biri üç farklı ağırlık kombinasyonu kullanılarak ( $y^1$ ,  $y^2$  ve  $y^3$ ) tek amaçlı problemlere dönüştürülmüş ve her biri elli defa FEPSO algoritması kullanılarak çözülmüştür. Tüm problemler için  $y^1$ ,  $y^2$  ve  $y^3$  kıstasları kullanılarak gerçekleştirilen tek amaçlı eniyilemelerde elde edilen sonuçlar ayrı ayrı çok amaçlı eniyileme ile elde edilen sonuçlarla karşılaştırıldığında, tek amaçlı eniyileme ile elde edilen çözümlerin üstünlüğü görülmüştür. Dolayısıyla MOFEPSO ile çok amaçlı olarak gerçekleştirilen eniyilemelerde elde edilen çözümlerin her bir tek amaçlı kıstas için gerçekleştirilen ayrı eniyilemelere denk veya daha iyi olduğu görülmüştür. Bununla birlikte ayrı çok amaçlı MOFEPSO koşullarında elde edilen Pareto cepheleri tek amaçlı kıstaslar bakımından incelenerek, söz konusu kıstasların farklı değerlerine denk gelen sonuçların elde edilebildiği görülmüştür. Ayrıca bir tasarım problemi için her iki yaklaşımla elde edilen sonuçlar karşılaştırılmıştır. Bu sonuçlar ışığında:

Amaç fonksiyonlarına önceden ağırlık verilmeksizin gerçekleştirilen çok amaçlı eniyileme koşulları ile elde edilen sonuçlar herhangi bir ağırlık kombinasyonu için sonuçlar içerir. Problem hakkında yeterli bilgi elde edilmeden amaç fonksiyonlarına ağırlık atanarak tek amaçlı eniyileme yaklaşımlarının kullanılmasından kaçınılmalıdır.

Tek bir çok amaçlı eniyileme koşusu ile birçok tek amaçlı koşulda elde edilebilecek sonucun elde edilebilmesi sebebi ile çok amaçlı yaklaşımın daha verimli olduğu görülmüştür. Başka bir deyişle çok amaçlı yaklaşım hesaplama karmaşıklığı bakımından önemli avantaj sağlayabilmektedir. Bu avantaj özellikle sonlu elemanlar yöntemi gibi hesaplama karmaşıklığı yüksek amaç fonksiyonları kullanıldığında önemli hale gelir.

Tek amaçlı yaklaşımdan farklı olarak, çok amaçlı yaklaşım amaç fonksiyonlarının gerçekleştirilebilir değerleri ve aralarındaki ilişki hakkında sağladığı bilgi sayesinde tasarımcıya bilinçli bir ödünleşim gerçekleştirme imkânı sağlar.

#### KAYNAKLAR (REFERENCES)

- Zhou A, Qu B-Y, Li H, Zhao S-Z, Suganthan PN, Zhang Q, Multiobjective evolutionary algorithms: A survey of the state of the art, *Swarm Evol Comput*, 1 (1), 32–49, 2011.
- Coello Coello CA, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, *Comput Methods Appl Mech Eng*, 191 (11–12), 1245–87, Jan 4, 2002.
- Mezura-Montes E, Coello CAC, Constraint-handling in nature-inspired numerical optimization: Past, present and future, *Swarm Evol Comput*, 1 (4), 173–94, 2011.
- Hasanoğlu MS, Dolen M, Feasibility enhanced particle swarm optimization for constrained mechanical design problems, *Proc Inst Mech Eng Part C J Mech Eng Sci*, 232 (2), 381–400, Jan 1, 2018.
- Hasanoğlu MS, Dolen M, Multi-objective feasibility enhanced particle swarm optimization, *Eng Optim*, Feb 27, 2018.
- Eberhart RC, Kennedy J, A new optimizer using particle swarm theory, *Proceedings of the sixth international symposium on micro machine and human science*, New York, NY, 39–43, 1995.
- Bratton D, Kennedy J, Defining a Standard for Particle Swarm Optimization, 2007 IEEE Swarm Intelligence Symposium, 120–7, 2007.
- Ray T, Tai K, Seow KC, Multiobjective Design Optimization by an Evolutionary Algorithm, *Eng Optim*, 33 (4), 399–424, Apr 1 2001.
- Poli R, Analysis of the Publications on the Applications of Particle Swarm Optimisation, *J Artif Evol App*, 2008, 3, 1–3, 10, 2008.
- Lalwani S, Singhal S, Kumar R, Gupta N, A comprehensive survey: Applications of multi-objective particle swarm optimization (MOPSO) algorithm, *Trans Comb*, 2 (1), 39–101, 2013.
- Özdağ R, The optimization of network lifetime with sensor deployment for target coverage problem in wireless sensor networks, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 32 (4), 1155-1167, 2017.
- Çimen ME, Boz AF, Parameter identification of a non-minimum phase second order system with time delay using relay test and PSO, CS, FA algorithms, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 34 (1), 461-477, 2019.
- Deb K, An efficient constraint handling method for genetic algorithms, *Comput Methods Appl Mech Eng*, 186 (2–4), 311–38, Jun 9, 2000.
- Jordehi AR, A review on constraint handling strategies in particle swarm optimisation, *Neural Comput Appl*, 1–11, 2015.
- Yan D, Lu Y, Recent Advances in Particle Swarm Optimization for Large Scale Problems, *J Auton Intell*, 1 (1), 22, Oct 14 2018
- Banks A, Vincent J, Anyakoha C, A review of particle swarm optimization. Part I: background and development, *Nat Comput*, 6 (4), 467–84, Oct 2007.
- Banks A, Vincent J, Anyakoha C, A review of particle swarm optimization. Part II: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications, *Nat Comput*, 7 (1), 109–24, Mar 2008.
- Zitzler E, Thiele L, Multiobjective optimization using evolutionary algorithms — A comparative case study, *Parallel Problem Solving from Nature — PPSN V*,

- Springer, Berlin, Heidelberg, (Lecture Notes in Computer Science), 292–301, 1998.
19. Srinivas N, Deb K, Multiobjective optimization using nondominated sorting in genetic algorithms, *Evol Comput*, 2 (3), 221–48, 1994.
  20. Coello CAC, An updated survey of evolutionary multiobjective optimization techniques: state of the art and future trends, *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99* (Cat No 99TH8406), p. 13 Vol. 1, 1999.
  21. Ishibuchi H, Nojima Y, Doi T, Comparison between Single-Objective and Multi-Objective Genetic Algorithms: Performance Comparison and Performance Measures, 2006 IEEE International Conference on Evolutionary Computation, 1143–50, 2006.
  22. Mirjalili S, Jangir P, Saremi S, Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems, *Appl Intell*, 46 (1), 79–95, Jan 2017.
  23. Deb K, Pratap A, Agarwal S, Meyarivan T, A fast and elitist multiobjective genetic algorithm: NSGA-II, *Evol Comput IEEE Trans On*, 6 (2), 182–97, 2002.
  24. Pomrehn LP, Papalambros PY, Discrete Optimal Design Formulations With Application to Gear Train Design, *J Mech Des*, 117 (3), 419–24, Sep 1995.
  25. Dolen M, Kaplan H, Seireg A, Discrete parameter-nonlinear constrained optimisation of a gear train using genetic algorithms, *Int J Comput Appl Technol*, 24 (2), 110–21, 2005.
  26. Shapiro SS, Wilk MB, An Analysis of Variance Test for Normality (Complete Samples), *Biometrika*, 52 (3/4), 591–611, 1965.
  27. Wilcoxon F, Individual Comparisons by Ranking Methods, *Biom Bull*, 1 (6), 80–3, 1945.
  28. Augusto OB, Bennis F, Caro S, Multiobjective engineering design optimization problems: a sensitivity analysis approach, *Pesqui Oper*, 32 (3), 575–96, Dec 4 2012.