

**DİFFERANSİYELLENEMEYEN
OPTİMİZASYON PROBLEMLERİ İÇİN
ZAYIF SUBGRADYANT TEMELLİ
ÇÖZÜM YÖNTEMLERİ**

Doktora Tezi

Gülçin DİNÇ YALÇIN

Eskişehir, 2017

**DİFFERANSİYELLENEMEYEN OPTİMİZASYON PROBLEMLERİ İÇİN
ZAYIF SUBGRADYANT TEMELLİ ÇÖZÜM YÖNTEMLERİ**

Gülçin DİNÇ YALÇIN

DOKTORA TEZİ

Endüstri Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Refail KASIMBEYLİ

Eskişehir

Anadolu Üniversitesi

Fen Bilimleri Enstitüsü

Ocak, 2017

Bu tez çalışması BAP komisyonunca kabul edilen 1602F082 no.lu proje kapsamında desteklenmiştir.

JÜRİ VE ENSTİTÜ ONAYI

Gülçin DİNÇ YALÇIN'ın "Differansiyellenemeyen optimizasyon problemleri için zayıf subgradyant temelli çözüm yöntemleri" başlıklı tezi 27/01/2017 tarihinde aşağıdaki jüri tarafından değerlendirilerek "Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliği"nin ilgili maddeleri uyarınca, Endüstri Mühendisliği Anabilim dalında Doktora tezi olarak kabul edilmiştir.

	<u>Unvanı-Adı Soyadı</u>	<u>İmza</u>
Üye(Tez Danışmanı)	: Prof. Dr. Refail KASIMBEYLİ	
Üye	: Prof. Dr. Hüseyin AKÇAY	
Üye	: Prof. Dr. Emre Alper YILDIRIM	
Üye	: Prof. Dr. Metin TÜRKEY	
Üye	: Doç. Dr. Gürkan ÖZTÜRK	

Enstitü Müdürü

ÖZET

DİFFERANSİYELLENEMEYEN OPTİMİZASYON PROBLEMLERİ İÇİN ZAYIF SUBGRADYANT TEMELLİ ÇÖZÜM YÖNTEMLERİ

Gülçin DİNÇ YALÇIN

Endüstri Mühendisliği Anabilim Dalı

Anadolu Üniversitesi, Endüstri Mühendisliği Anabilim Dalı, Ocak, 2017

Danışman: Prof. Dr. Refail KASIMBEYLİ

Bu tezin amacı, differansiyellenemeyen ve dışbükey olmayan kısıtsız optimizasyon problemleri için zayıf subgradyant kavramı kullanılarak bir çözüm algoritması geliştirmektir. Zayıf subgradyant kavramı, fonksiyonların grafiklerinin, dışbükey analizdeki hiperdüzlemler yerine konilerle desteklenmesi fikrine dayandığından zayıf subgradyant dışbükeylik talep etmez ve dolayısı ile daha geniş bir fonksiyon sınıfını kapsar. Ancak zayıf subgradyantları hesaplamak kolay bir iş değildir. Bu sebeple öncelikle zayıf subgradyantların tahmini üzerinde çalışılmış ve zayıf subgradyantlar yönlü türevin zayıf subgradyantların supremumuna eşitliği hakkındaki teorem kullanılarak tahmin edilmiştir. Daha sonra zayıf subgradyant algoritması geliştirilmiş ve sabit adım uzunluğu, azalan adım uzunluğu ve dört farklı dinamik adım uzunluğu olmak üzere toplam altı adet adım uzunluğu için algoritmanın yakınsaklık özellikleri araştırılmıştır. Geliştirilen zayıf subgradyant algoritması, Python programlama dilinde kodlanmıştır. Son olarak literatürde differansiyellenemeyen ve dışbükey olmayan sadece kutu kısıtlarına sahip olan optimizasyon problemleri geliştirilen zayıf subgradyant algoritması ile çözülmüş ve algoritmanın performansı üzerinde tartışılmıştır.

Anahtar Kelimeler: Differansiyellenemeyen optimizasyon, dışbükey olmayan optimizasyon, kısıtsız optimizasyon, zayıf subgradyant, çözüm algoritması

ABSTRACT

WEAK SUBGRADIENTS BASED SOLUTION METHODS IN NONCONVEX AND NONSMOOTH OPTIMIZATION

Gülçin DİNÇ YALÇIN

Department of Industrial Engineering

Anadolu University, Graduate School of Science, January, 2017

Supervisor: Prof. Dr. Refail KASIMBEYLİ

The aim of this thesis is to develop an algorithm for nonsmooth nonconvex optimization problems by using weak subgradient. The idea of weak subgradient is based on supporting cones to graphs of functions instead of hyperplanes in convex analysis. Therefore, weak subgradient does not require any convexity assumption and thereby involve broader class of functions. However estimating of weak subgradients is not an easy task. For this reason, firstly the estimating of weak subgradients are studied and the weak subgradients are estimated by using the equality of directional derivative and supremum of weak subgradients. Then, the weak subgradient algorithm is developed and investigated the convergence properties for totally six stepsize parameter as constant, diminishing and four different type of dynamic step sizes. The proposed weak subgradient algorithm is coded with Python. Lastly, the test problems from literature that are nonsmooth and nonconvex and has onl box constraints are solved by the proposed weak subgradient algorithm and the performance of the algorithm is discussed.

Keywords: Nonsmooth optimization, nonconvex optimization, unconstrained optimization, weak subgradient, solution algorithm

TEŞEKKÜRLER

Öğrencisi olduğum günden itibaren desteğini esirgemeyen değerli hocam sayın Prof. Dr. Refail KASIMBEYLİ'ye teşekkürü bir borç bilirim.

Her konuda desteğini esirgemeyen eşim Taha Hasan YALÇIN'a ve oğlum Salih Mete YALÇIN'a teşekkür ederim.

Bu günlere gelmemi sağlayan ve her konuda desteklerini esirgemeyen aileme teşekkür ederim.

Doktora Burs Programı ile desteklerinden dolayı TUBİTAK'a teşekkürü bir borç bilirim.

Gülçin DİNÇ YALÇIN

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Bu tezin bana ait, özgün bir çalışma olduğunu; çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; bu çalışma kapsamında elde edilemeyen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi; bu çalışmanın Anadolu Üniversitesi tarafından kullanılan “bilimsel intihal tespit programı”yla tarandığını ve hiçbir şekilde “intihal içermediğini,” beyan ederim. Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçlara razı olduğumu bildiririm.

Gülçin DİNÇ YALÇIN

İÇİNDEKİLER

Sayfa

BAŞLIK SAYFASI	i
JÜRİ VE ENSTİTÜ ONAYI.....	ii
ÖZET	iii
ABSTRACT.....	iv
TEŞEKKÜRLER.....	v
ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ.....	vi
İÇİNDEKİLER	vii
TABLOLAR DİZİNİ	ix
ŞEKİLLER DİZİNİ.....	x
SİMGE VE KISALTMALAR DİZİNİ	xi
1. GİRİŞ	1
2. DİFFERANSELLENEMEYEN OPTİMİZASYON.....	3
2.1. Gösterim ve Tanımlar	3
2.2. Subgradyant	4
2.2.1. Subgradyant algoritması	5
2.4. Zayıf Subgradyant.....	7
3. ZAYIF SUBGRADYANTIN TAHMİNİ	12
4. DİFFERANSİYELLENEMEYEN DİŞBÜKEY OLMAYAN KISITSIZ OPTİMİZASYON PROBLEMLERİ İÇİN GELİŞTİRİLEN ZAYIF SUBGRADİENT YÖNTEMİ.....	17
4.1. Sabit Adım Uzunluğu ile Zayıf Subgradyant Algoritması	19
4.2. Azalan Adım Uzunluğu ile Zayıf Subgradyant Algoritması	20
4.3. f^* ve x^* ile Dinamik Adım Uzunluğu ile Zayıf Subgradyant Algoritması	21
4.4. $f^{lev} > f^*$ ile Dinamik Adım Uzunluğu ile Zayıf Subgradyant Algoritması	23
4.5. $f^{lev} < f^*$ ile Dinamik Adım Uzunluğu ile Zayıf Subgradyant Algoritması	24
4.6. f_k^{lev} ile Dinamik Adım Uzunluğu ile Zayıf Subgradyant Algoritması.....	27
5. ZAYIF SUBGRADYANT ALGORİTMASININ ANALİZİ.....	30
5.1. Yerel Optimal Çözümlere Sahip Analitik Bir Problem	30
5.2. Zayıf Subgradyant Algoritmasında c Parametresinin Belirlenmesi için Bir Yaklaşım	31

5.3. Zayıf Subgradyant Algoritmasının İteratif Gösterimi	36
5.3. Dinamik Adım Uzunluğu (4.3) ve (4.4)'ün Birlikte Kullanılması	37
5.4. Yakınsaklığı Etkileyen Parametreler	39
6. HESAPSAL SONUÇLAR	41
6.1. Küçük Boyutlu Test Problemleri	42
6.2. Büyük Boyutlu Test Problemleri	48
6.3. Tartışma	58
6.3.1. Küçük boyutlu problemler	58
6.3.2. 50 Değişkenli problemler.....	61
6.3.3. 200 Değişkenli problemler.....	64
7. SONUÇLAR	66
KAYNAKLAR	69
EKLER	74
ÖZGEÇMİŞ	92

TABLÖLAR DİZİN

Tablo 4.1.	Zayıf subgradyant algoritması için yakınsaklık özellikleri araştırılan adım uzunlukları	18
Tablo 5.1.	Analitik problemin zayıf subgradyant algoritması ile sonuçları.....	31
Tablo 5.2.	Adım uzunluklarında c_k alabileceği en büyük değer.....	32
Tablo 5.3.	Analitik fonksiyon (5.1) için sabit adım uzunluğunda c_k hesaplanması ...	33
Tablo 5.4.	Analitik fonksiyon (5.1) için azalan adım uzunluğunda (4.1) c_k hesaplanması	33
Tablo 5.5.	Analitik fonksiyon (5.1) için f^* ve x^* ile dinamik adım uzunluğunda (4.2) c_k hesaplanması.....	34
Tablo 5.6.	Analitik fonksiyon (5.1) için $f^{lev} > f^*$ ile dinamik adım uzunluğunda (4.3) c_k hesaplanması.....	34
Tablo 5.7.	Analitik fonksiyon (5.1) için $f^{lev} < f^*$ ile dinamik adım uzunluğunda (4.4) c_k hesaplanması.....	35
Tablo 5.8.	Analitik fonksiyon (5.1) için f_k^{lev} ile dinamik adım uzunluğunda (4.5) c_k hesaplanması.....	35
Tablo 5.9.	Analitik fonksiyonun (5.1) f_k^{lev} ile dinamik adım uzunluğu (4.5) ile çözümünün iteratif gösterimi.....	36
Tablo 5.10.	Dinamik adım uzunlukları (4.3) ve (4.4)'ün beraber kullanımı	38
Tablo 5.11.	Her bir adım uzunluğunun yakınsaklık sonuçlarını etkileyen değerler	39
Tablo 6.1.	Zayıf subgradyant algoritması için belirlenen parametreler	41
Tablo 6.2.	Küçük boyutlu test problemlerinin sonuçları.....	43
Tablo 6.3.	Küçük boyutlu test problemlerini çözmek için gereken CPU	44
Tablo 6.4.	Küçük boyutlu test problemlerinin optimal sonuçlara göre değerlendirilmesi	45
Tablo 6.5.	EXP problemi için zayıf subgradyant algoritması ile bulunan sonuçlar ..	47
Tablo 6.6.	“50” değişkenli test problemlerinin sonuçları	49
Tablo 6.7.	“50” değişkenli test problemlerini çözmek için gereken CPU	50
Tablo 6.8.	“50” değişkenli test problemlerinin sonuçlarının optimal sonuçlara göre değerlendirilmesi	51
Tablo 6.9.	“200” değişkenli test problemlerinin sonuçları	54
Tablo 6.10.	“200” değişkenli test problemlerini çözmek için gereken CPU	55
Tablo 6.11.	“200” değişkenli test problemlerinin sonuçlarının optimal sonuçlara göre değerlendirilmesi	56
Tablo 7.1.	Zayıf subgradyant algoritmasının her bir adım uzunluğuna göre problemlerdeki başarı yüzdeleri	67

ŞEKİLLER DİZİNİ

Şekil 5.1.	Analitik fonksiyonunun (5.1) grafiği; sınırlar her bir değişken için $[-5,5]$ ve $[-0.25,0.25]$ (Audet ve ark., 2008).....	30
Şekil 5.2.	f^{lev} ve f^{eniyi} değerlerinin değişimi	39
Şekil 6.1.	Küçük boyutlu test problemleri için adım uzunluklarının optimal değere ulaşma yüzdeleri $< 5 \times 10^{-4}$	46
Şekil 6.2.	Küçük boyutlu test problemleri için adım uzunluklarının optimal değere ulaşma yüzdeleri $< 10^{-3}$	46
Şekil 6.3.	Küçük boyutlu test problemleri için adım uzunluklarının optimal değerden farkları	47
Şekil 6.4	“50” değişkenli test problemleri için adım uzunluklarının optimal değere ulaşma yüzdeleri $< 5 \times 10^{-4}$	52
Şekil 6.5.	“50” değişkenli test problemleri için adım uzunluklarının optimal değere ulaşma yüzdeleri $< 10^{-3}$	52
Şekil 6.6	“50” değişkenli test problemleri için adım uzunluklarının optimal değerden farkları	53
Şekil 6.7.	“200” değişkenli test problemleri için adım uzunluklarının optimal değere ulaşma yüzdeleri $< 10^{-3}$	57
Şekil 6.8.	“200” değişkenli test problemleri için adım uzunluklarının optimal değerden farkları.....	57
Şekil 6.9.	Muffin 2 problemi – Sabit adım uzunluğu (a) $\ x_k - x^*\ $ (b) $f(x_k)$ iterasyonlar boyunca değişimi.....	58
Şekil 6.10.	PBC1 problemi – $f^{lev} < f^*$ ile dinamik adım uzunluğu (4.4) – 40000 iterasyon (a) $\ x_k - x^*\ $ (b) $f(x_k)$ iterasyonlar boyunca değişimi	59
Şekil 6.11.	PBC1 problemi – $f^{lev} < f^*$ ile dinamik adım uzunluğu (4.4) – 500000 iterasyon	60
Şekil 6.12.	Test 29’daki Problem17– f_k^{lev} ile dinamik adım uzunluğu (4.4) (a) $\ x_k - x^*\ $ (b) $f(x_k)$ iterasyonlar boyunca değişimi	61
Şekil 6.13.	Test 29’daki Problem17– Azalan adım uzunluğu (4.1) (a) $\ x_k - x^*\ $ (b) f_k iterasyonlar boyunca değişimi	62
Şekil 6.14.	Test 29’daki Problem17– Azalan adım uzunluğunun (4.1)	63
Şekil 6.15.	Test 29’daki Problem17– f^* ve x^* ile dinamik adım uzunluğu (4.2) (a) $\ x_k - x^*\ $ (b) $f(x_k)$ iterasyonlar boyunca değişimi.....	64
Şekil 6.16.	Problem 7– $f^{lev} < f^*$ ile dinamik adım uzunluğu (4.4) (a) $\ x_k - x^*\ $ (b) $f(x_k)$ iterasyonlar boyunca değişimi	65

SİMGE VE KISALTMALAR DİZİNİ

$\langle x, y \rangle$	: $x, y \in R^n$ vektörlerinin skaler çarpımı
$cl Y$	: $Y \subset R^n$ kümesinin kapanışı
$co Y$	: $Y \subset R^n$ kümesinin dışbükey örtüsü
$f'(x; g)$	: f fonksiyonunun g yönündeki yönlü türevi
$\partial f(x)$	: Subdifferansiyel
v	: Subgradyant vektörü
L	: Lipschitz Sabiti
(v, c)	: Zayıf subgradyant çifti
$\partial^w f(x)$	: Zayıf differansiyel
x^*	: Optimal çözüm
$f(x^*)$	: Optimal amaç fonksiyonu değeri
d_X	: X kümesinin diametri
α_k	: Adım uzunluğu
D	: Zayıf subgradyant çiftinin v vektörünün üst sınırı
f^{lev}	: Hedef amaç fonksiyonu değeri
f_k^{lev}	: k . iterasyondaki hedef amaç fonksiyonu değeri
λ	: Zayıf subgradyantın tahmini için parametre
α	: Zayıf subgradyantın tahmini için parametre
$G(e)$	: Birim hiperküp kümesi

1. GİRİŞ

Differansiyellenemeyen optimizasyon, amaç fonksiyonunun ve/veya kısıtların türevinin olmadığı problemlerle ilgilenir. Veri analizi, optimal kontrol problemleri, çeşitli mühendislik uygulamaları, görüntü onarma, doğrusal olmayan gelir vergisi problemi gibi bir çok gerçek hayat problemleri türevlenemeyen yapıdadır. Differansiyellenemeyen optimizasyon, 1950'lerde R.T. Rockafellar ve W. Fenchel tarafından verilen subdifferansiyel kavramına dayanmakta olup bu alandaki ilk çözüm yöntemleri 1960'larda V.V. Demyanov, N. Shor, Y. Ermolyev ve meslektaşları tarafından geliştirilmiştir. Dışbükey fonksiyonlar için tanımlanmış olan subgradyanlar, gradyanların genelleştirilmiş halidir ve fonksiyonun türevinin olmadığı noktalarda gradyan yerine subgradyan kullanılır. Subgradyanlar dışbükey fonksiyonun epigrafının destek hiperdüzlemlerin normal vektörleridir. Subgradyan metodları ise differansiyellenebilen optimizasyondaki gradyan yöntemlerinin genelleştirilmesidir. Differansiyellenebilir optimizasyondan farklı olarak keyfi subgradyan yönünde gitmek azalan yönü garanti etmez. Bu sebeple zigzaglar çizerek ilerler ve yakınsaklık hızı yavaştır. Ancak uygulaması kolaydır. Shor (1985) ise subgradyan metodunun dışbükey fonksiyon sınıfı ile sınırlı olduğunu eksiklik olarak belirtmiştir.

Azimov ve Kasimbeyli (Gasimov), 1999 yılında dışbükey olmayan ve differansiyellenemeyen optimizasyon problemlerine yönelik daha analitik bir yaklaşım ile subgradyan kavramını genelleştirerek ilk defa zayıf subgradyan kavramını tanımlamışlar ve böylece dışbükey olmayan optimizasyon problemlerinin optimal çözümlerinin hem karakterizasyonu hem de hesaplanması yönünde olanak sağlamışlardır. Aslında zayıf subgradyan kavramı, fonksiyonların grafiklerinin, dışbükey analizdeki hiperdüzlemler yerine konilerle desteklenmesi fikrine dayanmaktadır. Bu da aslında optimizasyon ve genellikle yöneylem araştırması problemleri ile ilgili uygulamalarda çok güçlü bir kıstas olan dışbükeylik koşulunun esnetilebilmesi yönünde önemli bir adım olmuştur. Bu kavram kullanılarak ilerleyen yıllarda, dışbükey olmayan ve türevlenemeyen optimizasyon problemlerinin optimal çözümlerinin araştırılması yönünde önemli çalışmalar yapılmıştır. Kısıtlı optimizasyon için etkili türevsiz (derivative-free) algoritma Kasimbeyli (Gasimov) (2002) tarafından geliştirilmiştir, daha sonra bu algoritma Kasimbeyli (Gasimov) ve Rubinov (2004) tarafından genelleştirilmiş ve Burachik vd. (2006) tarafından çeşitli yakınsaklık özellikleri önerilmiştir.

Bu tezin amacı,

$$\min_{x \in X} f(x)$$

$f: R^n \rightarrow R$ gerçel değerli fonksiyonu şeklinde tanımlanan differansiyellenemeyen sadece kutu kısıtlarına (box constraint) sahip olan ve dışbükey olmayan optimizasyon problemleri için zayıf subgradyant kavramını kullanan bir algoritma geliştirmektir. Ancak tıpkı subgradyantlarda olduğu gibi pratikte zayıf subgradyantları da hesaplamak kolay bir iş değildir. Bu sebeple öncelikle zayıf subgradyantların tahmini üzerinde çalışılmış ve zayıf subgradyantlar yönlü türevin zayıf subgradyantların supremumuna eşitliği kullanılarak tahmin edilmiştir. Daha sonra zayıf subgradyant algoritması geliştirilmiş ve sabit adım uzunluğu, azalan adım uzunluğu ve dört farklı dinamik adım uzunluğu olmak üzere toplam altı adet adım uzunluğu için algoritmanın yakınsaklık özellikleri araştırılmıştır. Geliştirilen zayıf subgradyant algoritması, Python programlama dilinde kodlanmıştır. Son olarak literatürde differansiyellenemeyen ve dışbükey olmayan sadece kutu kısıtlarına sahip olan optimizasyon problemleri geliştirilen zayıf subgradyant algoritması ile çözülmüş ve algoritmanın performansı üzerinde tartışılmıştır.

İkinci bölümde tez boyunca kullanılan genel gösterim ve tanımlar verilmiş, subdifferansiyel tanımı ve özellikleri ile subgradyant algoritmalarından ve zayıf subdifferansiyel tanımı ve özelliklerinden bahsedilmiştir. Üçüncü bölümde ise zayıf subdifferansiyelin nasıl tahmin edildiği önermeler ve ispatları ile açıklanmıştır. Dördüncü bölümde ise differansiyellenemeyen dışbükey olmayan kısıtsız optimizasyon problemleri için geliştirilen zayıf subgradyant algoritması verilmiş, ele alınan toplam altı adım uzunluğu için yakınsaklık özelliklerinin önermeleri ve ispatları açıklanmıştır. Beşinci bölümde çok fazla yerel optimal değere sahip bir fonksiyonda zayıf subgradyant algoritmasının çalışması, zayıf subdifferansiyelin c değerinin belirlenmesi, zayıf subgradyant algoritmasının iteratif ilerlemesi ve $f^{lev} > f^*$ ile dinamik adım uzunluğuyla $f^{lev} < f^*$ ile dinamik adım uzunluğunun birlikte nasıl kullanılabileceği analiz edilmiştir. Daha sonra altıncı bölümde literatürden alınan differansiyellenemeyen ve dışbükey olmayan sadece kutu kısıtlarına sahip olan optimizasyon problemleri, bu problemlerin geliştirilen zayıf subgradyant metodu ile elde edilen sonuçları verilmiş ve sonuçlar tartışılmıştır. Son olarak yedinci bölümde sonuçlardan ve gelecekte yapılması planlanan çalışmalardan bahsedilmiştir.

2. DIFFERANSELLNEMEYEN OPTİMİZASYON

Differansiyellenemeyen optimizasyon, differansiyellenemeyen fonksiyonların en küçüklenmesi veya en büyüklenmesi ile ilgilenir. Bu tip problemlerinin veri analizi, optimal kontrol problemleri, çeşitli mühendislik uygulamaları, görüntü onarma, doğrusal olmayan gelir vergisi problemi gibi bir çok pratik alanlarda uygulamaları vardır. Bu tezde

$$\min_{x \in X} f(x) \quad (P)$$

$f: R^n \rightarrow R$ gerçel değerli fonksiyonu şeklinde tanımlanan differansiyellenemeyen sadece kutu kısıtlarına (box constraint) sahip olan ve dışbükey olmayan optimizasyon problemleri ele alınacaktır.

Öncelikle tez boyunca kullanılan genel gösterim ve tanımlar verilmiş, subdifferansiyel tanımı ve özellikleri ile subgradyant algoritmalarından ve zayıf subdifferansiyel tanımı ve özelliklerinden bahsedilmiştir.

2.1. Gösterim ve Tanımlar

Bu bölümde tez boyunca kullanılan genel gösterim ve tanımlar verilecektir.

$R_+^n := \{y = (y_1, \dots, y_n): y_i \geq 0, i = 1, \dots, n\}$ olsun. Tez boyunca R_+^n negatif olmayan gerçel sayılar kümesini gösterecektir.

$x, y \in R^n$ olmak üzere $\langle x, y \rangle$ skaler çarpımı $\langle x, y \rangle = \sum_{i=1}^n x_i y_i$ olarak tanımlanır ve burada $x_i, y_i \in R$ gösterimi x ve y vektörlerinin i . bileşenleridir. $\|x\|$ olarak gösterilen norm için l_2, l_1 ve l_∞ normları sırası ile $\|x\|_2 = (\langle x, x \rangle)^{1/2}$, $\|x\|_1 = \sum_{i=1}^n |x_i|$ ve $\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|$ olarak hesaplanır.

$cl Y$ ve $co Y$ sırası ile $Y \subset R^n$ kümesinin kapanışını ve dışbükey örtüsünü göstermektedir.

Tanım 1: $Y \subset R^n$ kümesinin dışbükey örtüsü

$$co Y = \{x \in R^n: x = \sum_{i=1}^k \lambda_i x^i, \sum_{i=1}^k \lambda_i = 1, x^i \in Y, \lambda_i \geq 0, k > 0 \text{ ve tamsayı}\}$$

olarak tanımlanır.

Tanım 2: $f : R^n \rightarrow R$ fonksiyonu,

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) \quad \forall x, y \in R^n \quad \lambda \in [0,1]$$

ise dışbükeydir.

Tanım 3: $x \in R^n$ olsun.

$$|f(y) - f(x)| \leq L\|y - x\| \quad \forall x, y \in R^n$$

olacak şekilde bir L (Lipschitz sabiti) negatif olmayan sabit var ise $f: R^n \rightarrow R$ fonksiyonuna Lipschitz fonksiyonu denir.

Tanım 4: $x \in R^n$ olsun.

$$f(y) - f(x) \geq -L\|y - x\|$$

olacak şekilde bir L negative olmayan sabit sayı (Lipschitz sabiti) ve x noktasının bir $N(x)$ komşuluğu var ise $f: R^n \rightarrow R$ fonksiyonuna x noktasında alttan yerel Lipschitz fonksiyonu denir. Eğer bu eşitsizlik $\forall y \in R^n$ için sağlanırsa, f fonksiyonuna x noktasında alttan Lipschitz denir.

Tanım 5:

$$f'(x; g) = \lim_{\lambda \downarrow 0} \frac{f(x + \lambda g) - f(x)}{\lambda}$$

olarak tanımlanan limite f fonksiyonunun x noktasındaki $g \in R^n$ yönündeki yönlü türevi denir.

Tanım 6: $C \subseteq R^n$ kümesi tüm $x \in C$ için $\lambda > 0$ olmak üzere $\lambda x \in C$ ise konidir.

2.2. Subgradyant

Dışbükey fonksiyonlar için tanımlanmış olan subgradyantlar, gradyanların genelleştirilmiş halidir ve fonksiyonun türevinin olmadığı noktalarda gradyant yerine subgradyant kullanılır. Subgradyantlar dışbükey fonksiyonun epigrafının destek hiperdüzlemleridir.

Tanım 7: $f: R^n \rightarrow R$ fonksiyonu dışbükey olsun.

$$f(y) \geq f(x) + \langle v, y - x \rangle, \quad \forall y \in R^n \quad (2.1)$$

koşulunu sağlayan $v \in R^n$ vektörüne f fonksiyonunun x noktasındaki subgradyantı denir.

Denklem (2.1)'den görülmektedir ki f fonksiyonunun x noktasındaki subgradyantlarının koleksiyonu dışbükey kümedir (Bazaraa, Sherali ve Sheety, 2006). f fonksiyonunun x noktasındaki subgradyantlarının kümesine,

$$\partial f(x) = \{v \in R^n: f(y) \geq f(x) + \langle v, y - x \rangle, \forall y \in R^n\}$$

f fonksiyonunun x noktasındaki subdifferensiyali denir.

Teorem 1 (Bazaraa vd., 2006, Teorem 3.2.5, s.107) : $f: R^n \rightarrow R$ fonksiyonu dışbükey olsun. x noktası için, $epi f$ fonksiyonun epigrafını $(x, f(x))$ noktasında destekleyen

$$H = \{(y, \alpha): f(x) + \langle v, y - x \rangle = \alpha\}$$

hiperdüzleminde $v \in R^n$ vektörü vardır ve f fonksiyonunun x noktasındaki subgradyantıdır.

Teorem 2 (Shor, 1985, Teorem 1.7, s. 9) : $f: R^n \rightarrow R$ fonksiyonu dışbükey olsun. f fonksiyonunun x noktasındaki subdifferansiyeli sınırlı, dışbükey ve kapalıdır.

Sonuç 1 (Shor, 1985, Sonuç, s.12): $f: R^n \rightarrow R$ fonksiyonunun global minimumu $\bar{x} \in R^n$ noktası için $0 \in \partial f(\bar{x})$ olur.

Dışbükey fonksiyonların yönlü türevlerinin bu fonksiyonun subgradyantların noktasal maksimumu olarak gösterimi üzerine olan teoremi dışbükey analizin ana teoremlerinden biridir (Borwein, 1982; Shor, 1985; Borwein ve Lewis, 2000; Bagirov, Karmista ve Mäkelä, 2014).

Teorem 3 (Bagirov vd., 2014, Teorem 2.28, s. 43) : $f: R^n \rightarrow R$ dışbükey fonksiyon olsun. Tüm $x \in R^n$ için

- i. $f'(x, g) = \lim_{\lambda \rightarrow +0} \frac{f(x+\lambda g) - f(x)}{\lambda} \geq \langle v, g \rangle \quad \forall v \in \partial f(x), \forall g \in R^n$ ve
- ii. $f'(x; g) = \max \{\langle v, g \rangle: v \in \partial f(x)\} \quad \forall g \in R^n$

2.2.1. Subgradyant algoritması

Subgradyant algoritması 60'lı yılların ortasında tanımlanmıştır. Demyanov (1968), Polyak (1969a, 1969b, 1970) ve Shor (1985), bu konunun öncüleridir. Subgradyant metotları differansiyellenebilen optimizasyondaki gradyant yöntemlerinin genelleştirilmesidir. Differansiyellenebilir optimizasyonda kullanılan yöntemlerden

farklı olarak keyfi subgradyant yönünde gitmek azalan yönü garanti etmez. Bu sebeple zigzaglar çizerek ilerler ve yakınsaklık hızı yavaştır. Ayrıca dışbükey fonksiyonlar için geçerlidir. Ancak uygulaması kolaydır ve düşük depolama gereksinimine sahiptir.

Subgradyant algoritması;

$$x_{k+1} = x_k - \alpha_k v_k, \quad v_k \in \partial f(x_k)$$

formülüne göre $\{x_k\}$ dizisini üretir ve burada α_k k. iterasyondaki adım uzunluğudur. Standart yakınsaklık teoremlerinin elde edilebilmesi için α_k adım uzunluğunun 0'a yaklaşması gereklidir. Fakat adım uzunluğundaki azalma çok hızlı olmamalıdır. Adım uzunluğu üzerindeki klasik şart: $\alpha_k > 0, k \rightarrow \infty$ iken $\alpha_k \rightarrow 0$ ve $\sum_{k=1}^{\infty} \alpha_k = +\infty$ şeklindedir. Yakınsaklık teoremlerinin ispatı $d_k = \min\|x_k - x^*\|$ şeklinde tanımlanmış $\{d_k\}$ dizisinin analizine dayanır (Shor, 1985).

Bir (Temel) Subgradyant Algoritmasının Özeti (Bazaraa vd., 2006)

Başlangıç Adımı: Bir başlangıç çözümü seç $x_1 \in X$, mevcut üst sınırı $UB_1 = f(x_1)$ olacak şekilde belirle ve şimdiki iyi çözümü $x^* = x_1$ şeklinde ata, $k = 1$ yap ve ana adıma git.

Ana Adım: Verilen x_k için, x_k noktasında f fonksiyonunun $v_k \in \partial f(x_k)$ bir subgradyantını bul. Eğer $v_k = 0$ ise dur, x_k (veya x^*) (P) problemini çözer. Diğer durumda, α_k adım uzunluğunu seç, $x_{k+1} = x_k - \alpha_k v_k$ hesapla. Eğer $f(x_{k+1}) < UB_k$ ise $UB_{k+1} = f(x_{k+1})$ ve $x^* = x_{k+1}$. Diğer durumda $UB_{k+1} = UB_k$. k 'yi 1 arttır ve ana adımı tekrar et.

Algoritma subgradyantı keyfi olarak seçtiğinden, problemin optimum noktası olsa ve bulunsa bile durdurma kriteri $v_k = 0$ hiçbir zaman gerçekleşmeyebilir. İterasyona maksimum bir limit tanımlama sıklıkla kullanılan durdurma kriteridir.

Teorem 4 (Bazaraa vd., 2006, Teorem 8.9.2, s.437) : (P) probleminin optimum noktası olduğu ve $\{\lambda_k\} \rightarrow 0^+$ ve $\sum_{k=0}^{\infty} \lambda_k = \infty$ koşullarını sağlayan $\{\lambda_k\}$ adım uzunluğunun kullanıldığı varsayılın. Bu durumda algoritma sonlu optimal çözüm ile sonlanacaktır veya sonsuz dizi aşağıdakini üretecektir.

$$\{UB_k\} \rightarrow f^* \equiv \min\{f(x): x \in X\}$$

Teorem 5 (Bazaraa vd., 2006, Önerme 1.8.1, s.168) : $f: R^n \rightarrow R$ dışbükey fonksiyon olsun. f fonksiyonu $\bar{x}$ noktasında $\langle v, x - \bar{x} \rangle \geq 0 \quad \forall x \in R^n$ koşulunu sağlayan v subgradyanta sahip ise $\bar{x} \in R^n$ noktası problemin optimal çözümüdür.

Literatürde subgradyant algoritmaları, kısıtsız optimizasyon problemlerini çözmek için kullanılırken (Kiwiel 1983; Kim ve Ahn, 1991; Konnov, 2003; Kiwiel, 2004; Li vd. 2006; Nedic vd. 2008; Nedic ve Bertsekas, 2010; Neto vd. 2013; Hu, vd. 2015), lagrange çarpanları kullanılarak kısıtsız probleme dönüştürülen kısıtlı optimizasyon problemleri için de yoğun bir şekilde kullanılmıştır (Sherali ve Ulular, 1989; Bertsekas, 2001; Kiwiel, 2001; Kasımbeyli (Gasimov), 2002; Auslender ve Teboulle, 2004; Kasımbeyli (Gasimov)ve Rubinov, 2004; Ouorou, 2007; Filik ve Kurban, 2010; Ali ve Zhu, 2013).

Subgradyant algoritmalarının yakınsaklık özellikleri genellikle sabit, azalan ve farklı durumlar için dinamik adım uzunlukları ele alınarak araştırılmıştır (Ermolyev, 1966; Polyak, 1969a, 1969b; Goffin, 1977; Bazaraa ve Sherali, 1981; Allen vd., 1987; Kulikov ve Fazylov, 1990; Kim vd., 1991; Kim ve Um, 1993; Correa ve Lemaréchal, 1993; Brännlund, 1993; Kwiel, 1996a, 1996b; Bertsekas, 1999; Goffin and Kwiel, 1999; Kiwiel vd., 1998; Slodov ve Zavrier, 1998; Nedic, 2002). Sabit adım uzunluğu için elde edilen yakınsaklık özellikleri optimal çözümü elde etmede başarılı olamasa da diğer adım uzunlukları için optimal çözümü elde eden veya optimal çözüme oldukça yakın değerler ile yakınsaklık önermeleri ispatlanmıştır. Subgradyant tanımı dışbükeylik şartı altında geçerli olduğundan dolayı subgradyant algoritmalarının yakınsaklık sonuçları da sadece dışbükey optimizasyon problemleri için geçerli olmaktadır.

2.4. Zayıf Subgradyant

Klasik subdifferansiyelin geliştirilmesi olan zayıf subdifferansiyel kavramı Azimov ve Kasımbeyli (Gasimov) (1999, 2002) tarafından verilmiştir. Bu kavram analitik olarak verilen dışbükey olmayan kümeyi destekleyen konik yüzeylerin sınıfını verir. Benzer fikirden yola çıkılarak kısıtlı optimizasyon için etkili türevsiz (derivative-free) algoritma (Modified Subgradient Algorithm) Kasımbeyli (Gasimov) (2002) tarafından geliştirilmiştir. Daha sonra bu algoritma Kasımbeyli (Gasimov) ve Rubinov (2004) tarafından geliştirilmiş, Burachik vd. (2006) tarafından çeşitli yakınsaklık özellikleri önerilmiştir.

Tanım 8: $x \in R^n$ noktası $f(x)$ fonksiyonunun sonlu olduğu bir nokta olsun.

$$f(y) \geq f(x) + \langle v, y - x \rangle - c \|y - x\|, \quad \forall y \in R^n \quad (2.2)$$

koşulunu sağlayan $(v, c) \in R^n \times R_+$ çiftine f fonksiyonunun x noktasındaki zayıf subgradyantı denir

f fonksiyonunun x noktasındaki,

$$\partial^w f(x) = \{(v, c) \in R^n \times R_+ : f(y) \geq f(x) + \langle v, y - x \rangle - c \|y - x\|, \forall y \in R^n\}$$

tüm zayıf subgradyantları kümesine f fonksiyonunun x noktasındaki zayıf subdifferensiyali denir.

Eğer f fonksiyonu x noktasında subdifferensallenebilir ise f fonksiyonu x noktasında zayıf subdifferansiyellenebilir olduğu Açıkça görülmektedir (Azimov ve Kasimbeyli (Gasimov), 2002). Örneğin $v \in \partial f(x)$ ise her $c \geq 0$ için $(v, c) \in \partial^w f(x)$ olur. Eğer

$$g(y) = f(x) + \langle v, y - x \rangle - c \|y - x\| \quad (2.3)$$

şeklinde tanımlanan sürekli içbükey fonksiyon var ise Tanım 8'den $(v, c) \in R^n \times R_+$ çifti f fonksiyonunun x noktasındaki zayıf subgradyantıdır. Tüm $y \in R^n$ için $g(y) \leq f(y)$ ve $g(x) = f(x)$ olur.

$\text{hypo}(g) = \{(y, \alpha) \in R^n \times R : g(y) \geq \alpha\}$ kümesi köşesi (vertex) $(x, f(x))$ noktası olan kapalı konidir. Aslında,

$$\begin{aligned} \text{hypo}(g) - (x, f(x)) &= \{(y - x, \alpha - f(x)) \in R^n \times R : \langle v, y - x \rangle - c \|y - x\| \geq \alpha - f(x)\} \\ &= \{(u, \beta) \in R^n \times R : \langle v, u \rangle - c \|u\| \geq \beta\} \end{aligned}$$

elde ederiz. Böylece, (2.2) ve (2.3)'ten

$$\text{graph}(g) = \{(y, \alpha) \in R^n \times R : g(y) = \alpha\}$$

olarak elde edilen konik yüzey,

$$\text{epi}(f) = \{(x, \alpha) \in R^n \times R : f(x) \leq \alpha\}$$

için $(x, f(x))$ noktasında destek konisi olur bu bakımdan

$$\text{epi}(f) \subset \text{epi}(g) \text{ ve } \text{cl}(\text{epi}(f)) \cap \text{graph}(g) \neq \emptyset.$$

elde edilir.

Teorem 6 (Azimov ve Kasimbeyli (Gasimov), 2002, Teorem 1): $f: R^n \rightarrow R$ tanımlı $f(x)$ fonksiyonunun sonlu olduğu herhangi bir x noktası için aşağıdaki özellikler eşittir.

- (i) f fonksiyonu x noktasında zayıf subdifferensallenebilir,
- (ii) f fonksiyonu x noktasında alttan Lipschitzdir,
- (iii) f fonksiyonu x noktasında alttan yerel Lipschitzdir ve

$$f(y) \geq -p\|y\| + q \quad \forall y \in R^n$$

olacak şekilde $p \geq 0$ ve q sayıları vardır.

İzleyen teorem zayıf subdifferansiyelin en önemli özelliklerinden birini göstermektedir.

Teorem 7 (Kasimbeyli ve İnceoğlu, 2010, Teorem 3) : $f: R^n \rightarrow R$ fonksiyonunun $\partial^w f(x)$ zayıf subdifferansiyelinin boş olmadığını varsayalım. Bu durumda $\partial^w f(x)$ kümesi kapalı ve konvektir.

Önerme 1: $f: R^n \rightarrow R$ fonksiyonu $\bar{x} \in R^n$ noktasında zayıf subdifferansiyellenebilir ve L Lipschitz sabiti ile Lipschitz fonksiyon olsun. Her $(v, c) \in \partial^w f(\bar{x})$ için $0 < c \leq L$ olduğunu varsayalım. Öyleyse $\partial^w f(\bar{x})$ kümesi $R^n \times R_+$ 'nın bir kompakt altkümesidir.

İspat: $\varepsilon > 0$ keyfi pozitif sayısı alalım. f fonksiyonunun $\bar{x}$ noktasında sürekli olduğundan dolayı $\delta > 0$ sayısı

$$\|x - \bar{x}\| \leq \delta \text{ ise } |f(x) - f(\bar{x})| < \varepsilon$$

olacak şekilde vardır.

$$x(\delta, v) = \bar{x} + \frac{\delta v}{\|v\|}$$

noktasını düşünelim. Zayıf subgradyantin tanımından

$$\varepsilon > f\left(\bar{x} + \frac{\delta v}{\|v\|}\right) - f(\bar{x}) \geq \langle v, \bar{x} + \frac{\delta v}{\|v\|} - \bar{x} \rangle - c \left\| \bar{x} + \frac{\delta v}{\|v\|} - \bar{x} \right\|$$

$$\varepsilon > \delta \|v\| - c\delta$$

$$\frac{\varepsilon}{\delta} > \|v\| - c$$

$$\frac{\varepsilon}{\delta} + c > \|v\|$$

$c \leq L$ olduğundan

$$\frac{\varepsilon}{\delta} + L > \|v\|$$

elde edilir ve zayıf subdifferansiyelin sınırlılığı ispat edilmiş olur. Teorem 7'den zayıf subdifferansiyelin kapalı olduğu bilindiğine göre teoremin ispatı tamamlanmıştır. $\square$

Not 1 : Bütün zayıf subgradyanlar $(v, c) \in \partial^w f(\bar{x})$ için L Lipschitz sabiti olmak üzere $0 < c \leq L$ olduğu varsayılmıştır. L Lipschitz sabiti ile $\bar{x}$ noktasında alttan Lipschitz olan her f fonksiyonu için $(0, L) \in \partial^w f(\bar{x})$ olacaktır. Diğer taraftan eğer L değeri f fonksiyonu için $\bar{x}$ noktasında Lipschitz sabiti ise açıktır ki $M > L$ olan her M sayısı da bir Lipschitz sabitidir.

Önerme 2 (Kasimbeyli ve İnceoğlu, 2010, Önerme 3,): $f: R^n \rightarrow R$ fonksiyonu $\bar{x} \in R^n$ noktasında zayıf subdifferansiyellenebilen ve $\bar{x} \in R^n$ noktasında global minimum olsun. Bu durumda $(0,0) \in \partial^w f(\bar{x})$ olur.

Kasimbeyli ve Mammadov (2009) zayıf subgradyan kavramını kullanarak konveks analizin temel teoremlerinden biri olan dışbükey fonksiyonların yönlü türevlerinin bu fonksiyonun subgradyanların noktasal maksimumu olarak gösterimi üzerine olan teoremi dışbükey olamayan fonksiyonlar için genelleştirmişlerdir. Herhangi bir dışbükeylik varsayımı olmaksızın yönlü türevlenebilir fonksiyonların özel bir sınıfını sunmuşlar ve yönlü türevi zayıf subgradyanların noktasal supremumu olarak göstermişlerdir.

Lemma 1 (Kasimbeyli ve Mammadov, 2009, Lemma 2.9) : $f: R^n \rightarrow R$ fonksiyonu x noktasında zayıf subdifferansiyellenebilen olsun. Eğer f fonksiyonu bu noktada $g \in R^n$ yönünde yönlü differansiyellenebilir ise keyfi $(v, c) \in \partial^w f(x)$ çifti için,

$$f'(x; g) = \lim_{\lambda \rightarrow +0} \frac{f(x + \lambda g) - f(x)}{\lambda} \geq \langle v, g \rangle - c \|g\|$$

sağlanır.

Varsayım 1:

- $f: R^n \rightarrow R$ fonksiyonu $x \in R^n$ noktasında yönlü türevlenebilir olsun,
- f fonksiyonunun x noktasındaki $f'(x)$ yönlü türevi, O_{R^n} noktasının bir komşuluğunda alttan sınırlı olsun ve
- İzleyen koşul sağlansın.

$$f(y) - f(x) \geq f'(x)(y - x) \quad \forall y \in R^n$$

Teorem 9 (Kasimbeyli ve Mammadov, 2009, Teorem 4.5): $f: R^n \rightarrow R$ fonksiyonu için Varsayım 1 sağlansın. f fonksiyonu $x \in R^n$ noktasında zayıf subdifferansiyellenebilendir ve

$$f'(x; g) = \sup\{\langle v, g \rangle - c\|g\| : (v, c) \in \partial^w f(x)\} \quad \forall g \in R^n$$

eşitliği sağlanır.

3. ZAYIF SUBGRADYANTIN TAHMİNİ

Bagirov (1998) dışbükey fonksiyonların en küçüklenmesi için yeni bir yaklaşım geliştirmiştir ve bu yaklaşım dışbükey fonksiyonların subdifferensiyelinin sürekli tahminine dayanır. Bu sürekli tahmin kesikli gradyant kavramını (discrete gradient) (Bagirov, 1992; Bagirov, 1994; Bagirov, 2003) kullanır. Kesikli gradyant, sadece amaç fonksiyonunun değerlerini kullanır ki bu yöntemin en önemli özelliğidir. Kesikli gradyant verilen bir yöne göre tanımlanır ve böylece verilen fonksiyonun yönlü türevinin tahminine olanak sağlar. Bagirov, Karagözsen ve Sezer (2008) ise kesikli gradyant kavramını kullanarak tahmin ettikleri subgradyantlar ile bir türevsiz algoritma (derivative-free algorithm) geliştirmişlerdir.

Zayıf subgradyantların tahmini için yönlü türevin zayıf subgradyantların supremumuna eşit olmasından yola çıkılarak kesikli gradyant kavramından faydalanılmış ve gerekli önermeler kurularak ispatlanmıştır.

$C = \{c \in R_+ : \exists v: (v, c) \in \partial^w f(x)\}$ kümesi tanımlansın. Her $c \in C$ için $V_c = \{v \in R^n : (v, c) \in \partial^w f(x)\}$ kümesi vardır. Her V_c kümesi için $V_c = coA$ olacak şekilde $A = \{a^1, a^2, \dots, a^m\} \in R^n$ boş olmayan küme vardır.

$G = \{e \in R^n : e = (e_1, e_2, \dots, e_n), |e_j| = 1, j = \overline{1, n}\}$ kümesi R^n 'de birim hiperkübün tüm köşelerinin kümesi olsun. Her $e \in G$ için n vektörün $\alpha \in (0, 1]$ ile $e^j = e^j(\alpha) = (\alpha e_1, \alpha^2 e_2, \dots, \alpha^j e_j, 0, \dots, 0), j = \overline{1, n}$ sırasını düşünelim.

Aşağıdaki kümeler sunulmuştur.

$$R_0(e) = A$$

$$R_j(e) = \{v \in R_{j-1}(e) : v_j e_j = \max\{w_j e_j : w_j \in R_{j-1}(e)\}\}$$

Açıkça görülmektedir ki,

$$R_j(e) \neq \emptyset, \forall j = \overline{1, n}; R_j(e) \subset R_{j-1}(e), \forall j = \overline{1, n}.$$

Dahası,

$$v_r = u_r, \quad \forall u, v \in R_j(e), \quad r = 1, \dots, j \quad (3.1)$$

Önerme 3: $R_n(e)$ tek elemanlı kümedir.

İspat: İspat, denklem (3.1)'ü izler. $\square$

İzleyen kümeleri düşünelim:

$$R(x, e^j(\alpha)) = \{v \in A : \langle v, e^j \rangle = \max \{\langle v, e^j \rangle : v \in A\}\}$$

Herhangi bir $a \in A$ vektörü alınsın. Eğer $a \notin R_n(e)$ ise $a \in R_t(e)$, $t = 0, \dots, r-1$ ve $a \notin R_r(e)$ olacak şekilde $r \in \{1, \dots, n\}$ vardır. $a \notin R_r(e)$ ifadesinden $v_r e_r > a_r e_r \quad \forall v \in R_r(e)$ elde edilir. $a \in A, a \notin R_n(e)$ için $d(a) = v_r e_r - a_r e_r > 0$ ve $\bar{d} = \min \{d(a) : a \in A \setminus R_n(e)\}$ sayısı tanımlanır. A kümesi sonlu ve $d(a) > 0 \quad \forall a \in A \setminus R_n(e)$ olduğundan $\bar{d} > 0$ olur.

Zayıf subgradyant sınırlı olduğundan,

$$\|v\| \leq D, c \leq L$$

eşitsizlikleri yazılır.

$r, j \in \{1, \dots, n\}, r < j$ ele alınsın. Her $v, w \in V_c$ ve $\alpha \in (0, 1]$, için aşağıdaki eşitsizlik vardır.

$$\left| \sum_{t=r+1}^j (v_t - w_t) \alpha^{t-r} e_t \right| < 2D\alpha n$$

$\alpha_0 = \min \{1, \bar{d} / (4Dn)\}$ olsun. Öyleyse herhangi bir $\alpha \in (0, \alpha_0]$ için

$$\left| \sum_{t=r+1}^j (v_t - w_t) \alpha^{t-r} e_t \right| < \frac{\bar{d}}{2} \quad (3.2)$$

elde edilir.

Önerme 4 $R(x, e^j(\alpha)) \subset R_j(e)$, $j = 1, \dots, n \quad \forall \alpha \in (0, \alpha_0]$ olacak şekilde $\alpha_0 > 0$ vardır.

İspat: Tersini varsayalım. Öyleyse $y \notin R_j(e)$ olacak şekilde $y \in R(x, e^j(\alpha))$ vardır. Dolayısıyla $y \notin R_r(e)$ ve $y \in R_t(e)$ herhangi bir $t = 0, \dots, r-1$ olacak şekilde $r \in \{1, \dots, n\}$, $r \leq j$ vardır. Herhangi bir $v \in R_j(e)$ ele alalım. Denklem (3.1)'den $v_t e_t = y_t e_t, t = 1, \dots, r-1, v_r e_r \geq y_r e_r + \bar{d}$ elde ederiz. (3.2)'den aşağıdaki eşitlik elde edilir.

$$\begin{aligned} \langle v, e^j \rangle - \langle y, e^j \rangle &= \sum_{t=1}^j (v_t - y_t) \alpha^t e_t = \alpha^r \left[v_r e_r - y_r e_r + \sum_{t=r+1}^j (v_t - y_t) \alpha^{t-r} e_t \right] \\ &> \alpha^r \bar{d} / 2 > 0 \end{aligned}$$

$\langle y, e^j \rangle = \max \{ \langle u, e^j \rangle : u \in V_c \}$ ve $v \in V_c$ olduğundan, elde ederiz ki,

$$\langle y, e^j \rangle \geq \langle v, e^j \rangle > \langle y, e^j \rangle + \alpha^r \bar{d}/2$$

bu bir çelişkidir. $\square$

Sonuç 2:

$$f'(x, e^j(\alpha)) = f'(x, e^{j-1}(\alpha)) + v_j \alpha^j e_j - c \alpha^j \quad \forall \alpha \in (0, \alpha_0]$$

olacak şekilde $\alpha_0 > 0$ vardır.

İspat: Önerme 4 $R_j(x, e^j(\alpha)) \subset R_j(e) \quad j = \overline{1, n}$ olduğunu göstermektedir. Öyleyse

$$f'(x, e^j(\alpha)) - f'(x, e^{j-1}(\alpha)) = \langle v, e^j \rangle - c \|e^j\| - \langle v^0, e^{j-1} \rangle + c \|e^j - 1\|$$

olacak şekilde $v \in R_j(e)$, $v^0 \in R_{j-1}(e)$ vardır.

l_1 -normu için,

$$\|e^j(\alpha)\| = \alpha|e_1| + \alpha^2|e_2| + \dots + \alpha^j|e_j| = \alpha + \alpha^2 + \dots + \alpha^j \text{ olduğundan}$$

$$f'(x, e^j(\alpha)) - f'(x, e^{j-1}(\alpha)) = v_j \alpha^j e_j - c \alpha^j$$

elde edilir. $\square$

$e \in G$ ve $\lambda > 0, \alpha > 0$ verilen sayılar olsun. Aşağıdaki sayılar alınsın:

$$x^0 = x, \quad x^j = x^0 + \lambda e^j(\alpha), \quad j = \overline{1, n}$$

$(v, c) \in \partial^w f(x)$ çifti için aşağıdaki koordinatlarla $v = v(e, \alpha, \lambda) \in \mathbb{R}^n$ bir vektör olsun.

$$v_j - \frac{c}{e_j} = \frac{f(x^j) - f(x^{j-1})}{\lambda \alpha^j e_j} \quad j = \overline{1, n}$$

Herhangi bir sabit $e \in G$ ve $\alpha > 0$ için aşağıdaki kümeyi tanımlandı:

$$W(e, \alpha) = \{(w, c) \in \mathbb{R}^n \times \mathbb{R}_+ : \exists (\lambda_k \rightarrow +0, k \rightarrow +\infty), w = \lim_{k \rightarrow +\infty} v(e, \alpha, \lambda_k)\}$$

Önerme 5: $W(e, \alpha) \subset \partial^w f(x) \quad \forall \alpha \in (0, \alpha_0]$ olacak şekilde $\alpha > 0$ vardır.

İspat: $v = v(e, \alpha, \lambda)$ vektörünün tanımından

$$\begin{aligned}
v_j - \frac{c}{e_j} &= \frac{f(x^j) - f(x^{j-1})}{\lambda \alpha^j e_j} = \frac{f(x^j) - f(x) - [f(x^{j-1}) - f(x)]}{\lambda \alpha^j e_j} \\
&= \frac{\lambda f'(x, e^j) - \lambda f'(x, e^{j-1}) + o(\lambda, e^j) - o(\lambda, e^{j-1})}{\lambda \alpha^j e_j}
\end{aligned}$$

burada $\lambda^{-1}o(\lambda, e^i) \rightarrow 0$ $\lambda \rightarrow +0, i = j-1, j$ olarak ele alınmıştır. $w \in R_n(e)$ ele alalım. Denklem (3.1)'den w tektir. $w \in R_n(e)$ belirtmektedir ki $w \in R_j(e) \forall j \in \{1, \dots, n\}$ her kümenin elemanıdır. Sonuç 2'den görülmektedir ki her $\alpha \in (0, \alpha_0]$ için:

$$\begin{aligned}
v_j(e, \alpha, \lambda) - \frac{c}{e_j} &= \frac{\lambda (\alpha^j e_j w_j - c \alpha^j) + o(\lambda, e^j) - o(\lambda, e^{j-1})}{\lambda \alpha^j e_j} \\
&= w_j - \frac{c}{e_j} + \frac{o(\lambda, e^j) - o(\lambda, e^{j-1})}{\lambda \alpha^j e_j}
\end{aligned}$$

olacak şekilde $\alpha_0 > 0$ vardır.

Öyleyse, herhangi bir sabit $\alpha \in (0, \alpha_0]$ için,

$$\lim_{\lambda \rightarrow +0} |v_j(e, \alpha, \lambda) - \frac{c}{e_j} - (w_j - \frac{c}{e_j})| = 0$$

olur. Dolayısı ile $\lim_{\lambda \rightarrow +0} v(e, \alpha, \lambda) = w \in V_c$ $\square$

Özetle zayıf subgradyantı tahmin eden algoritma;

Algoritma-ZST:

Adım 1: $e \in G = \{e \in \mathbb{R}^n : e = (e_1, e_2, \dots, e_n), |e_j| = 1, j = \overline{1, n}\}$ olacak şekilde e vektörünü ve $\lambda > 0, \alpha \in (0, 1]$ sayılarını belirle. $(v, c) \in \partial^w f(x)$ çiftinin c değerini belirle.

Adım 2: $x^0 = x, x^j = x^0 + \lambda e^j(\alpha), j = \overline{1, n}$

Adım 3: $(v, c) \in \partial^w f(x)$ çiftinin v vektörü,

$$v_j - \frac{c}{e_j} = \frac{f(x^j) - f(x^{j-1})}{\lambda \alpha^j e_j} \quad j = \overline{1, n}$$

Örnek: $-x_1 + 2(x_1^2 + x_2^2 - 1) + 1.75|x_1^2 + x_2^2 - 1|$ fonksiyonunun $(1,0)$ noktasındaki zayıf subgradyanı izleyen şekilde hesaplanır.

Adım 1: $e = (1, -1)$ ve $\lambda = 0,1$; $\alpha = 0,9$ olsun. $(v, c) \in \partial^w f(x)$ çiftinin c değerini 3 olarak belirlenmiştir.

Adım 2: $x^0 = (1,0),$

$$x^1 = (1,0)^T + 0,1 * (0,9; 0)^T = (1,09; 0)^T, \quad j = 1$$

$$x^2 = (1,0)^T + 0,1 * (0,9; -0,81)^T = (1,09; -0,081)^T, \quad j = 2$$

Adım 3: $(v, c) \in \partial^w f(x)$ çiftinin v vektörü,

$$v_1 - \frac{3}{1} = \frac{f(x^1) - f(x^0)}{0,1 * 0,9^1 * 1} = \frac{(-0,384625) - (-1)}{0,09} = 0,615375 \quad j = 1$$

$$v_1 = 0,615375 + 3 = 3,615375 \quad j = 1$$

$$v_2 - \frac{3}{-1} = \frac{f(x^2) - f(x^1)}{0,1 * 0,9^2 * 1} = \frac{(-0,360021) - (-0,384625)}{0,081} = 0,024603 \quad j = 2$$

$$v_2 = 0,024603 - 3 = -2,975397 \quad j = 2$$

$$v = (0,615375; -2,975397)^T$$

$$(v, c) = (0,615375; -2,975397; 3)^T$$

4. DİFFERANSİYELLENEMEYEN DIŞBÜKEY OLMAYAN KISITSIZ OPTİMİZASYON PROBLEMLERİ İÇİN GELİŞTİRİLEN ZAYIF SUBGRADİENT YÖNTEMİ

Zayıf subdifferansiyel tanımını kullanılarak differansiyellenemeyen ve dış bükey olmayan kısıtsız optimizasyon problemleri için geliştirilen zayıf subgradyant algoritması;

Adım 1: Bir başlangıç çözümü seç $x_0 \in X$, mevcut üst sınırı $UB = f(x_0)$ olacak şekilde belirle ve şimdiki iyi çözümü $x^{eniyi} = x_0$ şeklinde ata. İterasyon sayısını belirle.

$$k=0$$

Adım 2: f fonksiyonunun x_k noktasındaki $(v_k, c_k) \in \partial^w f(x_k)$ bir zayıf subgradyantını Algoritma-ZST ile tahmin et.

Adım 3: α_k adım uzunluğunu belirle. (Tablo 4.1)

Adım 4: $x_{k+1} = P_X(x_k - \alpha_k v_k)$ hesapla.

Adım 5: Eğer $f(x_{k+1}) < UB$ ise $UB = f(x_{k+1})$ ve $x^{eniyi} = x_{k+1}$

$$k = k + 1$$

Adım 6: $k < \text{iterasyon sayısı}$ ise Adım 2'ye dön, değil ise DUR.

Burada P_X ile gösterilen X kümesi üzerine izdüşümüdür. X kümesi $a_j \leq x_j \leq b_j \forall j = 1, \dots, n$ olacak şekilde kutu kısıtları olarak tanımlanmıştır ve bu sebeple izdüşümü hesaplamak kolaydır.

İzleyen notasyonlar bu bölümde kullanılmıştır:

(N1): Her $(v_k, c_k) \in \partial^w f(x_k)$ için $\|v_k\| \leq D$ ve $c_k \leq L$

(N2): $\|x_k - x^*\| \leq d_X, d_X = \text{diam}(X) = \max_{x_1, x_2 \in X} \|x_1 - x_2\|$

(N3): x^* ve f^* sırası ile optimal çözümü ve optimal amaç fonksiyonu değerini gösterir.

İzleyen adım uzunlukları için zayıf subgradyant metodunun yakınsaklık özellikleri araştırılmıştır.

Tablo 4.1. Zayıf subgradyant algoritması için yakınsaklık özellikleri araştırılan adım uzunlukları

Adım Uzunluğu İsmi	Adım Uzunluğu
Sabit adım uzunluğu	α_k pozitif bir sayıdır.
Azalan adım uzunluğu	$\lim_{k \rightarrow \infty} \alpha_k = 0 \quad \sum_{k=0}^{\infty} \alpha_k = \infty$ (4.1)
f^* ve x^* ile dinamik adım uzunluğu	$\alpha_k = \gamma_k \frac{f(x_k) - f^* - c_k \ x_k - x^*\ }{\ v_k\ ^2} \quad 0 < \underline{\gamma} < \gamma_k \leq \bar{\gamma} < 2$ (4.2)
$f^{lev} > f^*$ ile dinamik adım uzunluğu*	$\alpha_k = \gamma_k \frac{f(x_k) - f^{lev} - c_k d_X}{\ v_k\ ^2} \quad 0 < \underline{\gamma} < \gamma_k \leq \bar{\gamma} < 2$ (4.3)
$f^{lev} < f^*$ ile dinamik adım uzunluğu*	$\alpha_k = \gamma_k \frac{f(x_k) - f^{lev} - c_k d_X}{\ v_k\ ^2} \quad 0 < \underline{\gamma} < \gamma_k \leq \bar{\gamma} < 1$ (4.4)
f_k^{lev} ile dinamik adım uzunluğu**	$\alpha_k = \gamma_k \frac{f(x_k) - f_k^{lev} - c_k d_X}{\ v_k\ ^2} \quad 0 < \underline{\gamma} < \gamma_k \leq \bar{\gamma} < 1$ $f_k^{lev} = \min_k f(x_k) - \delta_k$ $\delta_{k+1} = \begin{cases} \min\{\beta_2 \delta_k, \bar{\delta}\} & \text{eğer } f(x_{k+1}) < f_k^{lev} \\ \max\{\beta_1 \delta_k, \underline{\delta}\} & \text{eğer } f(x_{k+1}) > f_k^{lev} \end{cases}$ $1 < \beta_1, 0 < \beta_2 < 1 \text{ ve } \delta_0, \bar{\delta}, \underline{\delta} \text{ pozitif sabitlerdir.}$ (4.5)

* f^{lev} amaç fonksiyonu için verilen tahmini değer

** f_k^{lev} amaç fonksiyonu için dinamik olarak hesaplanan tahmini bir değer

İspatlarda kullanılacak olan optimal çözüm ile zayıf subgradyant algoritmasıyla türetilen bir nokta arasındaki uzaklığı gösteren eşitsizlik Lemma 2’de verilmiştir.

Zayıf subgradyant eşitsizliği aşağıdaki gibi tanımlanır.

$$\langle v_k, x_k - x \rangle \geq f(x_k) - f(x) - c_k \|x - x_k\| \quad (v_k, c_k) \in \partial^w f(x_k)$$

Lemma 2: Zayıf subgradyant tarafından türetilen $\{x_k\}$ dizisi için;

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\alpha_k [f(x_k) - f^* - c_k \|x^* - x_k\|] + \alpha_k^2 \|v_k\|^2$$

İspat:

$$\|x_{k+1} - x^*\|^2 = \|x_k - \alpha_k v_k - x^*\|^2$$

$$\|x_{k+1} - x^*\|^2 = \|x_k - x^*\|^2 - 2\alpha_k \langle v_k, x_k - x^* \rangle + \alpha_k^2 \|v_k\|^2$$

Zayıf subgradyant eşitsizliğinden

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\alpha_k [f(x_k) - f^* - c_k \|x^* - x_k\|] + \alpha_k^2 \|v_k\|^2$$

elde edilir. $\square$

Not 2: $0 \leq c_k \leq L$ olduğundan dolayı $\{c_k\}$ serisi azalan olarak seçilebilir. X kümesi kompakt küme olduğundan dolayı, $\{c_k d_X\}$ serisi ve $\{\|x^* - x_k\|\}$ serisinin azalan olduğu durumlarda $\{c_k \|x^* - x_k\|\}$ serisi de azalandır ve bir limit noktasına sahiptir.

4.1. Sabit Adım Uzunluğu ile Zayıf Subgradyant Algoritması

İzleyen önerme subgradyant algoritmasının sabit adım uzunluğu ile ilgili yakınsaklık sonucunun (Nedic ve Bertsekas, 2002, Önerme 2.1) genelleştirilmesi olup sabit adım uzunluğu ile zayıf subgradyant algoritmasının yakınsaklık sonucunu göstermektedir.

Önerme 6: Sabit adım uzunluğu α ile zayıf subgradyant tarafından türetilen $\{x_k\}$ dizisi için

$$(a) \text{ Eğer } f^* = -\infty \text{ ise } \liminf_{k \rightarrow \infty} f(x_k) = -\infty$$

$$(b) \text{ Eğer } f^* > -\infty \text{ ise } \liminf_{k \rightarrow \infty} f(x_k) \leq f^* + \alpha \frac{D^2}{2} + \lim_{k \rightarrow \infty} c_k d_X$$

İspat:

(a) ve (b) aynı anda ispatlanacaktır. Eğer tersini varsayarsak, $\varepsilon > 0$ olacak şekilde

$$\liminf_{k \rightarrow \infty} f(x_k) \geq f^* + \alpha \frac{D^2}{2} + \lim_{k \rightarrow \infty} c_k d_X + \varepsilon$$

eşitsizliği yazılır ve k_0 yeterince büyük olsun öyle ki tüm $k \geq k_0$ olacak şekilde bir alt dizisi için,

$$f(x_k) - f^* - c_k d_X \geq \alpha \frac{D^2}{2} + \varepsilon \quad \forall k \geq k_0 \quad (4.6)$$

elde edilir. Lemma 2'den $\alpha_k = \alpha$ olarak alınarak

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\alpha [f(x_k) - f^* - c_k \|x^* - x_k\|] + \alpha^2 \|v_k\|^2$$

(N1) ve (N2) ile

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\alpha [f(x_k) - f^* - c_k d_X] + \alpha^2 D^2$$

(4.6) kullanılarak

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\alpha \left[\alpha \frac{D^2}{2} + \varepsilon \right] + \alpha^2 D^2$$

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \alpha^2 D^2 - 2\alpha \varepsilon + \alpha^2 D^2$$

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \alpha^2 D^2 - 2\alpha \varepsilon + \alpha^2 D^2$$

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\alpha \varepsilon$$

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &\leq \|x_k - x^*\|^2 - 2\alpha \varepsilon \leq \|x_{k-1} - x^*\|^2 - 4\alpha \varepsilon \leq \dots \\ &\leq \|x_{k_0} - x^*\|^2 - 2(k+1-k_0)\alpha \varepsilon \end{aligned}$$

elde edilir ki yeterince k terimi ∞ gittikçe eşitsizliğin son terimi de ∞ gidecektir. Ancak $\|x_{k+1} - x^*\|^2$ terimi negatif olamaz. Öyleyse bu bir çelişkidir. $\square$

4.2. Azalan Adım Uzunluğu ile Zayıf Subgradyant Algoritması

İzleyen önerme subgradyant algoritmasının azalan adım uzunluğu ile ilgili yakınsaklık sonucunun (Nedic ve Bertsekas, 2002, Önerme 2.4) genelleştirilmesi olup azalan adım uzunluğu ile zayıf subgradyant algoritmasının yakınsaklık sonucunu göstermektedir.

Önerme 7: Azalan adım uzunluğu α_k (4.1) ile zayıf subgradyant algoritması tarafından türetilen $\{x_k\}$ dizisi için

$$\liminf_{k \rightarrow \infty} f(x_k) \leq f^* + \lim_{k \rightarrow \infty} c_k d_X$$

İspat:

Eğer tersini varsayarsak, $\varepsilon > 0$ olacak şekilde

$$\liminf_{k \rightarrow \infty} f(x_k) \geq f^* + \lim_{k \rightarrow \infty} c_k d_X + \varepsilon$$

eşitsizliği yazılır ve k_0 yeterince büyük olsun öyle ki tüm $k \geq k_0$ olacak şekilde bir altdizisi için

$$f(x_k) - f^* - c_k d_X \geq \varepsilon \quad \forall k \geq k_0 \quad (4.7)$$

elde edilir. Lemma 2'den (N2) ile

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\alpha_k [f(x_k) - f^* - c_k d_X] + \alpha_k^2 \|v_k\|^2$$

(4.7) kullanılarak

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\alpha_k \varepsilon + \alpha_k^2 \|v_k\|^2$$

$\alpha_k \rightarrow 0$ olduğundan dolayı $\varepsilon > \alpha_k \|v_k\|^2 \quad \forall k \geq k_0$ olacak şekilde k_0 yeterince büyük olduğunu varsayabiliriz. Öyleyse,

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\alpha_k \varepsilon + \alpha_k \varepsilon$$

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \alpha_k \varepsilon$$

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \alpha_k \varepsilon \leq \|x_{k-1} - x^*\|^2 - (\alpha_k + \alpha_{k-1}) \varepsilon \leq \dots$$

$$\leq \|x_{k_0} - x^*\|^2 - \sum_{j=k_0}^k \alpha_j \varepsilon$$

$\sum_{k=0}^{\infty} \alpha_k = \infty$ olduğundan dolayı eşitsizliğin son terimi de ∞ gidecektir. Ancak $\|x_{k+1} - x^*\|^2$ terimi negatif olamaz. Öyleyse bu bir çelişkidir $\square$

4.3. f^* ve x^* ile Dinamik Adım Uzunluğu ile Zayıf Subgradyant Algoritması

f^* ve x^* ile dinamik adım uzunluğu kullanılırken adım uzunluğunun pozitifliğinin korunabilmesi için $c_k < \frac{f(x_k) - f^*}{\|x_k - x^*\|^2}$ olacak şekilde seçilmelidir.

İzleyen önerme optimal çözüm ile f^* ve x^* ile dinamik adım uzunluğu ile zayıf subgradyant algoritması ile türetilen bir nokta arasındaki uzaklığı göstermektedir.

Önerme 8: x^* optimal çözüm olsun. Öyleyse (4.2) dinamik adım uzunluğu ile zayıf subgradyant algoritması tarafından türetilen $\{\|x_k - x^*\|^2\}$ serisi $\varphi \geq 0$ sayısına yakınsar.

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2$$

İspat: Lemma 2'den (4.2) kullanılarak

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\gamma_k \frac{[f(x_k) - f^* - c_k \|x^* - x_k\|]^2}{\|v_k\|^2} + \gamma_k^2 \frac{[f(x_k) - f^* - c_k \|x^* - x_k\|]^2}{\|v_k\|^2}$$

elde edilir. Açıkça görülmektedir ki eşitsizliğin son iki teriminin toplamı negatiftir. Böylece

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2$$

elde edilir. $\square$

İzleyen önerme f^* ve x^* ile dinamik adım uzunluğu ile zayıf subgradyant algoritmasının yakınsaklık sonucunu göstermektedir.

Önerme 9: (4.2) dinamik adım uzunluğu ile zayıf subgradyant algoritması tarafından türetilen $\{x_k\}$ serisi için

$$\liminf_{k \rightarrow \infty} f(x_k) \leq f^* + \lim_{k \rightarrow \infty} c_k \|x_k - x^*\|^2$$

İspat: Eğer tersini varsayarsak, $\varepsilon > 0$ olacak şekilde

$$\liminf_{k \rightarrow \infty} f(x_k) \geq f^* + \lim_{k \rightarrow \infty} c_k \|x_k - x^*\|^2 + \varepsilon$$

eşitsizliği yazılır ve k_0 yeterince büyük olsun öyle ki tüm $k \geq k_0$ olacak şekilde bir alt dizisi için

$$f(x_k) - f^* - c_k \|x_k - x^*\|^2 \geq \varepsilon \quad \forall k \geq k_0 \quad (4.8)$$

elde edilir. Lemma 2'den (4.2) dinamik adım uzunluğu kullanılarak,

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\gamma_k \frac{[f(x_k) - f^* - c_k \|x_k - x^*\|^2]^2}{\|v_k\|^2} + \gamma_k^2 \frac{[f(x_k) - f^* - c_k \|x_k - x^*\|^2]^2}{\|v_k\|^2}$$

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \gamma_k(2 - \gamma_k) \frac{[f(x_k) - f^* - c_k \|x_k - x^*\|^2]^2}{\|v_k\|^2}$$

(13) ve (N1) ile

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \gamma_k(2 - \gamma_k) \frac{\varepsilon^2}{D^2}$$

$0 < \underline{\gamma} < \gamma_k \leq \bar{\gamma} < 2$ kullanılarak

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \underline{\gamma}(2 - \bar{\gamma}) \frac{\varepsilon^2}{D^2}$$

ve tüm $k \geq k_0$ için

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &\leq \|x_k - x^*\|^2 - \underline{\gamma}(2 - \bar{\gamma}) \frac{\varepsilon^2}{D^2} \leq \|x_{k-1} - x^*\|^2 - 2\underline{\gamma}(2 - \bar{\gamma}) \frac{\varepsilon^2}{D^2} \leq \dots \\ &\leq \|x_{k_0} - x^*\|^2 - \underline{\gamma}(2 - \bar{\gamma}) \frac{(k + k_0 - 1)\varepsilon^2}{D^2} \end{aligned}$$

elde edilir. k terimi ∞ gittikçe eşitsizliğin son terimi de ∞ gidecektir. Ancak Önerme 8 göre $\{\|x_k - x^*\|^2\}$ monoton azalan bir dizidir ve bu dizi alttan sıfır ile sınırlıdır. Bu sebeple $\varphi \geq 0$ sayısına yakınsar. Öyleyse bu bir çelişkidir. $\square$

4.4. $f^{lev} > f^*$ ile Dinamik Adım Uzunluğu ile Zayıf Subgradyant Algoritması

$f^{lev} > f^*$ ile dinamik adım uzunluğu kullanılırken adım uzunluğunun pozitifliğinin korunabilmesi için $c_k < \frac{f(x_k) - f^{lev}}{d_X}$ olacak şekilde seçilmelidir.

İzleyen önerme optimal çözüm ile $f^{lev} > f^*$ ile dinamik adım uzunluğu ile zayıf subgradyant algoritması ile türetilen bir nokta arasındaki uzaklığı göstermektedir.

Önerme 10: x^* nın optimal çözüm olsun. $f(x_k) \geq f^{lev} \forall k$ olduğunu varsayalım. Öyleyse (4.3) dinamik adım uzunluğu ile zayıf subgradyant algoritması tarafından türetilen $\{\|x_k - x^*\|\}$ serisi $\varphi \geq 0$ sayısına yakınsar.

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2$$

İspat: Lemma 2'den $f^{lev} > f^*$ ile

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\alpha_k [f(x_k) - f^{lev} - c_k \|x_k - x^*\|^2] + \alpha_k^2 \|v_k\|^2$$

(4.3) kullanılarak

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\gamma_k \frac{[f(x_k) - f^{lev} - c_k \|x_k - x^*\|^2]^2}{\|v_k\|^2} + \gamma_k^2 \frac{[f(x_k) - f^{lev} - c_k \|x_k - x^*\|^2]^2}{\|v_k\|^2}$$

elde edilir. Açıkça görülmektedir ki eşitsizliğin son iki teriminin toplamı negatiftir. Böylece

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2$$

elde edilir. $\square$

İzleyen önerme bilinen $f^{lev} > f^*$ ile dinamik adım uzunluğu ile zayıf subgradyant algoritmasının yakınsaklık sonucunu göstermektedir.

Önerme 11: $f(x_k) \geq f^{lev} \forall k$ olduğunu varsayalım. Öyleyse (8) dinamik adım uzunluğu ile zayıf subgradyant algoritması tarafından türetilen $\{x_k\}$ serisi için

$$\liminf_{k \rightarrow \infty} f(x_k) \leq f^{lev} + \lim_{k \rightarrow \infty} c_k d_X$$

İspat: Eğer tersini varsayarsak, $\varepsilon > 0$ olacak şekilde

$$\liminf_{k \rightarrow \infty} f(x_k) \geq f^{lev} + \lim_{k \rightarrow \infty} c_k d_X + \varepsilon$$

eşitsizliği yazılır ve k_0 yeterince büyük olsun öyle ki tüm $k \geq k_0$ olacak şekilde bir altdizisi için

$$f(x_k) - f^* - c_k d_X \geq \varepsilon \quad \forall k \geq k_0 \quad (4.9)$$

elde edilir. Lemma 2'den $f^{lev} \geq f^*$ ve (N2) kullanılarak,

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\alpha_k [f(x_k) - f^{lev} - c_k d_X] + \alpha_k^2 \|v_k\|^2$$

(8) adım uzunluğu ile

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &\leq \|x_k - x^*\|^2 - 2\gamma_k \frac{[f(x_k) - f^{lev} - c_k d_X]^2}{\|v_k\|^2} + \gamma_k^2 \frac{[f(x_k) - f^{lev} - c_k d_X]^2}{\|v_k\|^2} \\ \|x_{k+1} - x^*\|^2 &\leq \|x_k - x^*\|^2 - \gamma_k(2 - \gamma_k) \frac{[f(x_k) - f^{lev} - c_k d_X]^2}{\|v_k\|^2} \end{aligned}$$

(14) ve (N1) ile

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \gamma_k(2 - \gamma_k) \frac{\varepsilon^2}{D^2}$$

$0 < \underline{\gamma} < \gamma_k \leq \bar{\gamma} < 2$ kullanılarak

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \underline{\gamma}(2 - \bar{\gamma}) \frac{\varepsilon^2}{D^2}$$

ve tüm $k \geq k_0$ için

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &\leq \|x_k - x^*\|^2 - \underline{\gamma}(2 - \bar{\gamma}) \frac{\varepsilon^2}{D^2} \leq \|x_{k-1} - x^*\|^2 - 2\underline{\gamma}(2 - \bar{\gamma}) \frac{\varepsilon^2}{D^2} \leq \dots \\ &\leq \|x_{k_0} - x^*\|^2 - \underline{\gamma}(2 - \bar{\gamma}) \frac{(k + k_0 - 1)\varepsilon^2}{D^2} \end{aligned}$$

elde edilir. k terimi ∞ gittikçe eşitsizliğin son terimi de ∞ gidecektir. Ancak Önerme 10 göre $\{\|x_k - x^*\|^2\}$ monoton azalan bir dizidir ve bu dizi alttan sıfır ile sınırlıdır. Bu sebeple $\varphi \geq 0$ sayısına yakınsar. Öyleyse bu bir çelişkidir. $\square$

4.5. $f^{lev} < f^*$ ile Dinamik Adım Uzunluğu ile Zayıf Subgradyant Algoritması

$f^{lev} < f^*$ ile dinamik adım uzunluğu kullanılırken adım uzunluğunun pozitifliğinin korunabilmesi için $c_k < \frac{f(x_k) - f^{lev}}{d_X}$ olacak şekilde seçilmelidir.

İzleyen önerme optimal çözüm ile $f^{lev} < f^*$ ile dinamik adım uzunluğu ile zayıf subgradyant algoritması ile türetilen bir nokta arasındaki uzaklığı göstermektedir.

Önerme 12: x^* 'nın optimal çözüm olsun. Öyleyse (4.4) dinamik adım uzunluğu ile zayıf subgradyant algoritması tarafından türetilen $\{\|x_k - x^*\|\}$ serisi $\varphi \geq 0$ sayısına yakınsar.

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2$$

İspat: Lemma 2'den (N2) ve (4.4) adım uzunluğu ile

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\gamma_k \frac{[f(x_k) - f^{lev} - c_k d_X]}{\|v_k\|^2} [f(x_k) - f^* - c_k d_X] + \gamma_k^2 \frac{[f(x_k) - f^{lev} - c_k d_X]^2}{\|v_k\|^2}$$

elde edilir. $f^{lev} < f^*$ ise $f(x_k) - f^{lev} - c_k d_X > f(x_k) - f^* - c_k d_X$ olacaktır. Öyleyse

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\gamma_k \frac{[f(x_k) - f^* - c_k d_X]^2}{\|v_k\|^2} + \gamma_k^2 \frac{[f(x_k) - f^{lev} - c_k d_X]^2}{\|v_k\|^2}$$

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \gamma_k \frac{[2(f(x_k) - f^* - c_k d_X)^2 - \gamma_k (f(x_k) - f^{lev} - c_k d_X)^2]}{\|v_k\|^2}$$

elde edilir. $A_k = f(x_k) - f^* - c_k d_X$ ve $B_k = f(x_k) - f^{lev} - c_k d_X$ olsun. Öyleyse kısaca

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \gamma_k \frac{[2A_k^2 - \gamma_k B_k^2]}{\|v_k\|^2}$$

şeklinde yazılabilir. $2A_k^2 - \gamma_k B_k^2 \geq 0$ olması istenmektedir. $B_k > A_k$ olduğu biliniyor. B_k değeri c_k 'nin seçiminden dolayı her zaman pozitifdir. Ancak bu durum A_k değerinin her zaman pozitif olmasını garanti etmez. İki koşul ortaya çıkar. (i) $B_k^2 \geq A_k^2$ veya (ii) $A_k^2 \geq B_k^2$.

(i) İncelenecek olursa;

$$2A_k^2 - \gamma_k B_k^2 \geq 0$$

$$B_k^2 \geq A_k^2$$

eşitsizlikleri beraber göz önüne alınırsa $A_k^2 \geq (\gamma_k - 1)B_k^2$ elde edilir ki bu eşitsizlik $\gamma_k < 1$ olduğu için her zaman doğrudur.

(ii) İncelenecek olursa;

$$2A_k^2 - \gamma_k B_k^2 \geq 0$$

$$A_k^2 \geq B_k^2$$

eşitsizlikleri beraber göz önüne alınırsa $A_k^2 \geq \frac{(\gamma_k+1)}{3} B_k^2$ elde edilir ki bu eşitsizlik $\gamma_k < 1$ olduğu için her zaman doğrudur.

Öyleyse açıkça görülmektedir ki eşitsizliğin son terimi negatiftir. Böylece

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2$$

elde edilir. $\square$

İzleyen önerme bilinen $f^{lev} > f^*$ için dinamik adım uzunluğu ile zayıf subgradyant algoritmasının yakınsaklık sonucunu göstermektedir.

Önerme 13: (4.4) dinamik adım uzunluğu ile zayıf subgradyant algoritması tarafından türetilen $\{x_k\}$ serisi için

$$\liminf_{k \rightarrow \infty} f(x_k) \leq f^* + (f^* - f^{lev}) + \lim_{k \rightarrow \infty} c_k d_X$$

İspat: Eğer tersini varsayarsak, $\varepsilon > 0$ olacak şekilde

$$\liminf_{k \rightarrow \infty} f(x_k) \geq f^* + (f^* - f^{lev}) + \lim_{k \rightarrow \infty} c_k d_X + \varepsilon$$

eşitsizliği yazılır ve k_0 yeterince büyük olsun öyle ki tüm $k \geq k_0$ olacak şekilde bir altdizisi için

$$f(x_k) - 2f^* + f^{lev} - c_k d_X \geq \varepsilon \quad (4.10)$$

elde edilir. Lemma 2'den (N2) ve (4.4) adım uzunluğu ile

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\gamma_k \frac{[f(x_k) - f^{lev} - c_k d_X]}{\|v_k\|^2} [f(x_k) - f^* - c_k d_X] + \gamma_k^2 \frac{[f(x_k) - f^{lev} - c_k d_X]^2}{\|v_k\|^2}$$

elde edilir. $\gamma_k < 1$ olduğundan $\gamma_k^2 < \gamma_k$ elde edilir. Öyleyse

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\gamma_k \frac{[f(x_k) - f^{lev} - c_k d_X]}{\|v_k\|^2} [f(x_k) - f^* - c_k d_X] + \gamma_k \frac{[f(x_k) - f^{lev} - c_k d_X]^2}{\|v_k\|^2}$$

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \gamma_k (f(x_k) - f^{lev} - c_k d_X) \frac{[2(f(x_k) - f^* - c_k d_X) - (f(x_k) - f^{lev} - c_k d_X)]}{\|v_k\|^2}$$

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2$$

$$- \gamma_k (f(x_k) - f^{lev} - c_k d_X + f^{lev} - f^{lev} + 2f^* - 2f^*) \frac{[f(x_k) - 2f^* + f^{lev} - c_k d_X]}{\|v_k\|^2}$$

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2$$

$$- \gamma_k (f(x_k) - 2f^* + f^{lev} - c_k d_X + 2(f^* - f^{lev})) \frac{[f(x_k) - 2f^* + f^{lev} - c_k d_X]}{\|v_k\|^2}$$

(15) ve (N1) ile

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \gamma_k[2(f^* - f^{lev}) + \varepsilon] \frac{\varepsilon}{D^2}$$

$0 < \underline{\gamma} < \gamma_k \leq \bar{\gamma} < 1$ kullanılarak

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \underline{\gamma}[2(f^* - f^{lev}) + \varepsilon] \frac{\varepsilon}{D^2}$$

Ve tüm $k \geq k_0$ için

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &\leq \|x_k - x^*\|^2 - \underline{\gamma}[2(f^* - f^{lev}) + \varepsilon] \frac{\varepsilon}{D^2} \leq \|x_k - x^*\|^2 - 2\underline{\gamma}[2(f^* - f^{lev}) + \varepsilon] \frac{\varepsilon}{D^2} \leq \\ \dots &\leq \|x_k - x^*\|^2 - (k + k_0 - 1)\underline{\gamma}[2(f^* - f^{lev}) + \varepsilon] \frac{\varepsilon}{D^2} \end{aligned}$$

elde edilir. k terimi ∞ gittikçe eşitsizliğin son terimi de ∞ gidecektir. Ancak Önerme 12 göre $\{\|x_k - x^*\|^2\}$ monoton azalan bir dizidir ve bu dizi alttan sıfır ile sınırlıdır. Bu sebeple $\varphi \geq 0$ sayısına yakınsar. Öyleyse bu bir çelişkidir. $\square$

4.6. f_k^{lev} ile Dinamik Adım Uzunluğu ile Zayıf Subgradyant Algoritması

f_k^{lev} ile dinamik adım uzunluğu kullanılırken adım adım uzunluğunun pozitifliğinin korunabilmesi için $c_k < \frac{f(x_k) - f_k^{lev}}{d_X}$ olacak şekilde seçilmelidir. İzleyen önerme f_k^{lev} ile dinamik adım uzunluğu ile zayıf subgradyant algoritmasının yakınsaklık sonucunu göstermektedir.

Önerme 6: (4.5) dinamik adım uzunluğu ile zayıf subgradyant tarafından türetilen $\{x_k\}$ serisi için

$$\liminf_{k \rightarrow \infty} f(x_k) \leq f^* + \bar{\delta} + \lim_{k \rightarrow \infty} c_k d_X$$

İspat: Eğer tersini varsayarsak, $\varepsilon > 0$ olacak şekilde

$$\liminf_{k \rightarrow \infty} f(x_k) \geq f^* + \bar{\delta} + \lim_{k \rightarrow \infty} c_k d_X + \varepsilon$$

eşitsizliği yazılır ve k_0 yeterince büyük olsun öyle ki tüm $k \geq k_0$ olacak şekilde bir altdizisi için

$$f(x_k) \geq f^* + \bar{\delta} + c_k d_X + \varepsilon$$

$$f(x_k) - f^* - c_k d_X \geq \bar{\delta} + \varepsilon \tag{4.11}$$

elde edilir. Lemma 2'den (N2) ve (4.5) adım uzunluğu ile

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\gamma_k \frac{[f(x_k) - f_k^{lev} - c_k d_X]}{\|v_k\|^2} [f(x_k) - f^* - c_k d_X] + \gamma_k^2 \frac{[f(x_k) - f_k^{lev} - c_k d_X]^2}{\|v_k\|^2}$$

elde edilir. $\gamma_k < 1$ olduğundan $\gamma_k^2 < \gamma_k$ elde edilir. Bu durumda

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\gamma_k \frac{[f(x_k) - f_k^{lev} - c_k d_X]}{\|v_k\|^2} [f(x_k) - f^* - c_k d_X] + \gamma_k \frac{[f(x_k) - f_k^{lev} - c_k d_X]^2}{\|v_k\|^2}$$

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \gamma_k (f(x_k) - f_k^{lev} - c_k d_X) \frac{[2(f(x_k) - f^* - c_k d_X) - (f(x_k) - f_k^{lev} - c_k d_X)]}{\|v_k\|^2}$$

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \gamma_k (f(x_k) - f_k^{lev} - c_k d_X) \frac{[(f(x_k) - f^* - c_k d_X) - (f^* - f_k^{lev})]}{\|v_k\|^2}$$

(4.11) ve (N1) ile

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \gamma_k (f(x_k) - f_k^{lev} - c_k d_X) \frac{[\bar{\delta} + \varepsilon - (f^* - f_k^{lev})]}{D^2} \quad (4.12)$$

(4.5) adım uzunluğuna göre $f_k^{lev} = \min_k f(x_k) - \delta_k$ göre belirlenir. Her zaman $\min_k f(x_k) \geq f^*$ olacaktır. Öyleyse,

$$f_k^{lev} \geq f^* - \delta_k$$

eşitsizliği elde edilir. (4.5) adım uzunluğundaki δ_k belirlenmesine göre $\delta_k \leq \bar{\delta}$ olacaktır. Öyleyse,

$$f_k^{lev} \geq f^* - \bar{\delta}$$

$$\bar{\delta} \geq f^* - f_k^{lev}$$

eşitsizliği elde edilecektir. Bu eşitsizlik (4.12) denkleminde kullanılırsa,

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \gamma_k (f(x_k) - f_k^{lev} - c_k d_X) \frac{[\bar{\delta} + \varepsilon - \bar{\delta}]}{D^2}$$

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \gamma_k (f(x_k) - f_k^{lev} - c_k d_X) \frac{\varepsilon}{D^2}$$

ve adım uzunluğunun pozitifliği için $f(x_k) - f_k^{lev} - c_k d_X > \delta, \delta > 0$ olmalıdır.

Öyleyse,

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \gamma_k \delta \frac{\varepsilon}{D^2}$$

elde edilir.

$0 < \underline{\gamma} < \gamma_k \leq \bar{\gamma} < 1$ kullanılarak

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \underline{\gamma} \delta \frac{\varepsilon}{D^2}$$

Ve tüm $k \geq k_0$ için

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \underline{\gamma} \delta \frac{\varepsilon}{D^2} \leq \|x_{k-1} - x^*\|^2 - 2\underline{\gamma} \delta \frac{\varepsilon}{D^2} \leq \dots \leq \|x_{k_0} - x^*\|^2 - (k + k_0 - 1)\underline{\gamma} \delta \frac{\varepsilon}{D^2}$$

elde edilir. k terimi ∞ gittikçe eşitsizliğin son terimi de ∞ gidecektir. Ancak $\|x_{k+1} - x^*\|^2$ terimi negatif olamaz. Öyleyse bu bir çelişkidir. $\square$

5. ZAYIF SUBGRADYANT ALGORİTMASININ ANALİZİ

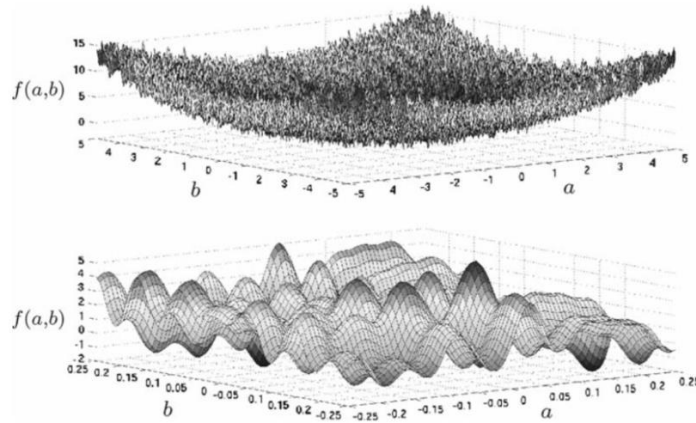
Bu bölümde, çok fazla yerel optimal değere sahip bir fonksiyonda zayıf subgradyant algoritmasının nasıl çalıştığı, zayıf subdiffereansiyelin c değerinin nasıl belirlenebileceği, zayıf subgradyant algoritmasının iteratif ilerlemesinin nasıl olduğu ve $f^{lev} > f^*$ ile dinamik adım uzunluğuyla (4.3) $f^{lev} < f^*$ ile dinamik adım uzunluğunun (4.4) birlikte nasıl kullanılabileceği analiz edilmiştir.

5.1. Yerel Optimal Çözümlere Sahip Analitik Bir Problem

Zayıf subgradyant algoritmasının yerel optimal çözümlere takılmadığını göstermek amacı ile Audet ve ark. (2008) çalışmasında yer alan problem (5.1) ele alınmıştır.

$$\begin{aligned} f(a,b) = & e^{\sin 50a} + \sin(60e^b) + \sin(70 \sin(a)) + \sin(\sin(80b)) \\ & - \sin(10(a+b)) + \frac{1}{4}(a^2 + b^2) \end{aligned} \quad (5.1)$$

Şekil 5.1’de fonksiyonun grafiğinden de problemin çok sayıda yerel optimal çözüme sahip olduğu görülmektedir.



Şekil 5.1. Analitik fonksiyonunun (5.1) grafiği; sınırlar her bir değişken için $[-5,5]$ ve $[-0.25,0.25]$ (Audet ve ark., 2008)

Problemin optimal noktası yaklaşık olarak $x^* = [-0,024; 0,211]$ olup, bu noktadan oldukça uzakta olan $x^1 = [3,3]$ noktasından başlanmış, kutu kısıtı olarak her bir değişken için $[-5,5]$ aralığı seçilmiş ve problem her bir adım uzunluğu ile

çözölmüştür. Sonular Tablo 5.1’de kullanılan parametreler ile birlikte verilmiştir. Sonular göstermektedir ki zayıf subgradyant algoritması yerel optimal özömlere takılmadan global optimal özöme yakınsamaktadır.

Tablo 5.1. Analitik problemin zayıf subgradyant algoritması ile sonuları

f^*	Sabit Adım Uzunluğu	Azalan Adım Uzunluğu (4.1)	f^* ve x^* ile Dinamik Adım Uzunluğu (4.2)	$f^{lev} > f^*$ ile Dinamik Adım Uzunluğu (4.3)	$f^{lev} < f^*$ ile Dinamik Adım Uzunluğu (4.4)	f_k^{lev} ile Dinamik Adım Uzunluğu (4.5)
-3,307	-3,305 ¹	-3,293 ²	-3,307 ³	-3,178 ⁴	-3,305 ⁵	-3,307 ⁶
CPU	0,621	0,739	0,786	0,409	0,738	17,556

$$^1 \alpha_k = 0,01; c_k = 1 - \left(\frac{1}{iter_{sayı}} * k \right); \lambda = 0,1; \alpha = 1$$

$$^2 \alpha_k = 1 - \left(\frac{1}{iter_{sayı}} * k \right); c_k = 1 - \left(\frac{1}{iter_{sayı}} * k \right); \lambda = 1, \alpha = 1$$

$$^3 c_k = c_k^{üst} * 0,5; \lambda = 0,1; \alpha = 1$$

$$^4 c_k = c_k^{üst} * 0,85; \lambda = 0,1; \alpha = 1; f^{lev} = -2,807$$

$$^5 c_k = c_k^{üst} * 0,4; \lambda = 0,001; \alpha = 1; f^{lev} = -3,807$$

$$^6 c_k = c_k^{üst} * 0,4; \lambda = 0,1; \alpha = 0,5; \delta_1 = f(x_1) * 0,15; \underline{\delta} = \delta_1 * 1,15, \beta_1 = 1,5; \beta_2 = 0,5$$

5.2. Zayıf Subgradyant Algoritmasında c Parametresinin Belirlenmesi için Bir Yaklaşım

Bir zayıf subdifferansiyel çiftinin (v, c) çiftinin tahmini için önerilen algoritmada c için bir değeri belirlenmekte ve bu değere göre v vektörünün değeri hesaplanmaktadır. c değeri belirlenmesi de önemli bir konudur. Zayıf subgradyant algoritmasında yeni bir nokta $x_{k+1} = x_k - \alpha_k v_k$ şeklinde üretilir. Sabit ve azalan adım uzunlukları için başlangıta belli bir değeri başlayarak her bir iterasyonda azalacak şekilde c değeri belirlenebilirken, dinamik adım uzunluklarında adım uzunluğunun pozitifliğini koruyabilmesi için c değeri her bir iterasyonda değışen üst değeri bulunmaktadır. Bu sebeple dinamik adım uzunluklarında c parametresi bu üst değeri küçük olacak şekilde seçilmelidir.

Tablo 5.2. Adım uzunluklarında c_k alabileceği en büyük değer

Adım uzunluğu	$c_k^{üst}$
Sabit adım uzunluğu	L
Azalan adım uzunluğu	L
Bilinen x^* ve f^* için dinamik adım uzunluğu	$c_k^{üst} = \frac{f(x_k) - f^*}{\ x_k - x^*\ }$
$f^{lev} > f^*$ için dinamik adım uzunluğu	$c_k^{üst} = \frac{f(x_k) - f^{lev}}{d_x}$
$f^{lev} < f^*$ için dinamik adım uzunluğu	$c_k^{üst} = \frac{f(x_k) - f^{lev}}{d_x}$
f_k^{lev} ile dinamik adım uzunluğu	$c_k^{üst} = \frac{f(x_k) - f_k^{lev}}{d_x}$

Ele alınan problem diferansiyellenebilir ve dış bükey olmadığı durumda c değeri daha deterministik bir yol ile belirlenebilir. v_k ve dinamik adım uzunluklarında α_k değerleri c_k değerinin alacağı değere bağlıdır. Bu sebeple $x_{k+1} = x_k - \varphi(c_k)$ olarak yazılabilir ve $c_k = \arg \min_{0 \leq c_k < c_k^{üst}} f(x_{k+1})$ olarak elde edilebilir. Burada $\min f(x_{k+1})$ probleminin çözülmesi içinde bir yöntemle ihtiyaç duyulacaktır. Problemimiz tek değişkenli ve dışbükey olmayan yapıda olacaktır. Ancak differansiyellenebilir olduğunda, dışbükey, diferansiyellenebilir ve tek değişkenli problemler için geliştirilmiş olan ikiye bölme (bisection) veya Newton yöntemi gibi yöntemler seçilebilir. Tabi ki bu yöntemler ile elde edilen c değeri global değil yerel en iyi çözüm olacaktır. c değerinin global en iyi olması çok kritik bir durum değildir.

Analitik fonksiyonu (5.1) zayıf subgradyant algoritmasının sabit adım uzunluğu ile c_k hem $c_k = 1 - \left(\frac{1}{iter_{sayı}} * k\right)$ hem de c_k için üst limit olmaksızın $c_k = \arg \min_{0 \leq c_k} f(x_{k+1})$ ile çözüldüğünde aldığı değerler ilk 10 iterasyon için Tablo 5.3'te verilmiştir. $c_k = \arg \min_{0 \leq c_k} f(x_{k+1})$ belirlenirken Newton yöntemi kullanılmış ve başlangıç değeri $c_k^0 = 1 - \left(\frac{1}{iter_{sayı}} * k\right)$ olarak alınmıştır. Eğer c_k^0 değerin ikinci türevi $f''(c_k^0) < 0$ ise $c_k^0 = c_k^0 + 1$ olacak şekilde ikinci türev $f''(c_k^0) > 0$ oluncaya kadar arttırılmıştır.

Tablo 5.3. Analitik fonksiyon (5.1) için sabit adım uzunluğunda c_k hesaplanması

İterasyon numarası	$c_k = 1 - \left(\frac{1}{iter_{sayı}} * k\right)$	$c_k = \arg \min_{0 \leq c_k} f(x_{k+1})$
1	0,999975	1,91901485
2	0,99995	7,07320017
3	0,999925	1,97764944
4	0,9999	2,06609044
5	0,999875	2,16577651
6	0,99985	0,01
7	0,999825	1,71961652
8	0,9998	1,01568591
9	0,999775	13,9300902
10	0,99975	3,15408563
f_{eniye}	-3,305	-3,305
CPU	0,621	8,252

* $\alpha_k = 0,01$; $\lambda = 0,1$; $\alpha = 1$

Tablo 5.3'e göre elde edilen c_k değerleri birbirinden farklı olmasına rağmen her iki durumda da optimal çözüme yaklaşık olarak -3,305 değeri ile yakınsamıştır. Ancak $c_k = 1 - \left(\frac{1}{iter_{sayı}} * k\right)$ eşitliğindeki değerler her problemin optimal çözümüne yakınsaklığı için farklılık gösterebilecek iken $c_k = \arg \min_{0 \leq c_k} f(x_{k+1})$ eşitliği her problem için aynı olacaktır. Tabi ki $c_k = \arg \min_{0 \leq c_k} f(x_{k+1})$ ile c_k değerinin hesaplanması çözüm için gereken süreyi arttıracaktır.

Sabit adım uzunluğuna benzer şekilde diğer adım uzunlukları içinde araştırılmış ve sonuçlar izleyen tablolarda verilmiştir.

Tablo 5.4. Analitik fonksiyon (5.1) için azalan adım uzunluğunda (4.1) c_k hesaplanması

İterasyon numarası	$c_k = 1 - \left(\frac{1}{iter_{sayı}} * k\right)$	$c_k = \arg \min_{0 \leq c_k} f(x_{k+1})$
1	0,999975	2,99787267
2	0,99995	0,96346753
3	0,999925	1,99701875
4	0,9999	3,83551829
5	0,999875	3,00609791
6	0,99985	2,9851226
7	0,999825	2,00064583
8	0,9998	3,01986325
9	0,999775	2,99811525
10	0,99975	1,98045477
f_{eniye}	-3,293	-3,305
CPU	0,739	5,127

* $\alpha_k = 1 - \left(\frac{1}{iter_{sayı}} * k\right)$; $\lambda = 1$, $\alpha = 1$

Tablo 5.5. Analitik fonksiyon (5.1) için f^* ve x^* ile dinamik adım uzunluğunda (4.2) c_k hesaplanması

İterasyon numarası	$c_k = c_k^{\text{üst}} * 0,5$	$c_k = \arg \min_{0 \leq c_k < c_k^{\text{üst}}} f(x_{k+1})$
1	0,97574981	0,97574981
2	0,65642992	1,0642967
3	0,85166745	0,85166745
4	0,67348532	0,67348532
5	0,9283385	0,98459709
6	1,09270696	1,02003141
7	1,10609762	1,00009749
8	1,35342631	1,99032599
9	0,98234948	1,02731181
10	1,14139731	1,14139731
f^{eni}	-3.307	-3,305
CPU	0,786	20,47

* $c_k = c_k^{\text{üst}} * 0,5$; $\lambda = 0,1$; $\alpha = 1$

Tablo 5.6. Analitik fonksiyon (5.1) için $f^{\text{lev}} > f^*$ ile dinamik adım uzunluğunda (4.3) c_k hesaplanması

İterasyon numarası	$c_k = c_k^{\text{üst}} * 0,85$	$c_k = \arg \min_{0 \leq c_k < c_k^{\text{üst}}} f(x_{k+1})$
1	0,26615567	0,26615567
2	0,20739024	0,20739024
3	0,1793718	0,1793718
4	0,33910008	0,33910008
5	0,30267989	0,01
6	0,23667707	0,01
7	0,28242685	0,28242685
8	0,21021407	0,21021407
9	0,39397747	0,01
10	0,26230167	0,26230167
f^{eni}	-3.178	-3,093
CPU	0,409	2,014

* $c_k = c_k^{\text{üst}} * 0,85$; $\lambda = 0,1$; $\alpha = 1$; $f^{\text{lev}} = -2,807$

Tablo 5.7. Analitik fonksiyon (5.1) için $f^{lev} < f^*$ ile dinamik adım uzunluğunda (4.4) c_k hesaplanması

İterasyon numarası	$c_k = c_k^{üst} * 0,4$	$c_k = \arg \min_{0 \leq c_k < c_k^{üst}} f(x_{k+1})$
1	0,301511	0,301511
2	0,33456346	0,33456346
3	0,33277345	0,33277345
4	0,3239299	0,3239299
5	0,21352716	0,21352716
6	0,31558717	0,31558717
7	0,35057517	0,35057517
8	0,28140975	0,28140975
9	0,32405658	0,32405658
10	0,24752852	0,01
f^{eniye}	-3,305	-3,254
CPU	0,738	18,429

* $c_k = c_k^{üst} * 0,4$; $\lambda = 0,001$; $\alpha = 1$; $f^{lev} = -3,807$

Tablo 5.8. Analitik fonksiyon (5.1) için f_k^{lev} ile dinamik adım uzunluğunda (4.5) c_k hesaplanması

İterasyon numarası	$c_k = c_k^{üst} * 0,4$	$c_k = \arg \min_{0 \leq c_k < c_k^{üst}} f(x_{k+1})$
1	0,02002959	0,02002959
2	0,01702515	0,01702515
3	0,02553772	0,02553772
4	0,08255205	0,01
5	0,06686446	0,06686446
6	0,01702515	0,01702515
7	0,12920326	0,12920326
8	0,02553772	0,02553772
9	0,08484934	0,08484934
10	0,05626588	0,05626588
f^{eniye}	-3,307	-3,307
CPU	17,556	45,279

* $c_k = c_k^{üst} * 0,4$; $\lambda = 0,1$; $\alpha = 0,5$; $\delta_1 = f(x_1) * 0,15$; $\underline{\delta} = \delta_1 * 1,15$; $\beta_1 = 1,5$; $\beta_2 = 0,5$

Tablo 5.4-5.9 incelendiğinde azalan adım uzunluğu için c_k değerleri birbirinden farklılık gösterirken, dinamik adım uzunlukları için çoğu zaman aynı çıkmıştır. Bunun sebebi ise c_k değerinin üstten $c_k^{üst}$ değeri ile sınırlı olması ve $\arg \min_{0 \leq c_k < c_k^{üst}} f(x_{k+1})$ değerinin $c_k^{üst}$ değerinden büyük çıksa bile adım uzunluğunun pozitifliğini olması için kullanılamamasıdır.

5.3. Zayıf Subgradyant Algoritmasının İteratif Gösterimi

Zayıf subgradyant algoritmasının iteratif nasıl çalıştığını göstermek için yine Analitik fonksiyon (5.1) kullanılmış ve f_k^{lev} ile dinamik adım uzunluğu optimal çözüme olan uzaklığın ($\|x_k - x^*\|$), f_k^{lev} ve $f(x_k)$ değerinin nasıl değiştiği Tablo 5.4'te verilmiştir. Optimal çözüme olan uzaklık ($\|x_k - x^*\|$) genel olarak her bir iterasyonda azalmış ancak arada bazı iterasyonlarda ufak artmalar olmuştur. f_k^{lev} değeri ise $f(x_k)$ değerinin değişimine göre aynı kalmış, azalmış veya artmıştır. $f(x_k)$ değeri ise problemin dışbükey olmayan yapısından dolayı her bir iterasyonda bir öncekine göre azalmış veya artmış ve 9917. iterasyonda optimal değere yakınsamıştır.

Tablo 5.9. Analitik fonksiyonun (5.1) f_k^{lev} ile dinamik adım uzunluğu (4.5) ile çözümünün iteratif gösterimi

İterasyon numarası	$\ x_k - x^*\ $	f_k^{lev}	$f(x_k)$
1	4,1143	4,0129	5,0922
3	4,1135	4,1191	5,1368
4	4,1126	4,1191	6,6597
5	4,1066	4,1191	5,7220
6	4,0983	4,1191	4,0312
7	4,0933	3,1283	2,5743
8	4,0928	1,2199	2,8769
9	4,0913	1,8971	4,5505
10	4,0725	1,9723	3,6162
40	3,9235	1,9153	3,6958
41	3,9191	1,9153	2,7792
42	3,9142	1,9153	3,5122
58	3,9098	1,9153	6,1661
59	3,9124	1,9153	2,9038
60	3,9138	1,9153	3,7637
71	3,9624	1,9153	1,4383
72	3,9648	0,5354	2,9511
73	3,9680	0,8363	1,5590
3079	1,6182	-3,0143	-2,0426
3080	1,6145	-3,0143	-2,2019
3081	1,6137	-3,0143	-1,9214
7995	0,0031	-3,8548	-3,2720
7996	0,0028	-3,8739	-3,2776
7997	0,0030	-3,8795	-3,2702
7998	0,0011	-3,8795	-3,3048
9915	0,0017	-3,9060	-3,2915
9916	0,0023	-3,9060	-3,2823
9917	0,0004	-3,9060	-3,3068

* $c_k = c_k^{ist} * 0,4$; $\lambda = 0,1$; $\alpha = 0,5$; $\delta_1 = f(x_1) * 0,15$; $\underline{\delta} = \delta_1 * 1,15$, $\beta_1 = 1,5$; $\beta_2 = 0,5$

5.3. Dinamik Adım Uzunluğu (4.3) ve (4.4)'ün Birlikte Kullanılması

Ele alınan bir problemin optimal değeri bilinmediği için belirlenecek olan f^{lev} değerinin optimal değerden f^* büyük veya küçük olduğunu bilinemez. Bu sebeple $f^{lev} > f^*$ ile dinamik adım uzunluğu (4.3) ile $f^{lev} < f^*$ ile dinamik adım uzunluğu (4.4) beraber kullanılarak problemin optimal çözümüne ulaşılabilir. Bu durum için izleyen adımlar önerilmiştir.

Adım 1: $f^{lev}, \epsilon > 0, \varepsilon > 0$ değerlerini belirle. $k=1$.

Adım 2: Zayıf subgradyant algoritması ile problemi çöz.

Adım 3: Eğer bulunan değer $f_k^{eniyi} > f^{lev} + \epsilon$ ise $f^{lev} = f^{lev} + \frac{|f^{lev} - f_k^{eniyi}|}{2}$

Eğer bulunan değer $f_k^{eniyi} < f^{lev}$ ise $f^{lev} = f^{lev} - 0,5$

Adım 4: Eğer $|f_k^{eniyi} - f_{k-1}^{eniyi}| < \varepsilon$ ise DUR.

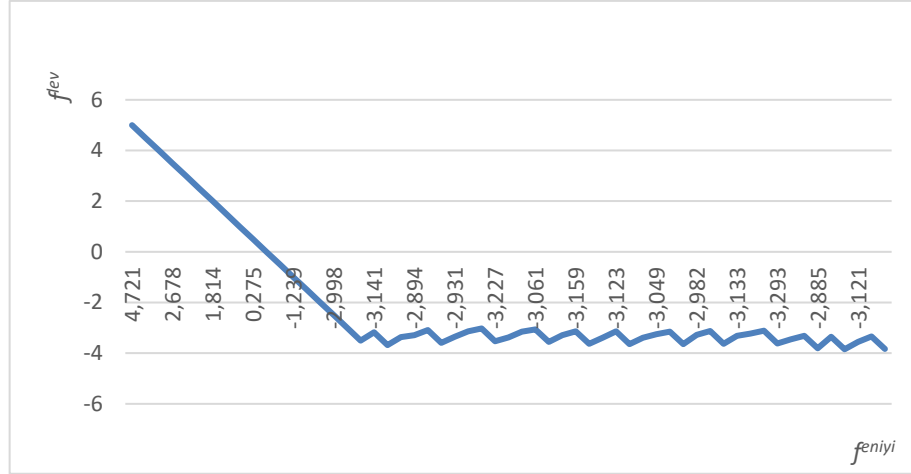
Diğer durumda, $k = k + 1$ ve ADIM 2'ye GİT.

Analitik problem (5.1) başlangıç $f^{lev} = 5$ olacak şekilde ve $\epsilon = 0,1$; $\varepsilon = 0.001$ değerleri ile çözülmüştür. Zayıf subgradyant algoritmasının dinamik adım uzunlukları (4.3) ve (4.4) için gereken parametreleri ise $c_k = c_k^{üst} * 0,4$; $\lambda = 0,1$; $\alpha = 1$ olarak belirlenmiştir. Elde edilen sonuçlar adım adım Tablo 5.10'da verilmiştir.

Tablo 5.10. Dinamik adım uzunlukları (4.3) ve (4.4)'ün beraber kullanımı

İterasyon numarası	f^{lev}	f^{eniyi}	CPU	İterasyon numarası	f^{lev}	f^{eniyi}	CPU
1	5,000	4,721	0,00003	30	-3,152	-2,968	0,71485
2	4,500	4,087	0,00007	31	-3,060	-3,061	0,15915
3	4,000	2,599	0,00006	32	-3,560	-3,021	0,72241
4	3,500	2,678	0,00006	33	-3,291	-2,979	0,70487
5	3,000	2,512	0,00030	34	-3,135	-3,159	0,29053
6	2,500	1,130	0,00088	35	-3,635	-3,141	0,70103
7	2,000	1,814	0,00081	36	-3,388	-2,890	0,71320
8	1,500	0,991	0,00034	37	-3,139	-3,123	0,73831
9	1,000	0,999	0,00211	38	-3,639	-3,150	0,71344
10	0,500	0,275	0,00333	39	-3,394	-3,098	0,73373
11	0,000	-0,115	0,00245	40	-3,246	-3,049	0,71564
12	-0,500	-1,107	0,00122	41	-3,147	-3,248	0,16314
13	-1,000	-1,239	0,00298	42	-3,647	-2,896	0,72546
14	-1,500	-1,663	0,00248	43	-3,272	-2,982	0,71421
15	-2,000	-2,001	0,01231	44	-3,127	-3,128	0,50417
16	-2,500	-2,998	0,08042	45	-3,627	-3,002	0,73086
17	-3,000	-3,000	0,02228	46	-3,314	-3,133	0,73114
18	-3,500	-2,853	0,75363	47	-3,224	-3,004	0,71317
19	-3,177	-3,141	0,71301	48	-3,114	-3,119	0,03752
20	-3,677	-3,054	0,71775	49	-3,614	-3,293	0,70786
21	-3,365	-3,215	0,73106	50	-3,453	-3,165	0,73360
22	-3,290	-2,894	0,73074	51	-3,309	-3,306	0,72317
23	-3,092	-3,094	0,11525	52	-3,809	-2,885	0,72671
24	-3,592	-3,123	0,72075	53	-3,347	-3,300	0,70129
25	-3,358	-2,931	0,71912	54	-3,847	-3,256	0,72547
26	-3,144	-2,903	0,72612	55	-3,552	-3,121	0,73986
27	-3,024	-3,024	0,03212	56	-3,337	-3,272	0,72324
28	-3,524	-3,227	0,70441	57	-3,837	-3,272	0,72577
29	-3,375	-2,929	0,70933				

57 iterasyonun sonunda algoritma durmuş ve 51. İterasyonda -3,306 değeri ile optimal sonuca yakınsamıştır. 18. iterasyona kadar f^{lev} değeri azalmış, daha sonraki iterasyonlarda f^{eniyi} değerine göre f^{lev} değeri bir önceki iterasyona göre artma ve azalma göstermiştir. f^{lev} ve f^{eniyi} değerlerinin değişimi izleyen grafikte görülmektedir.



Şekil 5.2. . f^{lev} ve f^{eniyi} değerlerinin değişimi

5.4. Yakınsaklığı Etkileyen Parametreler

Zayıf subgradyant algoritmasının yakınsaklık sonucunu etkileyen $\alpha_k, c_k, \lambda, x^0$ olmak üzere dört önemli parametre bulunmaktadır.

Tablo 5.11. Her bir adım uzunluğunun yakınsaklık sonuçlarını etkileyen değerler

Adım Uzunluğu İsmi	Yakınsaklığı Etkileyen Değerler	Adım Uzunluğu İsmi	Yakınsaklığı Etkileyen Değerler
Sabit adım uzunluğu	$\alpha_k, D, \lim_{n \rightarrow \infty} c_k d_X$	$f^{lev} > f^*$ ile dinamik adım uzunluğu	$f^{lev}, \lim_{n \rightarrow \infty} c_k d_X$
Azalan adım uzunluğu	$\lim_{n \rightarrow \infty} c_k d_X$	$f^{lev} < f^*$ ile dinamik adım uzunluğu	$(f^* - f^{lev}), \lim_{n \rightarrow \infty} c_k d_X$
f^* ve x^* ile dinamik adım uzunluğu	$\lim_{n \rightarrow \infty} c_k \ x_k - x^*\ $	f_k^{lev} ile dinamik adım uzunluğu	$\bar{\delta}, \lim_{n \rightarrow \infty} c_k d_X$

- Yakınsaklık teoremlerine bakılacak olursa her bir adım uzunluğu için ayrı ayrı yakınsaklık sonuçları elde edilmiştir ve Tablo 6.12’de bu yakınsaklık sonuçlarını etkileyen değerler verilmiştir. Her bir adım uzunluğunda bu değerlerin uygun seçilmesi önem taşımaktadır.
- c_k parametresi Tablo 6.12’den de görüleceği üzere her bir adım uzunluğunun yakınsaklık sonucunda önem taşımaktadır. Ayrıca c_k parametresini doğru seçmek zayıf subgradyantın doğru tahmin edilmesi demektir.

- λ parametresi yakınsaklık sonuçlarında yer almasa bile zayıf subgradyantın doğru tahmin edilmesinde etkili bir parametredir ve zayıf subgradyant algoritmasının iyi çalışmasında zayıf subgradyantın doğru tahmin edilmesi önem taşımaktadır.
- x^0 başlangıç çözümü iterasyon sayısı yeterince büyük alındığı takdirde ve zayıf subgradyant doğru tahmin edildiği takdirde yakınsaklığı etkilememektedir

6. HESAPSAL SONUÇLAR

Zayıf subgradyant algoritması Python programlama dilinde kodlanmış, problemler 2.5 GHz dört çekirdekli ve 16GB RAM sahip bilgisayarda çözdürülmüştür. Zayıf subgradyant algoritması için belirlenen parametreler Tablo 6.1’de verilmiştir.

Tablo 6.1. Zayıf subgradyant algoritması için belirlenen parametreler

Parametre İsmi	Zayıf subgradyantların tahmini için λ değeri	Sabit adım uzunluğu	Azalan adım uzunluğu	Sabit ve azalan adım uzunluğu için c değeri	Dinamik adım uzunlukları için c değeri
A	0,01 / 0,001 / 0,0001	0,001	$5/(2 * k)$	$10/k$	$c_k^{üst} * 0,9$
B	0,01 / 0,001 / 0,0001	0,001	$2 - \left(\frac{2}{iter_{sayı}}\right) * k$	$10 - \left(\frac{10}{iter_{sayı}}\right) * k$	$c_k^{üst} * 0,8$
C	0,01 / 0,001 / 0,0001	0,001	$1 - \left(\frac{1}{iter_{sayı}}\right) * k$	$5 - \left(\frac{5}{iter_{sayı}}\right) * k$	$c_k^{üst} * 0,85$
D	0,01 / 0,001 / 0,0001	0,001	$10 - \left(\frac{10}{iter_{sayı}}\right) * k$	$1 - \left(\frac{1}{iter_{sayı}}\right) * k$	$c_k^{üst} * 0,5$
E	0,01 / 0,001 / 0,0001	0,01	$1 - \left(\frac{1}{iter_{sayı}}\right) * k$	$1 - \left(\frac{1}{iter_{sayı}}\right) * k$	$c_k^{üst} * 0,4$

Zayıf subgradyant algoritması ile literatürde yer alan problemler çözümlerken 5 farklı parametre seti belirlenmiştir. Her bir parametre seti zayıf subgradyantların tahmini için λ değeri 0,01; 0,001 ve 0,0001 olarak belirlenmiş, zayıf subgradyantların tahmini için α değeri her bir parametre seti için 1 olarak alınmıştır. Her bir parametre seti ile problemler çözülmüş ve her bir adım uzunluğu için problemlerin bulunan en iyi sonuçları ilgili tablolarda verilmiş ve hangi parametre set ile elde edildiği yanına yazılmıştır. Zayıf subgradyantların tahmini için λ değeri 0,01 için parametre adı*, 0,001 için parametre adı ** ve 0,0001 için parametre adı *** olarak gösterilmiştir. $f^{lev} > f^*$ ile dinamik adım uzunluğu (4.3) için $f^{lev} = f_{opt} + 0,5$, $f^{lev} < f^*$ ile dinamik adım uzunluğu (4.4) için $f^{lev} = f_{opt} - 0,5$ olarak alınmıştır. f_k^{lev} ile dinamik adım uzunluğu (4.5) için $\delta_1 = f(x_1) * 0.15$; $\underline{\delta} = \delta_1 * 1.15$, $\beta_1 = 1,5$; $\beta_2 = 0,5$ olarak alınmıştır.

Zayıf subgradyant algoritması ile bulunan sonuçları için değerlendirme ölçütü Bagirov ve ark. (2014) olduğu gibi f_{ZSM} zayıf subgradyant algoritması ile bulunan sonuç olmak üzere $\left(\frac{f_{ZSM} - f^*}{1 + |f^*|}\right)$ olarak belirlemiştir. Küçük boyutlu ve değişken sayısı 50 olduğu problemlerde bu değer 5×10^{-4} ve 10^{-3} değerlerinden küçük olduğunda, değişken

sayısı 200 olduğu problemlerde ise 10^{-3} değerinden küçük olduğunda optimal değere ulaşıldığı kabul edilmiştir.

6.1. Küçük Boyutlu Test Problemleri

Differansiyellenemeyen, dışbükey olmayan ve sadece kutu kısıtlarına sahip olan küçük boyutlu test problemleri 19 adet olup isimleri, değişken sayıları, amaç fonksiyonları, optimal amaç fonksiyonu değeri, optimal çözümü ve başlangıç noktaları Ek 2’de verilmiştir. Kutu kısıtı olarak her bir değişken için $[-5,5]$ aralığı seçilmiş, eğer optimal çözüm değeri bu aralığın dışında kalıyor ise aralık $[x^* - 5, x^* + 5]$ olacak şekilde değiştirilmiştir. Her bir problem bu tez kapsamında yakınsaklık özellikleri araştırılan adım uzunlukları kullanılarak zayıf subgradyant algoritması ile çözülmüş, optimal sonuçlar ile kıyaslanmıştır.

Tablo 6.2. Küçük boyutlu test problemlerinin sonuçları

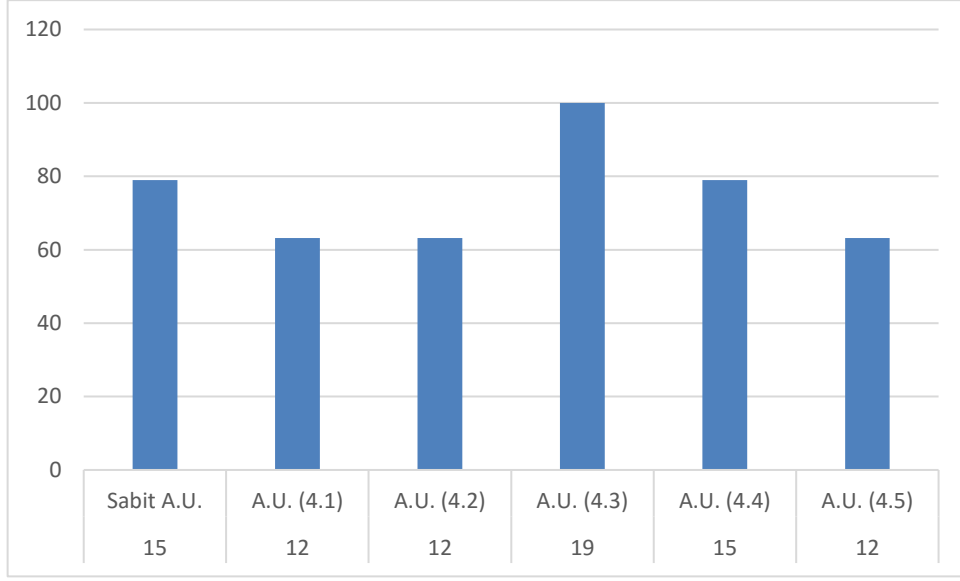
İsim	f^*	Sabit Adım Uzunluğu	Azalan Adım Uzunluğu (4.1)	f^* ve x^* ile Dinamik Adım Uzunluğu (4.2)	$f^{lev} > f^*$ ile Dinamik Adım Uzunluğu (4.3)	$f^{lev} < f^*$ ile Dinamik Adım Uzunluğu (4.4)	f_k^{lev} ile Dinamik Adım Uzunluğu (4.5)
Crescent	0	0 a**	0 a**	0 a**	0.5 a**	0 a**	0 a**
Mifflin 2	-1	-1 a**	-1 a**	-1 a**	-0.5 a**	-1 a**	-1 a**
WF	0	0 a*	0 a**	0.00071218 a**	0.5 a**	0 a**	0 a**
SPIRAL	0	0 a**	0 a**	0 a*	0.5 a**	0 a*	0 a*
EVD52	3.5991193	3,5991193 a*	3.59972865 a**	3,5997266 a**	4,0997193 a**	3,59972443 a**	3,59984305 a**
PBC3	0.0042021427	0.0042075 a**	0.006310416 e**	0.00420890708 e *	0.250397 a**	0.00420867 a**	0.00421077 a**
Bard	0,050816327	0.05087519 a****	0.05089106 a****	0.0508719 b*	0.5 a****	0.05086314 a**	0.0508552 a**
Polak 6	-44	-43.99 a**	-43.99 a**	-43.99 a**	-43.5 a**	-43.99 a**	-43.99 a**
El-Attar	0.5598131	0.55984302 a**	0.55990149 b**	0.55981921 b**	1.0598131 a**	0.55981367 a**	0.56171104 c**
Gill	9.7857721	9.79081678 a**	9.78893903 c**	9,793337e ***	10.2857721 a**	9.789556 e****	9.813723 a****
Problem 1	2	2 a**	2 a**	2 a**	2.38780222 a**	2 a**	2 a**
Rosenbrock L1 versiyonu	0	0.00631878 b**	0.00128238 c**	0.00061666 b*	0.49641875 a**	0.00012007 a**	0.00015433 a**
Wood fonksiyonunun L1 versiyonu	0	0.00301524 a**	0.01805092 c**	0.00028644 c**	0.5 a**	0.00220735 c**	0.0090316 a**
EXP	0,00012237125	-0.0121952 c**	-5.7516929 c**	0.00012857 c**	0.13212056 c**	-0.0622914 c**	-0.0024076 c**
Kowalik-Osborne	0.008084368	0.00816606 c**	0.01216215 c**	0.00852787 e**	0.0475133 a**	0.00816741 e**	0.00815057 d**
OET5	0.0026359735	0.00270181 b**	0.00309279 a**	0.00340681 c**	0.5026359 a**	0.00290543 a**	0,00325996 b*
OET 6	0.0020160753	0.00316036 b**	0.01109648 a**	0.00535588 c**	0.181600722 c**	0.00336216 a**	0.00317971 a**
PBC1	0.022340496	0.19568146 c**	0.02734077 c**	0.11975801 c**	0.5223405 a**	0.08501973 c**	0.11826176 b**
EVD61	0.034904926	0.03515377 c**	0.03549639 a**	0.03651809 c**	0.53490493 a**	0.03606183 a**	0.03578041 a**

Tablo 6.3. Küçük boyutlu test problemlerini çözmek için gereken CPU

İsim	Sabit Adım Uzunluğu	Azalan Adım Uzunluğu (4.1)	f^* ve x^* ile Dinamik Adım Uzunluğu (4.2)	$f^{lev} > f^*$ ile Dinamik Adım Uzunluğu (4.3)	$f^{lev} < f^*$ ile Dinamik Adım Uzunluğu (4.4)	f_k^{lev} ile Dinamik Adım Uzunluğu (4.5)
Crescent	0,01948595	0,014942884	0,631707907	0,631707907	0,434367895	3,1726861
Mifflin 2	0,034101009	0,014023066	0,219409943	0,566859007	0,061650991	0,462354898
WF	0,547713041	0,533092976	0,725044966	0,663811922	0,622557878	20,59515786
SPIRAL	0,281521082	0,86281395	1,024422884	0,912091017	0,952101946	20,95985794
EVD52	1,022166967	1,020539999	1,217097044	1,148584127	1,108290911	20,95801091
PBC3	9,269674063	9,183539152	9,327889919	9,327889919	9,248991013	26,57418299
Bard	1,943536043	1,943570137	2,124976873	2,056656122	2,07536602	22,84766889
Polak 6	2,697446108	2,847462893	3,108037949	2,990446091	2,97774601	20,02642322
El-Attar	32,12747598	36,26708293	36,24204493	36,0158658	36,07827997	53,48014712
Gill	140,1797779	140,2673838	140,470124	139,792407	139,6418769	157,4582298
Problem 1	0,164664984	0,095170975	1,18600297	1,152608156	1,149125099	18,26833701
Rosenbrock L1 versiyonu	0,302460909	0,320847988	0,492063046	0,422068834	0,422070026	35,27688599
Wood fonksiyonunun L1 versiyonu	0,765971899	0,759523869	1,032075882	0,898361921	0,870677948	18,93293786
EXP	0,008785963	0,006563902	0,012379169	0,012379169	0,000733852	0,020792961
Kowalik- Osborne	2,661329985	2,677920103	2,881727934	2,881727934	2,805956841	20,92395306
OET5	5,282094955	5,264348984	5,482936144	5,343361139	5,365101099	22,59601593
OET 6	4,073904037	4,05181694	4,300542116	4,300542116	4,179040909	21,37431383
PBC1	18,76057696	18,81023693	18,94459295	18,857306	18,71514201	36,15938878
EVD61	36,879673	36,7343452	37,17999387	37,06612515	36,97888398	54,38752317

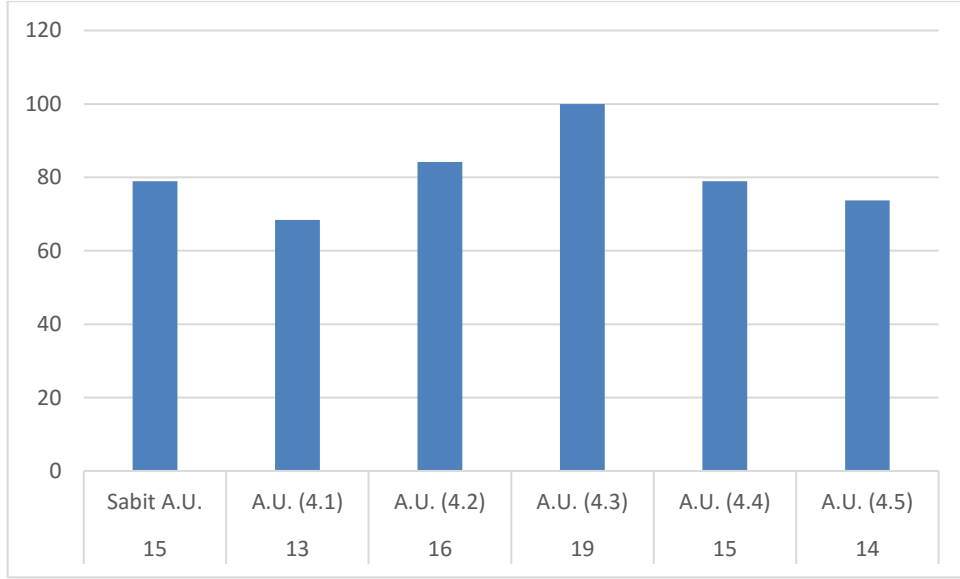
Tablo 6.4 Küçük boyutlu test problemlerinin optimal sonuçlara göre değerlendirilmesi $\left(\frac{f_{ZSM}-f_{opt}}{1+|f_{opt}|}\right)$

İsim	Sabit Adım Uzunluğu	Azalan Adım Uzunluğu (4.1)	f^* ve x^* ile Dinamik Adım Uzunluğu (4.2)	$f^{lev} > f^*$ ile Dinamik Adım Uzunluğu (4.3)	$f^{lev} < f^*$ ile Dinamik Adım Uzunluğu (4.4)	f_k^{lev} ile Dinamik Adım Uzunluğu (4.5)
Crescent	0	0	0	0	0	0
Mifflin 2	0	0	0	0	0	0
WF	0	0	0,00071218	0	0	0
SPIRAL	0	0	0	0	0	0
EVD52	0	0,00013249	0,00013205	0,00011767	0,00013158	0,00015737
PBC3	5,3349E-06	0,00209945	6,7361E-06	-0,1687307	6,5E-06	8,5912E-06
Bard	5,6016E-05	7,1119E-05	5,2886E-05	-0,0327675	4,4549E-05	3,6993E-05
Polak 6	0,00022222	0,00022222	0,00022222	0	0,00022222	0,00022222
El-Attar	1,9182E-05	5,6667E-05	3,9171E-06	0	3,6543E-07	0,00121677
Gill	0,00046772	0,00029362	0,00070138	0	0,00035082	0,00259146
Problem 1	0	0	0	-0,0320565	0	0
Rosenbrock L1 versiyonu	0,00631878	0,00128238	0,00061666	-0,0023875	0,00012007	0,00015433
Wood fonksiyonunun L1 versiyonu	0,00301524	0,01805092	0,00028644	0	0,00220735	0,0090316
EXP	-0,0123161	-5,7511115	6,198E-06	-0,2453145	-0,0624061	-0,0025297
Kowalik- Osborne	8,1037E-05	0,00404508	0,00043995	-0,3054014	8,2376E-05	6,5671E-05
OET5	6,5663E-05	0,00045562	0,00076881	-4,891E-08	0,00026875	0,00062235
OET 6	0,00114198	0,00906213	0,00333308	-0,2133235	0,00134338	0,00116129
PBC1	0,16955306	0,00489101	0,09528872	2,6275E-09	0,06130955	0,09382516
EVD61	0,00024045	0,00057152	0,00155876	2,606E-09	0,00111788	0,00084596



Şekil 6.1. Küçük boyutlu test problemleri için adım uzunluklarının optimal değere ulaşma yüzdeleri

$$\left(\frac{f_{ZSM} - f^*}{1 + |f^*|} < 5 \times 10^{-4} \right)$$

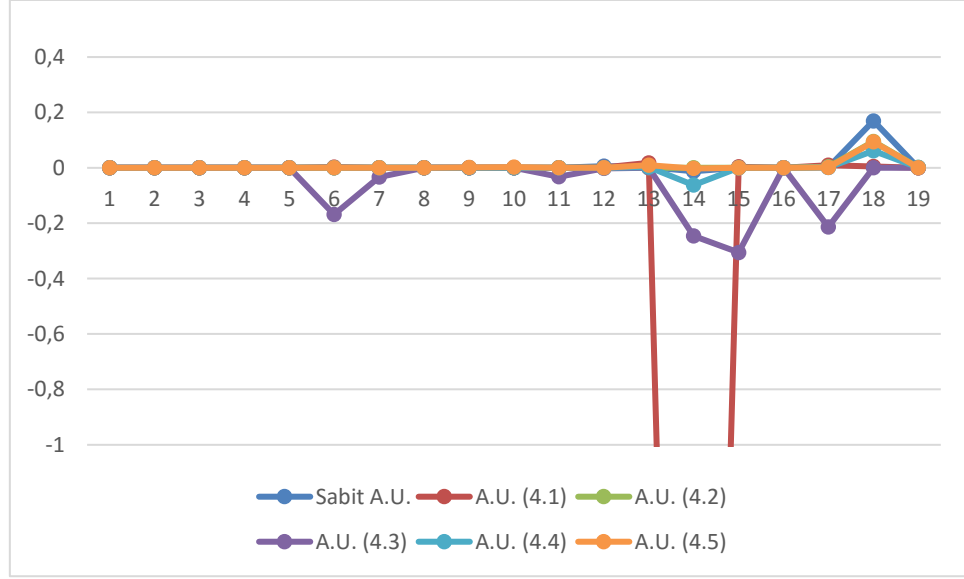


Şekil 6.2. Küçük boyutlu test problemleri için adım uzunluklarının optimal değere ulaşma yüzdeleri

$$\left(\frac{f_{ZSM} - f^*}{1 + |f^*|} < 10^{-3} \right)$$

Şekil 6.1 ve Şekil 6.2’de optimal değere ulaşma yüzdeleri verilmiştir. Küçük boyutlu test problemleri için $f^{lev} > f^*$ ile dinamik adım uzunluğu (4.3) %100 başarı göstermiştir. Bu demek oluyor ki optimal değerden daha büyük bir değer f^{lev} olarak seçildiğinde bu değere her seferinde ulaşılabilir. Sabit adım uzunluğu için her iki

değerlendirme kriterine göre %79, azalan adım uzunluğu (4.1) için %63 ve %68, f^* ve x^* ile dinamik adım uzunluğu (4.2) için %63 ve %84, $f_k^{lev} < f^*$ ile dinamik adım uzunluğu (4.4) için her iki değerlendirme kriterine göre %79 ve f_k^{lev} ile dinamik adım uzunluğu (4.5) için %63 ve %74'tür. Bunlara ek olarak Tablo 6.4'de bakıldığında bulunan tüm değerlerin optimal sonuçtan farkları Şekil 6.3'de de görüldüğü gibi 0,2'nin altındadır.



Şekil 6.3. Küçük boyutlu test problemleri için adım uzunluklarının optimal değerden farkları

Ayrıca EXP probleminde literatürde 0,00012237125 olarak verilen yaklaşık optimal değerinden daha iyi sonuçlar bulunmuştur. İzleyen tabloda bulunan x değerleri ayrıntılı olarak verilmiştir.

Tablo 6.5. EXP problemi için zayıf subgradyant algoritması ile bulunan sonuçlar

Adım uzunluğu	x değeri	f değeri
Sabit adım uzunluğu	(0,4419493; 0,058034182; -0,019803764; 0,019785165; -0,03978227152)	-0,0121952
Azalan adım uzunluğu	(-5; -0,210669565; 0,4713283899; 0,0347783024; -0,008888538198)	-5,7516929
$f_k^{lev} < f^*$ ile Dinamik Adım Uzunluğu (4.4)	(0,419541165; 0,0817195578; -0,033852981; 0,035499923; -0,03612751237)	-0,0622914
f_k^{lev} ile Dinamik Adım Uzunluğu (4.5)	(0,44561318; 0,054397455; -0,023483996; 0,0234806786; -0,02347538616)	-0,0024076

6.2. Büyük Boyutlu Test Problemleri

Differansiyellenemeyen ve dışbükey olmayan büyük boyutlu test problemleri 15 adet olup isimleri, değişken sayıları, amaç fonksiyonları, optimal amaç fonksiyonu değeri, optimal çözümü ve başlangıç noktaları Tablo 6.6’da verilmiştir. Kutu kısıtları küçük boyutlu test problemlerinde olduğu gibi her bir değişken için $[-5,5]$ aralığı seçilmiş, eğer optimal çözüm değeri bu aralığın dışında kalıyor ise aralık $[x^* - 5, x^* + 5]$ olacak şekilde değiştirilmiştir. Her bir problem 50 ve 200 değişkenli olarak ele alınmıştır.

Tablo 6.6 “50” değişkenli test problemlerinin sonuçları

İsim	n	f^*	Sabit Adım Uzunluğu	Azalan Adım Uzunluğu (4.1)	f^* ve x^* ile Dinamik Adım Uzunluğu (4.2)	$f^{lev} > f^*$ ile Dinamik Adım Uzunluğu (4.3)	$f^{lev} < f^*$ ile Dinamik Adım Uzunluğu (4.4)	f_k^{lev} ile Dinamik Adım Uzunluğu (4.5)
Aktif yüzlerin sayısı	50	0	0,00319693 a**	0,02869864 a**	0,00048501 b**	0,499999988 a**	0,004078518 a**	0,004235249 a**
Brown fonksiyonunun differansiyellenemeyen genelleştirilmesi 2	50	0	0,03272668 e***	0,003179896 a**	0,002742087 a**	0,5 a**	0,00971032 b**	0,01909278 a***
Chained mifflin 2	50	-34.795	-34,767706 b**	-34,780171 b**	NA	-34,295 b**	-34,767043 b**	-34,70324552 a**
Chained crescent I	50	0	0,00083241 c**	0,00048519 b**	0,011475414 e**	0,5 a**	0,00116549 b**	0,045976722 a**
Chained crescent II	50	0	0,0143396 a***	0,02202778 b**	0,0015307706 d***	0,5 a**	0 b***	0,004872756 a***
Test29'daki Problem 6	50	0	0,003469075 a**	0,01176838 b**	NA	0,5 b**	0,004783836 a**	0,004071199 a**
Test29'daki Problem 17	50	0	0 a**	0,00034937 a**	0 a**	0,00034937 a**	0,00034937 a**	0 a**
Test29'daki Problem 19	50	0	0,00040046 c**	0,074161531 a**	NA	0,50416224 b**	0,000933674 a**	0,002417618 a**
Test29'daki Problem 20	50	0	0,00543769 c**	0,652707923 a**	NA	0,504220913 a**	0,006593647 a**	0,0073205 d**
Test29'daki Problem 22	50	0	0,000681087 a**	0,000681087 a**	NA	0,000681087 a**	0,000681087 a**	0,000680983 a**
Test29'daki Problem 24	50	0	0,0155781 a***	0,06082041 c**	NA	0,5 a**	0,01266039 d***	0,012916485 d***
DC Maxl	50	0	0,733963734 a**	0 d***	0,0173642 b**	0,5 a**	0,013277907 a**	2,575630571 a**
DC Maxlq	50	0	0 b**	0 b**	1 a**	1 a**	1 a**	1 a**
Problem 6	50	0	0,002896596 a**	0,004477454 a**	0,00546972 e**	0,5 a**	0,001401189 a**	0,028024848 a**
Problem 7	50	0	0 a**	0,00034937 a**	0 a**	0,00034937 a**	0,00034937 a**	0 a**

NA: Bu problemlerin optimal çözümleri literatürde verilmediğinden dolayı f^* ve x^* ile dinamik adım uzunluğu kullanılarak çözülememiştir.

Tablo 6.7. “50” değişkenli test problemlerini çözmek için gereken CPU

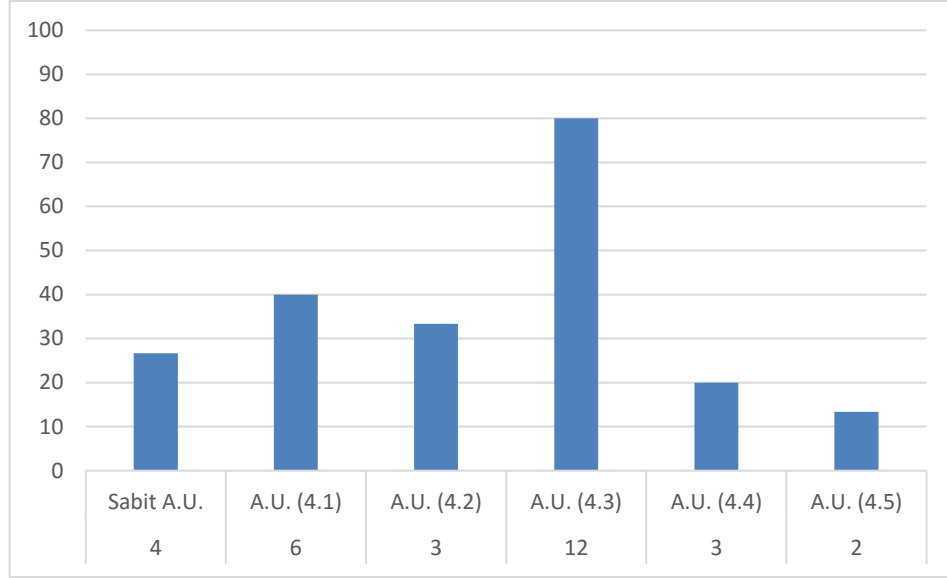
İsim	Sabit Adım Uzunluğu	Azalan Adım Uzunluğu (4.1)	f^* ve x^* ile Dinamik Adım Uzunluğu (4.2)	$f^{lev} > f^*$ ile Dinamik Adım Uzunluğu (4.3)	$f^{lev} < f^*$ ile Dinamik Adım Uzunluğu (4.4)	f_k^{lev} ile Dinamik Adım Uzunluğu (4.5)
Aktif yüzlerin sayısı	60,89470315	61,32440186	61,83431196	61,43170595	61,22593093	80,11220503
Brown fonksiyonunun differansiyellenemey en genelleştirilmesi 2	118,402369	119,1848428	120,2604761	119,8790069	119,8271601	137,5036418
Chained mifflin 2	140,248518	139,902606	NA	139,7290549	140,1551671	158,2280879
Chained crescent I	150,960669	151,9640241	152,2912049	152,103076	151,76353	169,998395
Chained crescent II	169,9192271	170,2918479	171,9059749	170,9188769	170,923506	189,8953011
Test29'daki Problem 6	57,08668494	55,80409384	NA	55,75443602	55,6193831	73,220474
Test29'daki Problem 17	358,947691	340,8179998	38,76913595	38,76913595	361,896925	9,850687981
Test29'daki Problem 19	75,80592895	75,24208403	NA	72,29903698	72,37821198	90,01108479
Test29'daki Problem 20	57,06797409	57,5266881	NA	57,39999795	57,69292903	75,31407619
Test29'daki Problem 22	120,7020991	122,7205071	NA	122,7205071	111,9200342	129,3395798
Test29'daki Problem 24	93,43427801	94,0051198	NA	93,92194915	93,91225505	113,7364788
DC Maxl	33,23734617	33,57928205	34,49485493	33,78634214	33,81907105	50,96383381
DC Maxlq	60,98545122	61,79449296	62,12458491	61,49921107	61,46381116	78,77859783
Problem 6	917,8916519	915,804148	916,0025389	918,0848951	915,8455608	935,6233439
Problem 7	358,947691	340,8179998	38,76913595	38,76913595	361,896925	9,850687981

NA: Bu problemlerin optimal çözümleri literatürde verilmediğinden dolayı f^* ve x^* ile dinamik adım uzunluğu kullanılarak çözülememiştir.

Tablo 6.8. “50” değişkenli test problemlerinin sonuçlarının optimal sonuçlara göre değerlendirilmesi $\left(\frac{f_{ZSM}-f_{opt}}{1+|f_{opt}|}\right)$

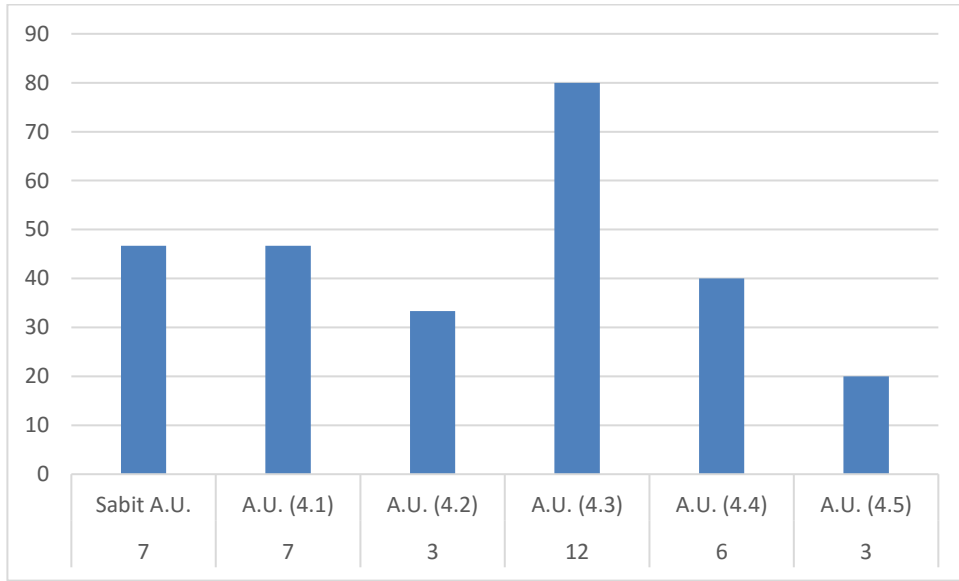
İsim	Sabit Adım Uzunluğu	Azalan Adım Uzunluğu (4.1)	f^* ve x^* ile Dinamik Adım Uzunluğu (4.2)	$f^{lev} > f^*$ ile Dinamik Adım Uzunluğu (4.3)	$f^{lev} < f^*$ ile Dinamik Adım Uzunluğu (4.4)	f_k^{lev} ile Dinamik Adım Uzunluğu (4.5)
Aktif yüzlerin sayısı	0,003197	0,028699	0,000485	-8E-09	0,004079	0,004235
Brown fonksiyonunun differansiyellenemey en genelleştirilmesi 2	0,032727	0,00318	0,002742	0	0,00971	0,019093
Chained mifflin 2	0,000763	0,000414	NA	0	0,000781	0,002563
Chained crescent I	0,000832	0,000485	0,011475	0	0,001165	0,045977
Chained crescent I	0,01434	0,022028	0,001531	0	0	0,004873
Chained crescent II	0,003469	0,011768	NA	0	0,004784	0,004071
Test29'daki Problem 6	0	0,000349	0	-0,3331	0,000349	0
Test29'daki Problem 17	0,0004	0,074162	NA	0,002775	0,000934	0,002418
Test29'daki Problem 19	0,005438	0,652708	NA	0,002814	0,006594	0,007321
Test29'daki Problem 20	0,000681	0,000681	NA	-0,33288	0,000681	0,000681
Test29'daki Problem 22	0,015578	0,06082	NA	0	0,01266	0,012916
Test29'daki Problem 24	0,733964	0	0,017364	0	0,013278	2,575631
DC Maxl	0	0	1	0,333333	1	1
DC Maxlq	0,002897	0,004477	0,00547	0	0,001401	0,028025
Problem 6	0	0,000349	0	-0,3331	0,000349	0

NA: Bu problemlerin optimal çözümleri literatürde verilmediğinden dolayı f^* ve x^* ile dinamik adım uzunluğu kullanılarak çözülememiştir.



Şekil 6.4 “50” değişkenli test problemleri için adım uzunluklarının optimal değere ulaşma yüzdeleri

$$\left(\frac{f_{ZSM} - f^*}{1 + |f^*|} < 5 \times 10^{-4} \right)$$

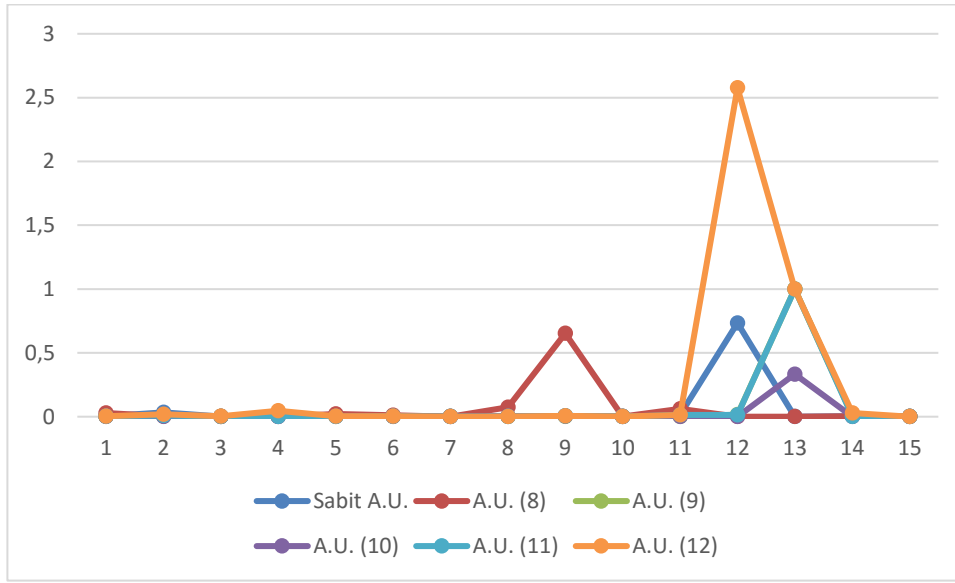


Şekil 6.5. “50” değişkenli test problemleri için adım uzunluklarının optimal değere ulaşma yüzdeleri

$$\left(\frac{f_{ZSM} - f^*}{1 + |f^*|} < 10^{-3} \right)$$

Şekil 6.4 ve Şekil 6.5’te optimal değere ulaşma yüzdeleri verilmiştir. “50” boyutlu test problemleri için her iki değerlendirme kriterine göre bakıldığında başarı yüzdeleri, sabit adım uzunluğu için %27 ve %47, azalan adım uzunluğu (4.1) için %40 ve %47, f^* ve x^* ile dinamik adım uzunluğu (4.2) için her iki değerlendirme kriterine göre %33, $f^{lev} > f^*$ ile dinamik adım uzunluğu (4.3) için her iki değerlendirme kriterine göre %80, $f^{lev} < f^*$ ile dinamik adım uzunluğu (4.4) için %20 ve %40 ve f_k^{lev} ile dinamik adım uzunluğu (4.5) için %13 ve %20’dir.

Bunlara ek olarak Tablo 6.7’den bakıldığında bulunan tüm değerlerin optimal sonuçtan farkları Şekil 6.6’da görülmektedir ve bu fark 13 (%87) problem için 0,7’nin altındayken bir problemde (DCMax1) 1’e ve bir problemde (Test29’daki Problem 24) 2,5’e ulaşmıştır. Bunun sebebinin zayıf subgradyantın tahmininden kaynaklanabilmektedir. Çünkü zayıf subgradyantın tahmini yönlü türeve dayanmaktadır ve yönlü türevin zayıf subgradyantların supremumuna eşit olması için 3 varsayım bulunmaktadır. Bu problemlerin bu varsayımlardan bir veya birkaçını sağlamadığı düşünülmektedir.



Şekil 6.6 “50” değişkenli test problemleri için adım uzunluklarının optimal değerden farkları

Tablo 6.9. “200” değişkenli test problemlerinin sonuçları

İsim	f^*	Sabit Adım Uzunluğu	Azalan Adım Uzunluğu (4.1)	f^* ve x^* ile Dinamik Adım Uzunluğu (4.2)	$f^{lev} > f^*$ ile Dinamik Adım Uzunluğu (4.3)	$f^{lev} < f^*$ ile Dinamik Adım Uzunluğu (4.4)	f_k^{lev} ile Dinamik Adım Uzunluğu (4.5)
Aktif yüzlerin sayısı	0	0.09010911 a*	0,97915948 d*	0,00264918 e**	0,5 a**	0,009383143 a**	0,01170317 a**
Brown fonksiyonunun differansiyellenemeyen genelleştirilmesi 2	0	0,1346936 e***	199 a***	0 a***	0,5 a**	0 a***	0,096967 a***
Chained mifflin 2	-140.86	-140,72646 e**	-140,79148 e**	NA	-139,979186 a**	-140.73536 e**	-139,8939 e**
Chained crescent I	0	0,002835493 a**	0,000968973 a**	0,02667114 e**	0,511846284 a**	0,00180416 e**	0,662850133 a**
Chained crescent II	0	0,05358298 e***	0 e***	0,00420896 e***	0,5 a**	0 e***	0,08312041 a***
Test29'daki Problem 6	0	0,0052669 e**	0,00838916 e***	NA	0,522376989 a**	0,006795878 a**	0,0093181 b**
Test29'daki Problem 17	0	0 a**	0 a**	0 a**	0 a**	0 a**	0 a**
Test29'daki Problem 19	0	0,00719051 e**	0,556159487 a**	NA	0,51424439 e**	0,00378467 b**	0,00852129 b**
Test29'daki Problem 20	0	0,48974114 e**	0,77293164 e**	NA	0,504458057 a**	0,027131183 d**	0,505147266 a**
Test29'daki Problem 22	0	0 a**	0 a**	NA	0 a**	0 a**	0 a**
Test29'daki Problem 24	0	0,01485382 a***	0,15269851 e**	NA	0,5 a**	0,01171833 e***	0,01205072 e***
DC Maxl	0	6,27698523 d*	0 a**	0,053693133 d**	0,48416392 e*	0,39277116 d*	9,54184555 d*
DC Maxlq	0	0 e**	0 a*	1 a**	1 a**	1 a**	1 a**
Problem 6	0	0,013907926 d**	0,132773358 a**	0,06056789 e**	0,5 e**	0,00645211 e**	0,061375149 a**
Problem 7	0	6.75767563 e**	0,53149851 e**	0,059584872 a**	0,5 a**	0,120413022 a**	1,170935921 a**

NA: Bu problemlerin optimal çözümleri literatürde verilmediğinden dolayı f^* ve x^* ile dinamik adım uzunluğu kullanılarak çözülememiştir.

Tablo 6.10. “200” değışkenli test problemlerini çözmek için gereken CPU

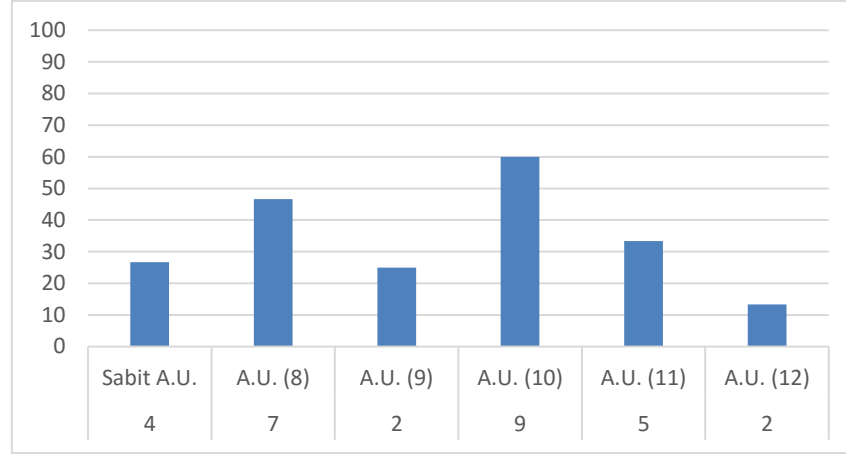
İsim	Sabit Adım Uzunluğu	Azalan Adım Uzunluğu (4.1)	f^* ve x^* ile Dinamik Adım Uzunluğu (4.2)	$f^{lev} > f^*$ ile Dinamik Adım Uzunluğu (4.3)	$f^{lev} < f^*$ ile Dinamik Adım Uzunluğu (4.4)	f_k^{lev} ile Dinamik Adım Uzunluğu (4.5)
Aktif yüzlerin sayısı	857,0965228	858,453763	862,784811	862,0762868	859,221395	879,2447381
Brown fonksiyonunun differansiyellenemeyen genelleştirilmesi 2	1821,448451	1936,892851	1828,617451	1825,738936	1826,521681	1849,44084
Chained mifflin 2	2156,047382	2156,041017	NA	2157,751231	2156,760214	2185,62943
Chained crescent I	2156,047382	2156,041017	2380,66207	2157,751231	2156,760214	2185,62943
Chained crescent II	2691,623174	2692,723938	2691,36668	2698,771745	2691,550648	2708,264858
Test29'daki Problem 6	784,9884422	787,202239	NA	787,8259768	787,801333	806,652524
Test29'daki Problem 17	5579,026889	5568,732214	1471,936929	1471,936929	5672,939605	251,4971209
Test29'daki Problem 19	1059,023876	1060,836025	NA	1078,361139	1084,608362	1076,036092
Test29'daki Problem 20	805,6136961	807,4834359	NA	810,8531868	818,7503982	833,4470501
Test29'daki Problem 22	1666,825887	1686,82367	NA	1686,82367	1669,217375	1694,619697
Test29'daki Problem 24	1609,523945	1621,387023	NA	1611,306507	1613,287288	1633,667805
DC Maxl	443,4699359	439,9732959	447,018518	444,5235229	442,949214	461,9291141
DC Maxlq	873,3499069	901,0368209	879,982172	877,4669721	876,510994	899,1966832
Problem 6	13683,90677	13565,5557	13571,807	13637,44187	13591,0403	13698,78747
Problem 7	15232,01185	15144,22455	15183,72359	15173,45608	15130,35306	15150,1776

NA: Bu problemlerin optimal çözümleri literatürde verilmediğinden dolayı f^* ve x^* ile dinamik adım uzunluğu kullanılarak çözülememiştir.

Tablo 6.11. “200” değişkenli test problemlerinin sonuçlarının optimal sonuçlara göre değerlendirilmesi $\left(\frac{f_{ZSM} - f_{opt}}{1 + |f_{opt}|} \right)$

İsim	Sabit Adım Uzunluğu	Azalan Adım Uzunluğu (4.1)	f^* ve x^* ile Dinamik Adım Uzunluğu (4.2)	$f^{lev} > f^*$ ile Dinamik Adım Uzunluğu (4.3)	$f^{lev} < f^*$ ile Dinamik Adım Uzunluğu (4.4)	f_k^{lev} ile Dinamik Adım Uzunluğu (4.5)
Aktif yüzlerin sayısı	0,090109	0,979159	0,002649	0	0,009383	0,011703
Brown fonksiyonunun differansiyellenemeyen enelleştirilmesi 2	0,134694	199	0	0	0	0,096967
Chained mifflin 2	0,000941	0,000483	NA	0,002694	0,000879	0,00681
Chained crescent I	0,002835	0,000969	0,026671	0,007898	0,001804	0,66285
Chained crescent II	0,053583	0	0,004209	0	0	0,08312
Test29'daki Problem 6	0,005267	0,008389	NA	0,014918	0,006796	0,009318
Test29'daki Problem 17	0	0	0	-0,33333	0	0
Test29'daki Problem 19	0,007191	0,556159	NA	0,009496	0,003785	0,008521
Test29'daki Problem 20	0,489741	0,772932	NA	0,002972	0,027131	0,505147
Test29'daki Problem 22	0	0	NA	-0,33333	0	0
Test29'daki Problem 24	0,014854	0,152699	NA	0	0,011718	0,012051
DC Maxl	6,276985	0	0,053693	-0,01056	0,392771	9,541846
DC Maxlq	0	0	1	0,333333	1	1
Problem 6	0,013908	0,132773	0,060568	0	0,006452	0,061375
Problem 7	6,757676	0,531499	0,059585	0	0,120413	1,170936

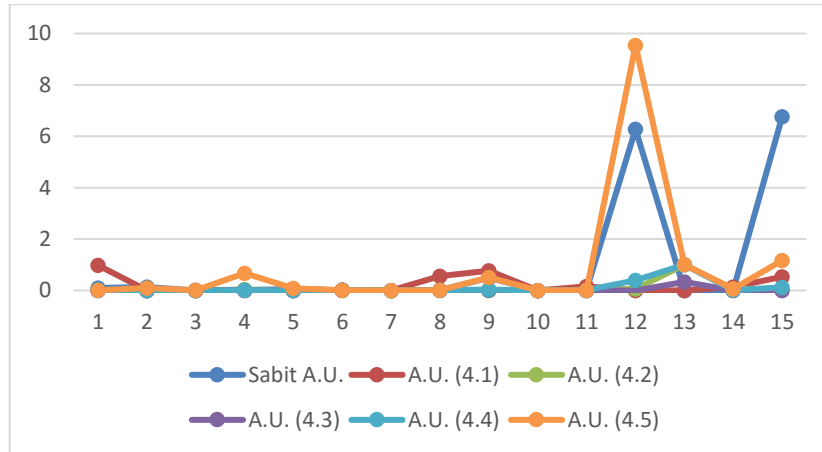
NA: Bu problemlerin optimal çözümleri literatürde verilmediğinden dolayı f^* ve x^* ile dinamik adım uzunluğu kullanılarak çözülememiştir.



Şekil 6.7. “200” değişkenli test problemleri için adım uzunluklarının optimal değere ulaşma yüzdeleri

$$\left(\frac{f_{ZSM} - f^*}{1 + |f^*|} < 10^{-3} \right)$$

Şekil 6.7’de optimal değerlere ulaşma yüzdeleri verilmiştir. Sabit adım uzunluğu için %27, azalan adım uzunluğu (4.1) için %48, f^* ve x^* ile dinamik adım uzunluğu (4.2) için %24, $f^{lev} > f^*$ ile dinamik adım uzunluğu (4.3) için %60, $f^{lev} < f^*$ ile dinamik adım uzunluğu (4.4) için %32 ve f_k^{lev} ile dinamik adım uzunluğu (4.5) için % 13 başarı yüzdesi elde edilmiştir. Tüm değerlerin optimal sonuçtan farkları Şekil 6.8’den de görüldüğü gibi genel olarak 1’in altındayken dört problemde (Brown fonksiyonunun differansiyellenemeyen genelleştirilmesi 2, azalan adım uzunluğu (4.1); DC Max1, sabit adım uzunluğu; DC Maxq, tüm dinamik adım uzunlukları; Problem 7 , sabit adım uzunluğu ve f_k^{lev} ile dinamik adım uzunluğu (4.5)) ciddi sapmalar olmuştur. Bu sapmalar başlangıç noktasından daha iyi bir noktaya gidilememesinden kaynaklanmaktadır.



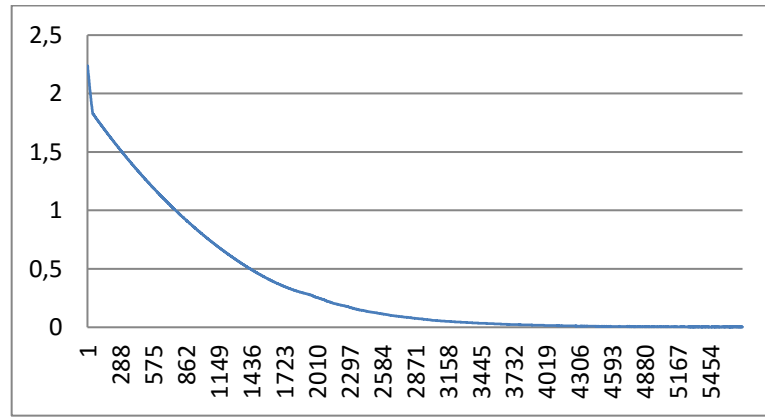
Şekil 6.8. “200” değişkenli test problemleri için adım uzunluklarının optimal değerden farklar

6.3. Tartışma

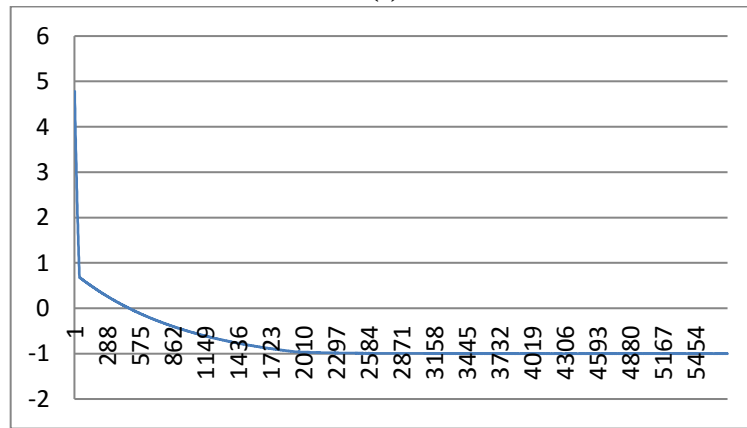
Zayıf subgradyant algoritması ile 6 farklı adım uzunluğu kullanılarak elde edilen sonuçlar bu bölümde iterasyonlar boyunca türetilen çözümlerin optimal çözümlere uzaklığının ve bu çözümlere karşılık gelen amaç fonksiyonu değerlerinin değişimi grafikler yardımı ile yorumlanmıştır. Küçük boyutlu, 50 değişkenli ve 200 değişkenli problemlerin her birinden optimal çözüme ulaşılan ve ulaşılamayanlardan bir adet problem incelenmiştir ve optimal çözüme ulaşılamayanların sebepleri tartışılmıştır.

6.3.1. Küçük boyutlu problemler

Küçük boyutlu problemlerden zayıf subgradyant algoritmasının her adım uzunluğunda optimal değere ulaşılmış olan problemlerden biri olan “Muffin 2” problemi için sabit adım uzunluğuna göre elde edilen grafikler Şekil 6.9’da verilmiştir.



(a)

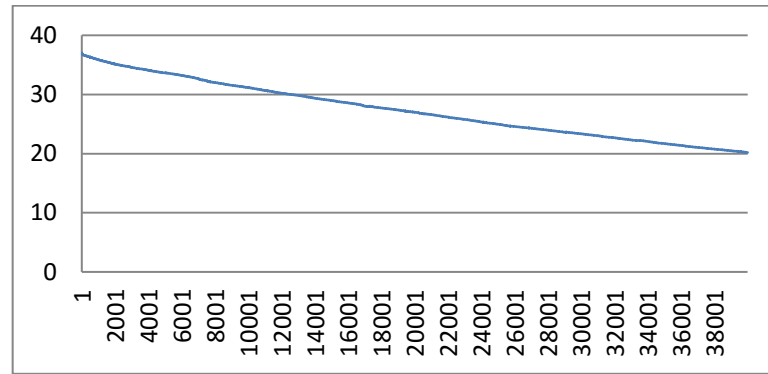


(b)

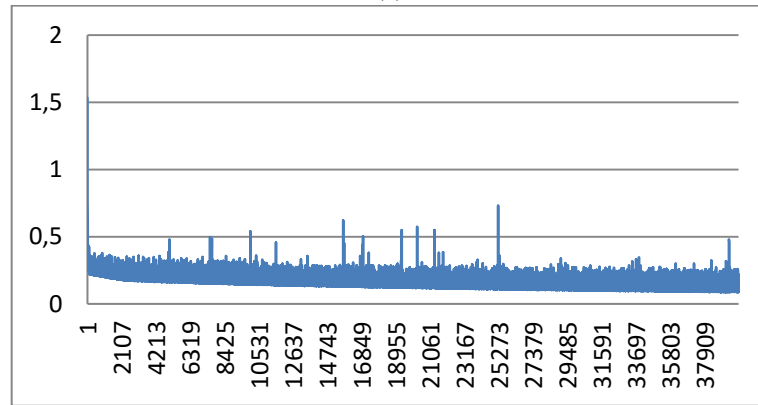
Şekil 6.9. Muffin 2 problemi – Sabit adım uzunluğu (a) $\|x_k - x^*\|$ (b) f_k iterasyonlar boyunca değişimi

Şekil 6.9'daki grafiklere bakıldığında hızlı bir şekilde optimal çözüm ile arasındaki uzaklık azalmış , 4000. iterasyondan sonra daha yavaş ilerlenmiştir. Benze şekilde amaç fonksiyonu değeri başlangıçta hızla azalırken, 2000. iterasyondan sonra azalmalar oldukça küçük olmuştur.

Küçük boyutlu problemlerden optimal çözüme diğer küçük boyutlu problemlere göre daha uzak sonuç bulan “PBC1” probleminin $f^{lev} < f^*$ ile dinamik adım uzunluğu (4.4) için oluşturulan grafikler Şekil 6.10'da görülmektedir. Her adımda optimal çözüme yaklaşıldığı açıkça görülmektedir. Buna bağlı olarak elde edilen amaç fonksiyonu değerinde çok hızlı bir şekilde 1,4 değerinden 0,2 değerine düşmüş, 0,2 değerinin civarında değişiklik göstermiştir. Bu problemde optimal çözümün elde edilememesinin nedeni iterasyon limitinin olduğu düşünülmüştür.



(a)

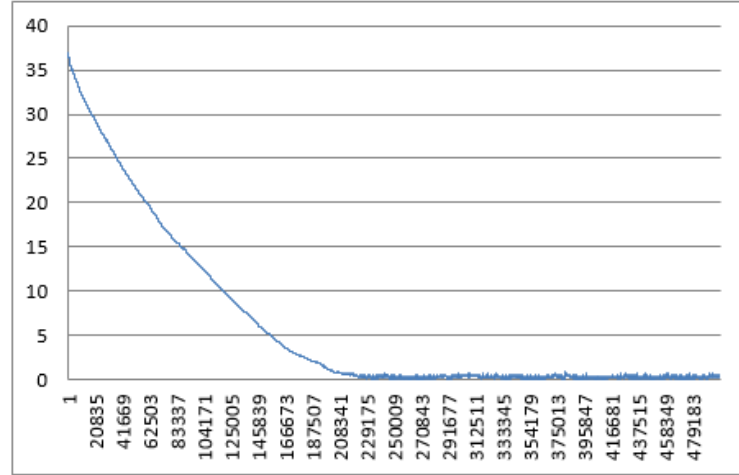


(b)

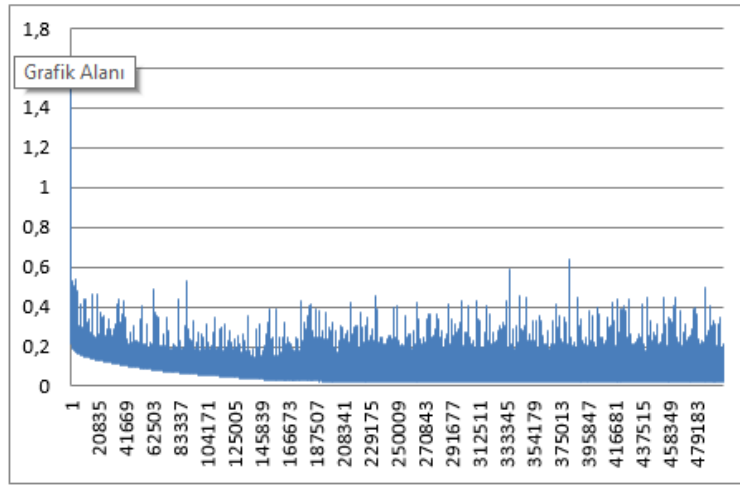
Şekil 6.10. PBC1 problemi – $f^{lev} < f^*$ ile dinamik adım uzunluğu (4.4) – 40000 iterasyon

(a) $\|x_k - x^*\|$ (b) f_k iterasyonlar boyunca değişimi

PBC1 problemi iterasyon sayısı 500000 verilerek çözülmüş ve Şekil 6.11’de verilen grafikler elde edilmiştir. Optimal çözüme olan uzaklık her iterasyonda giderek azalmış ve yaklaşık olarak 250000 iterasyon sonucunda optimal çözüme yakınsamıştır.



(a)



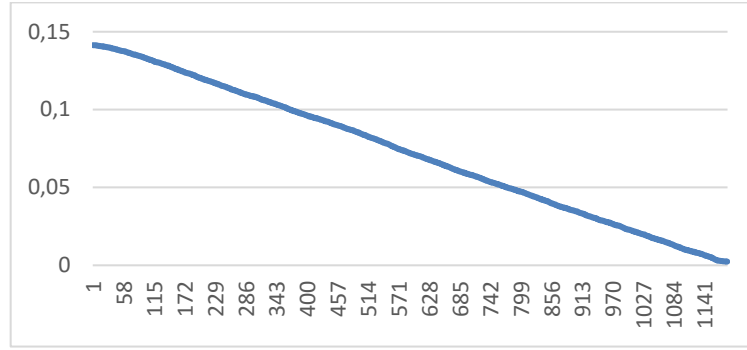
(b)

Şekil 6.11. PBC1 problemi – $f^{lev} < f^*$ ile dinamik adım uzunluğu (4.4) – 500000 iterasyon

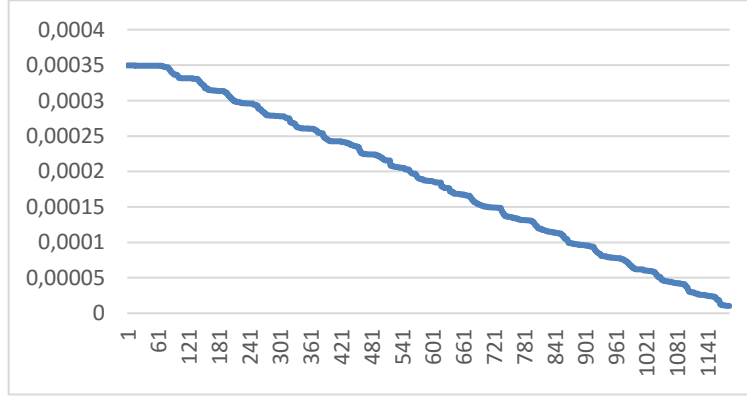
(a) $\|x_k - x^*\|$ (b) f_k iterasyonlar boyunca değişimi

6.3.2. 50 Değişkenli Problemler

“50” değişkenli problemlerden “Test 29’daki Problem17” incelenecek olursa f_k^{lev} ile adım uzunluğunda (4.5) optimal sonuç elde edilmiştir ve ilgili grafikler Şekil 6.12’de görüldüğü üzere yaklaşık 12000 iterasyon sonucunda optimal çözüme yakınsamıştır.



(a)

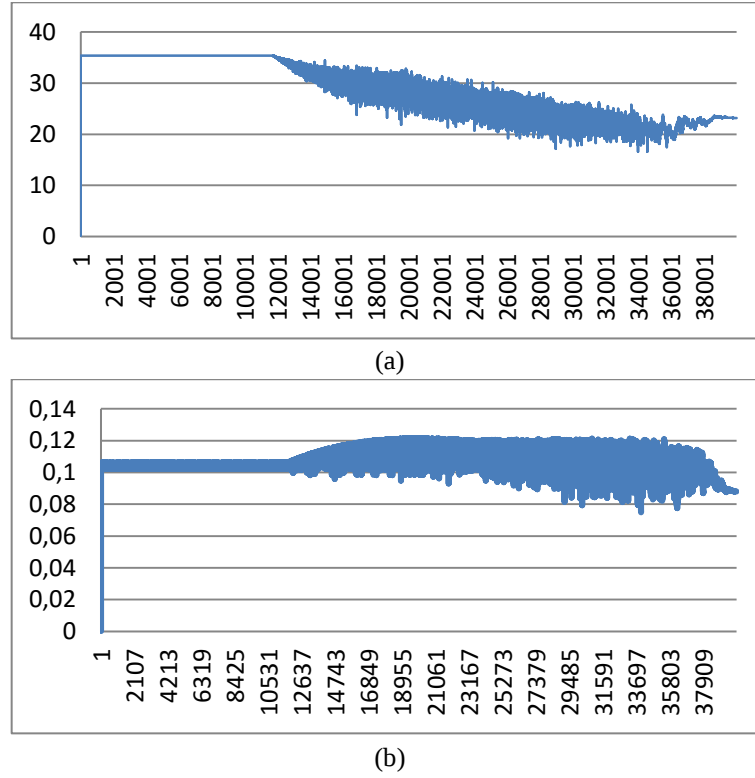


(b)

Şekil 6.12. Test 29’daki Problem17– f_k^{lev} ile dinamik adım uzunluğu (4.4)

(a) $\|x_k - x^*\|$ (b) f_k iterasyonlar boyunca değişimi

“50” değişkenli problemlerden “Test 29’daki Problem17” azalan adım uzunluğu (4.1) ile çözüldüğünde optimal çözüme ulaşamamıştır. İlgili grafikler Şekil 6.13’de görülmektedir. Optimal çözüme olan uzaklık yaklaşık 10000 iterasyon boyunca aynı kalmış daha sonra zigzaglar çizerek azalmıştır.

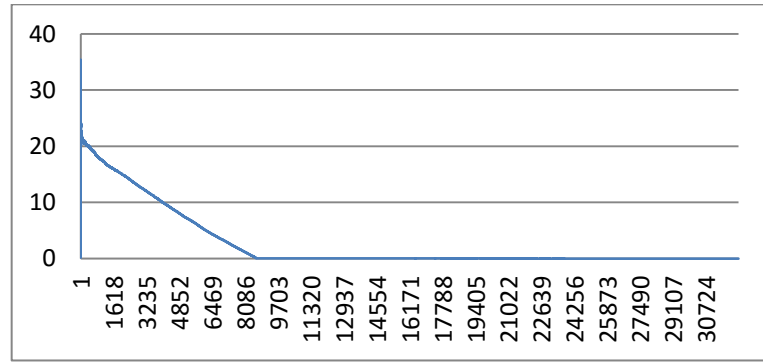


Şekil 6.13. Test 29’daki Problem17– Azalan adım uzunluğu (4.1)

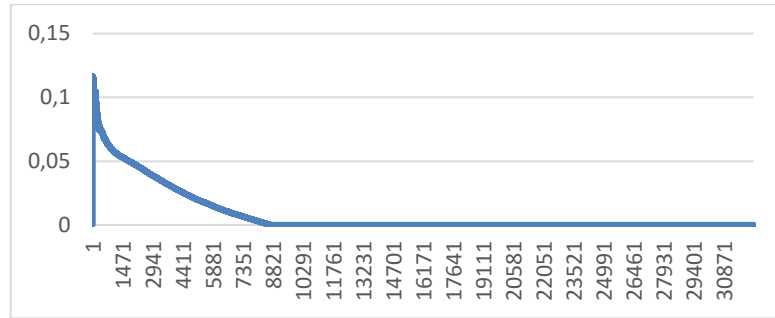
(a) $\|x_k - x^*\|$ (b) f_k iterasyonlar boyunca değişimi

“Test 29’daki Problem17” azalan adım uzunluğu (4.1) ile çözüldüğünde optimal çözüme ulaşamamasının sebebinin azalan adım uzunluğu $\frac{5}{2*k}$ olarak ele alınıp başlangıç iterasyonlarında adım uzunluğunun büyük olmasından kaynaklandığı düşünülmüştür. Adım uzunluğu parametre setinin içinde yer alan $1 - \left(\frac{1}{iter_{sayı}}\right) * k$ olacak şekilde seçilmiş, c_k parametresinin değişimi aynı şekilde $10/k$ olacak şekilde bırakılmıştır ve problemin optimal çözüme yakınsadığı görülmüştür. Şekil 6.14’de ilgili grafikler görülmektedir ve buna göre zayıf subgradyant algoritması yaklaşık 8100. iterasyonda optimal çözüme yakınsamıştır.

Belirlenen bu değerler ile c_k parametresi daha büyük bir değerden başlayıp iterasyon sayısına bağlı olarak hızla azalmış ve α_k parametresi küçük bir değerden başlayıp iterasyon sayısına göre daha yavaş azalmıştır. Bu iki değerın bereber kullanılması belirlenen parameter setinde yer almamıştır. Ancak bu problem göstermiştir ki $\alpha_k = 1 - \left(\frac{1}{iter_{sayı}}\right) * k$ ve $c_k = 10/k$ azalan adım uzunluğu için ele alınması gereken parametre setlerinden biri olmalıdır.



(a)



(b)

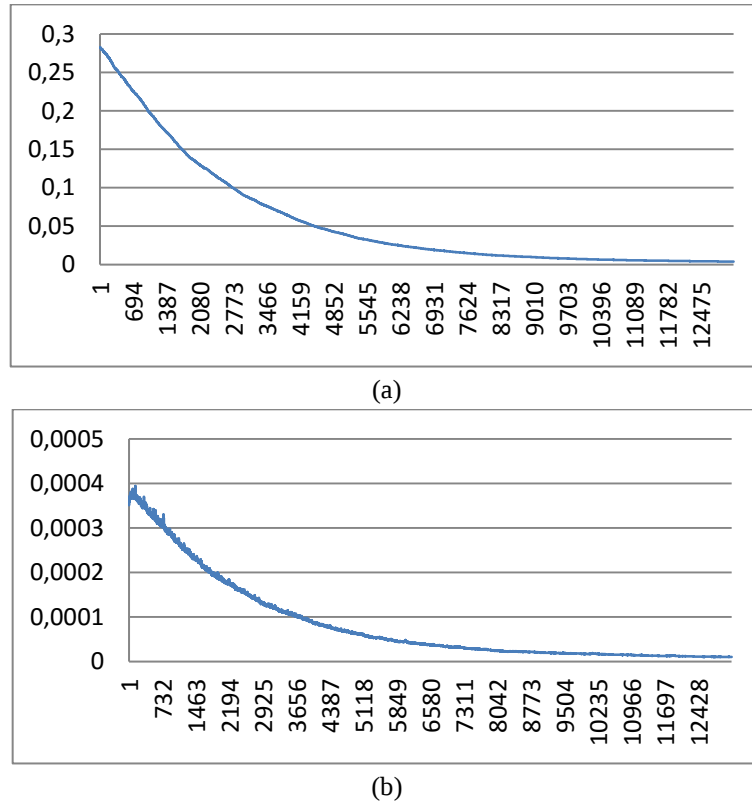
Şekil 6.14. Test 29'daki Problemler 7– Azalan adım uzunluğunun (4.1)

$$\alpha_k = 1 - \left(\frac{1}{iter_{sayı}}\right) * k \text{ ve } c_k = 10/k \text{ parameterleri ile}$$

(a) $\|x_k - x^*\|$ (b) f_k iterasyonlar boyunca değişimi

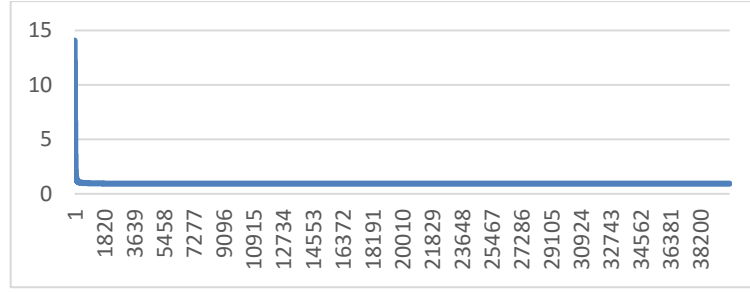
6.3.3. 200 Değişkenli Problemler

“Test 29’daki Problem17” zayıf subgradyant algortimasının her adım uzunluğu ile optimal çözüme yakınsamıştır. Adım uzunluklarından f^* ve x^* ile dinamik adım uzunluğu (4.2) ile ilgili grafikler Şekil 6.15’de görülmektedir. Toplamda 40000 iterasyon limiti verilmiş olmasına rağmen yaklaşık 13000 iterasyon sonucunda optimal çözüme yakınsamıştır.

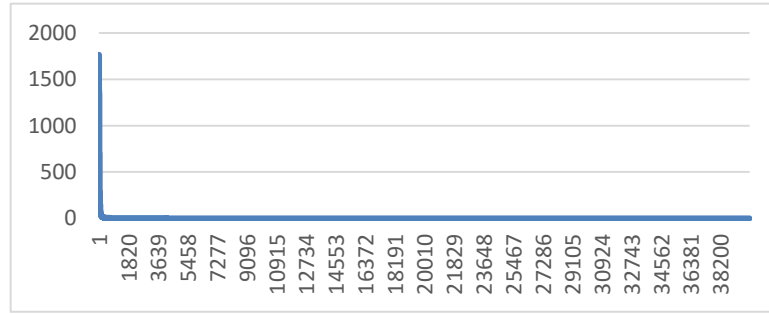


Şekil 6.15. Test 29’daki Problem17– f^* ve x^* ile dinamik adım uzunluğu (4.2)
(a) $\|x_k - x^*\|$ (b) f_k iterasyonlar boyunca değişimi

“Problem 7” ele alındığında adım uzunlukları ile optimal çözüme ulaşamamıştır. $f^{lev} < f^*$ ile adım uzunluğu (4.4) ile ilgili grafikler Şekil 6.16’da görülmektedir. 1600 iterasyon boyunca optimal çözüme olan uzaklık azalmış, daha sonra optimal çözüme olan uzaklık aynı kalmıştır. Bu durum Önerme 12’de ispatlanan $\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2$ ifadenin görsel bir sonucudur.



(a)



(b)

Şekil 6.16. Problem 7– $f^{lev} < f^*$ ile dinamik adım uzunluğu (4.4)

(a) $\|x_k - x^*\|$ (b) f_k iterasyonlar boyunca değişimi

7. SONUÇLAR

Differensallenemeyen optimizasyon problemleri için tanımlanmış olan subgradyant, gradyantin genelleşmiş halidir. Subgradyantlar, fonksiyonun destek hiperdüzlemleridir ve bu sebeple dışbükey fonksiyonlar için geçerlidir. Uygulaması kolay olan subgradyant metotları dışbükey fonksiyonların minimal noktalarının bulunması için yoğun bir şekilde kullanılmıştır.

Subgradyantin genelleşmiş hali olan zayıf subgradyantlar, fonksiyonun destek konileridir ve bu sebeple dışbükey olmayan fonksiyonlar için de geçerlidir. Bu tezin amacı zayıf subgradyantları kullanarak dışbükey olmayan fonksiyonların minimal noktalarını bulan bir algoritmanın geliştirilmesidir. Ancak pratikte zayıf subgradyantların hesaplanması kolay değildir. Bu sebeple öncelikle zayıf subgradyantların tahmini üzerinde çalışılmıştır. Zayıf subgradyant algoritması belirlendikten sonra adım uzunlukları üzerinde çalışılmış ve 6 farklı adım uzunluğu için zayıf subgradyant algoritmasının yakınsaklık özellikleri gösterilmiştir.

Öncelikle çok fazla yerel optimal çözüme sahip olan analitik fonksiyon (5.1) zayıf subgradyant algoritması ile 6 adım uzunluğunun her biri ile çözülmüş ve zayıf subgradyant algoritmasının yerel optimal çözümlerden kaçabildiği görülmüştür.

Zayıf subgradyantların tahmininde $c > 0$ olacak şekilde belirlenmekteydi. Bu sebeple analitik fonksiyon çözülürken her bir adım uzunluğunda farklı c değerleri belirlenmiştir. Ele alınan problem dışbükey olmayan ancak differansiyellenebilir olduğu durumda her bir iterasyonda c_k değeri $c_k = \arg \min_{0 \leq c_k < c_k^{üst}} f(x_{k+1})$ olacak şekilde daha belirlenebilme durumu tartışılmış ve analitik fonksiyon (5.1) çözülmüştür. Bu yaklaşım ile optimal çözümlere yakınsarken doğal olarak çözüm süresinde artma olmuştur. En büyük avantajı ise c_k değeri için baştan herhangi bir değer belirleme zorunluluğunu ortadan kaldırması olmuştur.

Analitik fonksiyon (5.1) üzerinde f_k^{lev} ile dinamik adım uzunluğuyla çözüm iteratif olarak incelenmiş ve problemin dışbükey olmayan yapısından dolayı doğal olarak iterasyonlarda bulunan $f(x_k)$ değerleri bir önceki iterasyona göre azalma veya artma göstermiştir. Benzer şekilde her bir iterasyonda f_k^{lev} değeri bir önceki iterasyonda bulunan $f(x_k)$ değerine göre aynı kalmış, artmış veya azalmıştır.

Gerçek hayatta ele alınan problemin optimal çözümü bilinmeyeceğinden dolayı verilen f^{lev} değerinin f^* değerinin üstünde veya altında olup olmadığı da

bilinemeyecektir. Bu sebeple $f^{lev} > f^*$ ile dinamik adım uzunluğu (4.3) ve $f^{lev} < f^*$ ile dinamik adım uzunluğu (4.4) berber nasıl kullanılabileceği üzerinde tartışılmış, analitik problem (5.1) üzerinde optimal sonuca ulaşılabildiği gösterilmiştir.

Literatürden alınan küçük boyutlu ($n < 10$) , “50” ve “100” değişkenli problemler zayıf subgradyant algoritması ile her adım uzunluğu kullanılarak çözülmüştür. Optimal sonuçtan farka bakılacak olursa küçük boyutlu test problemlerinde farklar 0,2’nin, “50” değişkenli problemlerde 2,5’in altında ve “200” değişkenli problemlerde ise genel olarak 1’in altındadır. “200” değişkenli problemlerde 4 problemde ciddi sapmalar olmuştur. Zayıf subgradyant algoritmasının her bir adım uzunluğuna göre problemlerdeki başarı yüzdeleri ise Tablo 7.1’de verilmiştir. $\frac{f_{ZSA}-f^*}{1+|f^*|}$ değeri 5×10^{-5} ve 10^{-3} değerinden küçük olduğu zaman optimal değere ulaşılmış, 10^{-2} ’den küçük olduğu durumda ise iyi sonuçların elde edildiği kabul edilmektedir.

Tablo 7.1. Zayıf subgradyant algoritmasının her bir adım uzunluğuna göre problemlerdeki başarı yüzdeleri

Adım Uzunluğu	Küçük Boyutlu Problemler			“50” Değişkenli Problemler			“200” Değişkenli Problemler		
	$\frac{f_{ZSA}-f^*}{1+ f^* }$	$\frac{f_{ZSA}-f^*}{1+ f^* }$	$\frac{f_{ZSA}-f^*}{1+ f^* }$	$\frac{f_{ZSA}-f^*}{1+ f^* }$	$\frac{f_{ZSA}-f^*}{1+ f^* }$	$\frac{f_{ZSA}-f^*}{1+ f^* }$	$\frac{f_{ZSA}-f^*}{1+ f^* }$	$\frac{f_{ZSA}-f^*}{1+ f^* }$	$\frac{f_{ZSA}-f^*}{1+ f^* }$
	< 5×10^{-5}	< 10^{-3}	< 10^{-2}	< 5 $\times 10^{-5}$	< 10^{-3}	< 10^{-2}	< 5 $\times 10^{-5}$	< 10^{-3}	< 10^{-2}
Sabit A.U.	%79	%79	%95	%27	%47	%74	%20	%27	%47
(4.1)	%63	%69	%95	%40	%47	%60	%40	%47	%54
(4.2)	%63	%84	%95	%33	%33	%75	%25	%25	%50
(4.3)	%100	%100	%100	%80	%80	%94	%60	%60	%87
(4.4)	%79	%79	%95	%20	%40	%80	%27	%34	%66
(4.5)	%63	%74	%95	%14	%20	%60	%14	%14	%34

5×10^{-5} ‘e göre küçük boyutlu problemlerde %100 ((4.3) adım uzunluğu) ve %84 ((4.4) adım uzunluğu); “50 değişkenli problemlerde %80 ((4.3) adım uzunluğu) ve %47 ((4.1) adım uzunluğu), “200” değişkenli problemlerde %60 ((4.3) adım uzunluğu) ve %40 ((4.1) adım uzunluğu) optimal değerler elde edilmiştir. 10^{-3} göre küçük boyutlu problemlerde %100 ((4.3) adım uzunluğu) ve %79 ((4.2) adım uzunluğu); “50 değişkenli

problemlerde %80 ((4.3) adım uzunluğu) ve %40 (Sabit ve (4.1) adım uzunlukları), “200” değişkenli problemlerde %60 ((4.3) adım uzunluğu) ve %47 ((4.1) adım uzunluğu) optimal değerler elde edilmiştir. 10^{-2} göre küçük boyutlu problemlerde %100 ((4.3) adım uzunluğu) ve %95(tüm adım uzunlukları); “50 değişkenli problemlerde %94 ((4.3) adım uzunluğu) ve %80 ((4.4) adım uzunluğu), “200” değişkenli problemlerde %87 ((4.3) adım uzunluğu) ve %66 ((4.4) adım uzunluğu) iyi sonuçlar elde edilmiştir.

Bundan sonraki çalışmalarda planlanan zayıf subgradyantın Kasımbeyli (2009) tarafından tanımlanmış olan radyal epitürev kavramı kullanılarak tahmin etmek ve yeni adım uzunlukları belirlenip bu adım uzunluklarının yakınsaklık özelliklerinin araştırmaktır.

KAYNAKLAR

- Ali, M., & Zhu, W. (2013). A penalty function-based differential evolution algorithm for constrained global optimization. *Computational Optimization and Applications*, 54, 707-739.
- Allen, E., Helgason, R., Kennington, J., & Shetty, B. (1987). A generalization of Polyak's convergence result for subgradient optimization. *Mathematical Programming*, 37, 309-317.
- Audet, C., Bechard, V., & Digabel, S. (2008). Nonsmooth optimization through Mesh Adaptive Direct Search and Variable Neighborhood Search. *Journal of Global Optimization*, 41, 299-318.
- Auslender, A., & Teboulle, M. (2004). Interior gradient and epsilon-subgradient descent methods for constrained convex minimization. *Mathematics of Operations Research*, 29, 1-26.
- Azimov, A., & Gasimov, R. (1999). On weak conjugacy, weak subdifferentials and duality with zero gap in nonconvex optimization. *International Journal of Applied Mathematics*, 1, 171-192.
- Azimov, A., & Gasimov, R. (2002). Stability and duality of nonconvex problems via augmented Lagrangian. *Cybernetics and Systems Analysis*, 3, 403-417.
- Bagirov, A. (1992). A method of approximating a subdifferential. *Zh. Vichisl. Mat. Mat. Fiz. İngilizce Computational Mathematics and Mathematical Physics*, 32(4), 652-658;561-566.
- Bagirov, A. (1994). Continuous approximation to a subdifferential of a function of a maximum. *Kibernetika i sistemnyi analiz İngilizce Cybernet and System Analysis*, 4, 180-184;626-630.
- Bagirov, A. (1998). A method for minimizing convex functions based on continuous approximations to the subdifferential. *Optimization Methods and Software*, 9, 1-17.
- Bagirov, A. (2002). A method for minimization of quasidifferentiable functions. *Optimization Methods and Software*, 17(1), 31-60.
- Bagirov, A. (2003). Continuous subdifferential approximations and their applications. *Journal of Mathematical Sciences*, 115(5), 2567-2609.

- Bagirov, A., & Ugon, J. (2011). Codifferential method for minimizing nonsmooth dc functions. *Journal of Global Optimization*, 50(1), 3-22.
- Bagirov, A., Karasözen, B., & Sezer, M. (2008). Discrete gradient method: derivative-free method for nonsmooth optimization. *Journal of Optimization Theory and Applications*, 137, 317-334.
- Bagirov, A., Karmista, N., & Mäkelä, M. (2014). *Introduction to Nonsmooth Optimization*. Springer.
- Bazaraa, M., & Sherali, H. (1981). On the choice of the step size in subgradient optimization. *European Journal of Operational Research*, 7, 380-388.
- Bazaraa, M., Sherali, H., & Shetty, C. (2006). *Nonlinear programming Theory and Algorithms* (4th ed.). New Jersey: Wiley.
- Bertsekas, D. P. (1999). *Nonlinear Programming*. Belmont, Massachusetts: Athena Scientific.
- Bertsekas, D., Nedic, A., & Ozdaglar, A. (2001). *Convexity, Duality and Lagrange Multipliers*. MIT.
- Borwein, J. (1982). A note on the existence of the subgradients. *Mathematical Programming*, 24, 225-228.
- Borwein, J., & Lewis, A. (2000). *Convex Analysis and Nonlinear Optimization: Theory and Examples*. New York: CMS Books in Mathematics, Springer.
- Brännlund, U. (1993). On relaxation methods for nonsmooth convex optimization. Stockholm, Sweden: Royal Institute of Technology.
- Burachik, R., Gasimov, R., Ismayilova, N., & Kaya, C. (2006). On a modified subgradient algorithm for dual problems via sharp augmented lagrangian. *Journal of Global Optimization*, 55-78.
- Clarke, F. (1983). *Optimization and nonsmooth analysis*. New York: Wiley-Interscience.
- Correa, R., & Lemaréchal, C. (1993). Convergence of some algorithms for convex minimization. *Mathematical Programming*, 62, 261-275.
- Demyanov, V. (1968). Algorithm for some minimax problem. *Computer and System Sciences*, 2, 342-380.
- Ermolyev, Y. (1966). Methods for solving nonlinear extremal problems. *Kibernetika*, 4, 1-17.

- Filik, U., & Kurban, M. (2010). Feasible Modified Subgradient Method for Solving the Thermal Unit Commitment Problem as a New Approach. *Mathematical Problems in Engineering*, 1-11.
- Gasimov, R. (2002). Augmented lagrangian duality and nondifferentiable optimization methods in nonconvex programming. *Journal of Global Optimization*, 24, 187-203.
- Gasimov, R., & Rubinov, A. (2004). On augmented Lagrangians for optimization problems with a single constraints. *Journal of Global Optimization*, 28, 153-173.
- Goffin, J. (1977). On convergent rates of subgradient optimization methods. *Mathematical Programming*, 13, 329-347.
- Goffin, J., & Kiwiel, K. (1999). Convergence of a simple subgradient level method. *Mathematical Programming*, 85, 207-211.
- Grothey, A. (2001). *Decomposition methods for nonlinear nonconvex optimization problems*. University of Edinburgh: PhD Tezi.
- Haarala, M., Miettinen, K., & Mäkelä, M. (2004). New limited memory bundle method for large-scale nonsmooth optimization. *Optimization Methods and Software*, 19(6), 673-692.
- Hu, Y., Yang, X., & Sim, C.-K. (2015). Inexact subgradient methods for quasi-convex optimization problems. *European Journal of Operational Research*, 240, 315-327.
- Kasimbeyli, R., & Inceoglu, G. (2010). The properties of the weak subdifferentials. *G.U. Journal of Science*, 23(1), 49-52.
- Kasimbeyli, R., & Mammadov, M. (2009). On weak subdifferentials, directional derivatives and radial epiderivatives for nonconvex functions. *SIAM Journal of Optimization*, 20(2), 841-855.
- Kasimbeyli, R., & Mammadov, M. (2011). Optimality conditions in nonconvex optimization via weak subdifferentials. *Nonlinear Analysis*, 74, 2534-2547.
- Kim, S., & Ahn, H. (1991). Convergence of a generalized subgradient method for nondifferentiable convex optimization. *Mathematical Programming*, 50, 75-80.
- Kim, S., & Um, B. (1993). An improved subgradient method for constrained nondifferentiable optimization. *Operations Research Letters*, 14, 61-64.
- Kim, S., Ahn, H., & Cho, S.-C. (1991). Variable target value subgradient method. *Mathematical Programming*, 49, 359-369.

- Kiwiel, K. (1983). An aggregate subgradient method for nonsmooth convex minimization. *Mathematical Programming*, 27, 320-341.
- Kiwiel, K. (1985). Methods of descent for nondifferentiable optimization. *Lecture Notes in Mathematics* (Cilt 1133). içinde Berlin: Springer.
- Kiwiel, K. (1996a). The efficiency of subgradient projection methods for convex optimization, Part I: General level methods. *SIAM Journal on Control and Optimization*, 34(2), 660-676.
- Kiwiel, K. (1996b). The efficiency of subgradient projection methods for convex optimization, Part II: Implementations and extensions. *SIAM Journal on Control and Optimization*, 34(2), 677-697.
- Kiwiel, K. (2001). Convergence and efficiency of subgradient methods for quasiconvex minimization. *Mathematical Programming Series A*, 90, 1-25.
- Kiwiel, K. (2004). Convergence of approximate and incremental methods for convex optimization. *SIAM Journal of Optimization*, 14(3), 807-840.
- Kiwiel, K., Larsson, T., & Lindberg, P. (1998). *The efficiency of ballstep subgradient level methods for convex optimization*. Linköpings University, Department of Mathematics, İsveç.
- Konnov, I. (2003). On convergence properties of a subgradient method. *Optimisation Methods and Soft*, 18(1), 53-62.
- Kulikov, A., & Fazylov, V. (1990). Convex optimization with prescribed accuracy. *USSR Computatioanl Mathematics and Mathematical Physics*, 30(3), 16-22.
- Li, G., Song, S., & Wu, C. (2006). Subgradient-based feedback neural networks for non-differentiable convex optimization problems. *Science in Chine Series F-Information Sciences*, 49, 421-435.
- Lukšan, L., & Vlček, J. (2000). *Test problems for nonsmooth unconstrained and linearly constrained optimization*. Institute of Computer Science. Prague: Academy of Sciences of the Czech Republic.
- Lukšan, L., Tcma, M., Šiška, M., Vlček, J., & Ramešová, N. (2002). *UFO 2002. Interactive system for universal functional optimization*. Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague.
- Mäkelä, M., & Neittaanmäki, P. (1992). *Nonsmooth optimization: Analysis and algorithms with applications to optimal control*. Singapore: Word Scientific Publishing Co.

- Nedic, A. (2002). *Subgradient methods for Convex Minimizaiton*. Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science.
- Nedic, A., & Bertsekas, D. (2010). The effect of deterministic noise in subgradient methods. *Mathematical Programming Series: A*, 125, 75-99. doi:10.1007/s10107-008-0262-5
- Nedic, A., Olshevsky, A., Ozdaglar, A., & Tsitsiklis, J. (2008). Distributed subgradient methods and quantization effects. *47th IEEE Conference on Decision and Control*, (s. 4177-4184). Cancun, Mexico.
- Neto, J., Da Silva, G., Ferreira, O., & Lopes, J. (2013). A subgradient method for multiobjective optimization. *Computational Optimization and Applications*, 54, 461-472.
- Ouorou, A. (2007). A proximal subgradient projection algorithm for linearly constrained strictly convex problems. *Optimization Method & Software*, 22, 617-636.
- Polak, E., Mayne, D., & Wardi, Y. (1991). Superlinearly convergent algorithm for min-max problems. *Journal of Optimization Theory and Applications*, 69, 407-439.
- Polyak, B. (1969a). Minimization of nonsmooth functions. *Vhchislitel'noy Matematiki i Matematicheskoy Fiziki*, 9, 14-29.
- Polyak, B. (1969b). The conjugate gradient method in extremal problems. *Vhchislitel'noy Matematiki i Matematicheskoy Fiziki*, 9, 94-112.
- Polyak, B. (1970). Iterative methods using lagrange multipliers for solving extremal problems with constraints of equation type. *Vhchislitel'noy Matematiki i Matematicheskoy Fiziki*, 10, 1098-1106.
- Rockafellar, R. (1970). *Convex analysis*. Princeton: Princeton University Press.
- Sherali, H., & Ulular, O. (1989). A primal-dual conjugate subgradient algorithm for specially structured linear and convex programming problems. *Applied Mathematics and Optimization*, 20, 193-221.
- Shor, N. (1985). *Minimization Methods for Non-Differentiable Functions*. (K. Kiwiel, & A. Ruszczyński, Çev.) Heidelberg: Springer-Verlag Berlin.
- Solodov, M., & Zavriev, S. (1998). Error stability properties of generalized gradient-type algorithms. *Journal of Optimization Theory and Applications*, 98(3), 663-680.

EKLER

EK-1 Zayıf Subgradyant Algoritmasının Python Kodları

- Zayıf subgradyantın tahmini

```
def appr_RE(f, x, lambda_dd, alpha_dd):
    v= [0]
    x_new = x[:]
    f_dd=[fk[index-1]]
    e=[0]
    for k in range(1, dim):
        e=random.uniform(0,1)
        if e<0.5:
            e=-1
        else:
            e=1
        x_new[k] = x[k] + e *lambda_dd* math.pow(alpha_dd,k)
        f_dd.append(f(x_new))
        v.append((((f_dd[k]-f_dd[k-1]) / (e*lambda_dd*math.pow(alpha_dd,k)))+(
c[index]/e))
    return v
```

- Sabit adım uzunluğuyla zayıf subgradyant algoritması

```
def ws_constant(xalt, xüst, x_0, f, iteration, alpha_k):
    start = time.time()
    x_k = x_0[:]
    global dim
    dim = len(x_0)
    global fbest, xbest, c, elapsed, fk
    c = [999999]
    fbest = f(x_k)
    xbest = x_k[:]
    fk = [f(x_k)]
    for i in range(1, iteration):
        global index
        index = i
        global cüst
        cüst=0
        c.append(5-(5/iteration)*i)
        v_k=appr_RE(f, x_k, lamda,aalpha)
        for j in range(1,dim):
            x_k[j]= x_k[j] - alpha_k * v_k[j]
```

```

for j in range (1,dim):
    if x_k[j] < xalt[j]: x_k[j]=xalt[j]
    if x_k[j] > xüst[j]: x_k[j] = xüst[j]
fk.append(f(x_k))
if fk[i] < fbest:
    fbest = fk[i]
    xbest = x_k[:]
elapsed = (time.time() - start)
return (fbest, xbest, elapsed)

```

- Azalan adım uzunluğuyla (4.1) zayıf subgradyant algoritması

```

def ws_diminishing(xalt, xüst, x_0, f, iteration):
    start = time.time()
    x_k = x_0[:]
    global dim
    dim = len(x_0)
    global fbest, xbest, c, elapsed, fk
    c = [999999]
    fbest = f(x_k)
    xbest = x_k[:]
    fk = [f(x_k)]
    for i in range(1, iteration):
        global index
        index = i
        c.append(5-(5/iteration)*i)
        v_k=appr_RE(f, x_k, lamda,aalpha)
        alpha_k=1-(1/iteration)*i
        norm= math.sqrt(sum([math.pow(v_k[j], 2) for j in range(1, dim)]))
        for j in range(1,dim):
            x_k[j]= x_k[j] - alpha_k * (v_k[j])
        for j in range (1,dim):
            if x_k[j] < xalt[j]: x_k[j]=xalt[j]
            if x_k[j] > xüst[j]: x_k[j] = xüst[j]
        fk.append(f(x_k))
        if fk[i] < fbest:
            fbest = fk[i]
            xbest = x_k[:]
    elapsed = (time.time() - start)
    return (fbest, xbest, elapsed)

```

- f^* ve x^* ile adım uzunluğuyla (4.2) zayıf subgradyant algoritması

```
def ws_dynamic_known(xalt, xüst, x_0, f, iteration, fstar, xstar):
    start = time.time()
    x_k = x_0[:]
    global dim
    dim = len(x_0)
    global fbest, xbest, c, elapsed, fk
    c = [999999]
    fbest = f(x_k)
    xbest = x_k[:]
    fk = [f(x_k)]
    for i in range(1, iteration):
        dx = math.sqrt(sum([math.pow(xstar[j] - x_k[j], 2) for j in range(1, dim)]))
        global index
        index = i
        global cüst
        cüst = (fk[i-1] - fstar) / dx
        c.append(cüst*ckatsayi)
        v_k=appr_RE(f, x_k, lamda,aalpha)
        gamma_k = random.uniform(0.1, 1.9)
        norm= math.sqrt(sum([math.pow(v_k[j], 2) for j in range(1, dim)]))
        alpha_k = gamma_k * ((fk[i-1] - fstar - c[i] * dx) / math.pow(norm,2))
        for j in range(1,dim):
            x_k[j]= x_k[j] - alpha_k * v_k[j]
        for j in range (1,dim):
            if x_k[j] < xalt[j]: x_k[j]=xalt[j]
            if x_k[j] > xüst[j]: x_k[j] = xüst[j]
        fk.append(f(x_k))
        if fk[i] < fbest:
            fbest = fk[i]
            xbest = x_k[:]
    elapsed = (time.time() - start)
    return (fbest, xbest, elapsed)
```

- $f^{lev} > f^*$ ile adım uzunluğuyla (4.3) zayıf subgradyant algoritması

```
def ws_dynamic_unknown_greater_flev(xalt, xüst, x_0, f, iteration, fstar, flev):
    start = time.time()
    x_k = x_0[:]
    global dim
    dim = len(x_0)
    global fbest, xbest, c, elapsed, fk
```

```

c = [999999]
fbest = f(x_k)
xbest = x_k[:]
fk = [f(x_k)]
dx = math.sqrt(sum([math.pow(xüst[j] - xalt[j], 2) for j in range(1, dim)]))
for i in range(1, iteration):
    global index
    index = i
    global cüst
    cüst = (fk[i-1] - flev) / dx
    if cüst < 0: return (fbest, xbest)
    c.append(cüst*ckatsayi)
    v_k=appr_RE(f, x_k, lamda,aalpha)
    gamma_k = random.uniform(0.1, 1.9)
    norm= math.sqrt(sum([math.pow(v_k[j], 2) for j in range(1, dim)]))
    alpha_k = gamma_k * ((fk[i-1] - flev - c[i] * dx) / math.pow(norm,2))
    for j in range(1,dim):
        x_k[j]= x_k[j] - alpha_k * v_k[j]
    for j in range (1,dim):
        if x_k[j] < xalt[j]: x_k[j]=xalt[j]
        if x_k[j] > xüst[j]: x_k[j] = xüst[j]
    fk.append(f(x_k))
    if fk[i] < fbest:
        fbest = fk[i]
        xbest = x_k[:]
    elapsed = (time.time() - start)
    return (fbest, xbest, elapsed)

```

- $f^{lev} < f^*$ ile adım uzunluğuyla (4.4) zayıf subgradyant algoritması

```

def ws_dynamic_unknown_lower_flev(xalt, xüst, x_0, f, iteration, fstar, flev):
    start = time.time()
    x_k = x_0[:]
    global dim
    dim = len(x_0)
    global fbest, xbest, c, elapsed, fk
    c = [999999]
    fbest = f(x_k)
    xbest = x_k[:]
    fk = [f(x_k)]
    dx = math.sqrt(sum([math.pow(xüst[j] - xalt[j], 2) for j in range(1, dim)]))
    for i in range(1, iteration):
        global index

```

```

index = i
global cüst
cüst = (fk[i-1] - flev) / dx
c.append(cüst*ckatsayi)
v_k=appr_RE(f, x_k, lamda,aalpha)
gamma_k = random.uniform(0.1, 0.9)
norm= math.sqrt(sum([math.pow(v_k[j], 2) for j in range(1, dim)]))
alpha_k = gamma_k * ((fk[i-1] - flev - c[i] * dx) / math.pow(norm,2))
for j in range(1,dim):
    x_k[j]= x_k[j] - alpha_k * v_k[j]
for j in range (1,dim):
    if x_k[j] < xalt[j]: x_k[j]=xalt[j]
    if x_k[j] > xüst[j]: x_k[j] = xüst[j]
fk.append(f(x_k))
if fk[i] < fbest:
    fbest = fk[i]
    xbest = x_k[:]
elapsed = (time.time() - start)
return (fbest, xbest, elapsed)

```

- f_k^{lev} ile adım uzunluğuyla (4.5) zayıf subgradyant algoritması

```

def ws_dynamic_unknown_dynamic_flev(xalt, xüst, x_0, f, iteration, fstar):
    start = time.time()
    x_k = x_0[:]
    global dim
    dim = len(x_0)
    global fbest, xbest, c, elapsed, fk
    c = [999999]
    fbest = f(x_k)
    xbest = x_k[:]
    fk = [f(x_k)]
    delta_0=f(x_0)*0.15
    lamda_üst= 1.5
    delta_üst= delta_0*1.15
    lamda_alt=0.5
    delta_alt= delta_0*0.85
    delta_k=delta_0
    dx = math.sqrt(sum([math.pow(xüst[i] - xalt[i], 2) for i in range(1, dim)]))
    for i in range(1, iteration):
        flev = min(fk)-delta_k
        global index
        index = i

```

```

global cüst
cüst = (fk[i-1] - flev) / dx
c.append(cüst*ckatsayi)
v_k=appr_RE(f, x_k, lamda,aalpha)
gamma_k = random.uniform(0.1, 0.9)
norm= math.sqrt(sum([math.pow(v_k[j], 2) for j in range(1, dim)]))
alpha_k = gamma_k * ((fk[i-1] - flev - c[i] * dx) / math.pow(norm,2))
for j in range(1,dim):
    x_k[j]= x_k[j] - alpha_k * v_k[j]
for j in range (1,dim):
    if x_k[j] < xalt[j]: x_k[j]=xalt[j]
    if x_k[j] > xüst[j]: x_k[j] = xüst[j]
fk.append(f(x_k))
if fk[i] < fbest:
    fbest = fk[i]
    xbest = x_k[:]
if fk[i]<flev:
    delta_k= min(lamda_üst*delta_k, delta_üst)
else:
    delta_k= max(lamda_alt*delta_k, delta_alt)
elapsed = (time.time() - start)
return (fbest, xbest, elapsed)

```

Küçük Boyutlu Problemlerin Python Kodları

- Crescent

```

def func21(x):
    return max(math.pow(x[1], 2) + math.pow((x[2] - 1), 2) + x[2] - 1, -1 * math.pow(x[1],
2) - math.pow((x[2] - 1), 2) + x[2] + 1)

```

- Mifflin 2

```

def func22(x):
    return -1*x[1]+2*(math.pow(x[1],2)+math.pow(x[2],2)-
1)+1.75*abs(math.pow(x[1],2)+math.pow(x[2],2)-1)

```

- WF

```

def func23(x):
    return max(0.5*(x[1]+((10*x[1])/(x[1]+0.1))+2*math.pow(x[2],2)), 0.5*(-
1*x[1]+((10*x[1])/(x[1]+0.1))+2*math.pow(x[2],2)), 0.5*(x[1]-
((10*x[1])/(x[1]+0.1))+2*math.pow(x[2],2)))

```

- SPIRAL

```
def func24(x):
    return max(math.pow(x[1]-
math.pow(math.pow(x[1],2)+math.pow(x[2],2),0.5)*math.cos(math.pow(x[1],2)+math.
pow(x[2],2)),2)+0.005*(math.pow(x[1],2)+math.pow(x[2],2)),          math.pow(x[2]-
math.pow(math.pow(x[1],2)+math.pow(x[2],2),0.5)*math.sin(math.pow(x[1],2)+math.p
ow(x[2],2)),2)+0.005*(math.pow(x[1],2)+math.pow(x[2],2)))
```

- EVD52

```
def func25(x):
    return max(math.pow(x[1],2)+math.pow(x[2],2)+math.pow(x[3],2)-1,
math.pow(x[1],2)+math.pow(x[2],2)+math.pow(x[3]-2,2), x[1]+x[2]+x[3]-1, x[1]+x[2]-
x[3]-1,          2*math.pow(x[1],3)+6*math.pow(x[2],2)+2*math.pow(5*x[3]-x[1]+1,2),
math.pow(x[1],2)-9*x[3])
```

- PBC3

```
def func26(x):
    fonk=[-99999]
    t=[0]
    y=[0]
    for k in range (1,22):
        t.append(10*((k-1)/20))
        y.append((3/20)*math.exp(-1*t[k])+(1/52)*math.exp(-5*t[k])-(1/65)*math.exp(-
2*t[k])*(3*math.sin(2*t[k])+11*math.cos(2*t[k])))
        fonk.append((x[3]/(x[2]+0.0000000001))*math.exp(-
1*t[k]*x[1])*math.sin(t[k]*x[2])-y[k])
    return max(fonk)
```

- Bard

```
def func27(x):
    fonk=[-999999999]
    u=[0,1,2,3,4,5,6,7,8,7,6,5,4,3,2,1]
    y=[0,0.14,0.18,0.22,0.25,0.29,0.32,0.35,0.39,0.37,0.58,0.73,0.96,1.34,2.10,4.39]
    for k in range (1,16):
        fonk.append(abs(x[1]+ (k/((16-k)*x[2]+u[k]*x[3]+0.0000000001))))
    return max (fonk)
```

- Kowalik-Osborne

```
def func28(x):
    fonk=[-999999999]
    u=[0,4,2,1,0.5,0.25,0.167,0.125,0.1,0.0833,0.0714,0.0625]
    y=[0,0.1957,0.1947,0.1735,0.16,0.0844,0.0627,0.0456,0.0342,0.0323,0.0235,0.0246]
    for k in range (1,12):

fonk.append(abs(((x[1]*(math.pow(u[k],2)+x[2]*u[k]))/(math.pow(u[k],2)+x[3]*u[k]+x
[4]+0.000001))-y[k]))
    return max (fonk)
```

- Polak 6

```
def func29(x):
    fonk=[-99999]
    fonk[1]=math.pow((x[1]-math.pow((x[4]+1),4)),2)+math.pow(x[2]-math.pow(x[1]-
math.pow(x[4]+1,4),4),2)+ 2* math.pow(x[3],2)+ math.pow(x[4],2)
    return max(fonk)
```

- OET5

```
def func30(x):
    return max(abs(x[4]-math.pow(x[1]*math.pow(0.25+0.75*((k-
1)/20),2)+x[2]*(0.25+0.75*((k-1)/20))+x[3],2)-math.pow(0.25+0.75*((k-1)/20),0.5)) for
k in range (1,22))
```

- OET6

```
def func31(x):
    return max(abs(x[1]*math.exp(x[3]*(-0.5+((kk-1)/20)))+(x[2]*math.exp(x[4]*(-
0.5+((kk-1)/20))))-(1/(1+(-0.5+((kk-1)/20))))for kk in range (1,22))
```

- EXP

```
def func32(x):
    return max(((x[1]+x[2]*(-1+((kk-1)/10)))/(1+x[3]*(-1+((kk-
1)/10))+x[4]*math.pow((-1+((kk-1)/10)),2)+x[5]*math.pow((-1+((kk-1)/10)),3)))-
math.exp((-1+((kk-1)/10))) for kk in range (1,22))
```

- PBC1

```
def func33(x):
    return max(abs(((x[1]+x[2]*(-1+(2*(kk-1)/29))+x[3]*math.pow((-1+(2*(kk-1)/29)),2))/(1+x[4]*(-1+(2*(kk-1)/29))+x[5]*math.pow((-1+(2*(kk-1)/29)),2))-((math.pow(math.pow(8*(-1+(2*(kk-1)/29))-1,2)+1,0.5)*math.atan(8*(-1+(2*(kk-1)/29))))/(8*(-1+(2*(kk-1)/29))))for kk in range (1,31))
```

- El-Attar

```
def func35(x):
    return sum(abs(x[1]*math.exp(-1*x[2]*((kk-1)/10))*math.cos(x[3]*3*((kk-1)/10)+x[4])+x[5]*math.exp(-1*x[6]*((kk-1)/10))-(0.5*math.exp(-1*((kk-1)/10))-math.exp(-2*((kk-1)/10))+0.5*math.exp(-3*((kk-1)/10))+1.5*math.exp(-1.5*((kk-1)/10))*math.sin(7*((kk-1)/10))+math.exp(-2.5*((kk-1)/10))*math.sin(5*((kk-1)/10))))for kk in range (1,52))
```

- EVD61

```
def func36(x):
    return max(abs(x[1]*math.exp(-1*x[2]*((kk-1)/10))*math.cos(x[3]*((kk-1)/10)+x[4])+x[5]*math.exp(-1*x[6]*((kk-1)/10))-(0.5*math.exp(-1*((kk-1)/10))-math.exp(-2*((kk-1)/10))+0.5*math.exp(-3*((kk-1)/10))+1.5*math.exp(-1.5*((kk-1)/10))*math.sin(7*((kk-1)/10))+math.exp(-2.5*((kk-1)/10))*math.sin(5*((kk-1)/10))))for kk in range (1,52))
```

- Gill

```
def func37(x):
    return max(sum(math.pow(x[kk]-1,2)for kk in range(1,11))+0.001*sum(math.pow(math.pow(x[kk],2)-0.25,2)for kk in range(1,11)),sum(math.pow(sum(x[kkk]*(kkk-1)*math.pow((kk-1)/29,kkk-2) for kkk in range(2,11))-math.pow(sum(x[kkk]*math.pow((kk-1)/29,kkk-1) for kkk in range(1,11)),2)-1,2) for kk in range (2,31))+math.pow(x[1],2)+math.pow(x[2]-math.pow(x[1],2)-1,2), sum(100*math.pow(x[kk]-math.pow(x[kk-1],2),2)+math.pow(1-x[kk],2) for kk in range (2,11)))
```

- Problem 1

```
def func38(x):
    fonk=[-999999]
    fonk.append(math.pow(x[1],4)+math.pow(x[2],2))
    fonk.append(math.pow(2-x[1],2)+math.pow(2-x[2],2))
```

```

fonk.append(2*math.exp(-1*x[1]+x[2]))
fonk.append(math.pow(x[1],2)-2*x[1]+math.pow(x[2],2)-4*x[2]+4)
fonk.append(2*math.pow(x[1],2)-5*x[1]+math.pow(x[2],2)-2*x[2]+4)
fonk.append(math.pow(x[1],2)+2*math.pow(x[2],2)-4*x[2]+1)
return max(fonk[kk] for kk in range (1,4))+ min(fonk[kk] for kk in range (4,7))

```

- Rosenbrock fonksiyonunun L1 versiyonu

```

def func39(x):
    return abs(x[1]-1)+100*abs(x[2]-abs(x[1]))

```

- Wood fonksiyonunun L1 versiyonu

```

def func40(x):
    return abs(x[1]-1)+100*abs(x[2]-abs(x[1]))+90*abs(x[4]-abs(x[3]))+abs(x[3]-1)+10.1*(abs(x[2]-1)+abs(x[4]-1))+4.95*(abs(x[2]+x[4]-2)-abs(x[2]-x[4]))

```

Büyük Boyutlu Test Problemlerinin Python Kodları

- Aktif yüzlerin sayısı

```

def g(x_62):
    return math.log(abs(x_62)+1)

```

```

def func62(x):
    fonk_max=max((g(x[kk]) for kk in range(1,dim)))
    return max(fonk_max, g(-1*sum(x[kk] for kk in range (1,dim))))

```

- Brown fonksiyonunun differansiyellenemeyen genelleştirilmesi 2

```

def func63(x):
    return sum(math.pow(abs(x[kk]),math.pow(x[kk+1],2)+1)+math.pow(abs(x[kk+1]),math.pow(x[kk],2)+1) for kk in range (1,dim-1))

```

- Chained Mifflin 2

```

def func64(x):
    return sum(-1*x[j]+2*(math.pow(x[j],2)+math.pow(x[j+1],2)-1)+1.75*abs(math.pow(x[j],2)+math.pow(x[j+1],2)-1) for j in range (1,dim-1))

```

- Chained crescent I

```
def func65(x):
    return max(sum(math.pow(x[j],2)+math.pow(x[j+1]-1,2)+x[j+1]-1 for j in range
(1,dim-1)),sum(-1*math.pow(x[j],2)-math.pow(x[j+1]-1,2)+x[j+1]+1 for j in range
(1,dim-1)))
```

- Chained crescent II

```
def func66(x):
    return sum(max(math.pow(x[kk],2)+math.pow(x[kk+1]-1,2)+x[kk+1]-1,-
1*math.pow(x[kk],2)-math.pow(x[kk+1]-1,2)+x[kk+1]+1) for kk in range (1,dim-1))
```

- Test29'daki Problem 6

```
def func67(x_func):
    x=x_func[:]
    x.append(0)
    return max(abs((3-2*x[kk])*x[kk]+1-x[kk-1]-x[kk+1]))for kk in range (1,dim))
```

- Test29'daki Problem 17

```
def func68(x):
    return max(abs((5-((((kk-1)//5)+1)*(1-math.cos(x[kk]*math.pi/180)))-
math.sin(x[kk]*math.pi/180))-sum(math.cos(x[kkk]*math.pi/180) for kkk in range
(5*((kk-1)//5)+1,5*((kk-1)//5)+6))) for kk in range (1,dim))
```

- Test29'daki Problem 19

```
def func69(x_func):
    x=x_func[:]
    x.append(0)
    return max(math.pow((3-2*x[kk])*x[kk]-x[kk-1]-2*x[kk+1]+1,2) for kk in range
(1,dim))
```

- Test29'daki Problem 20

```
def func70(x_func):
    x=x_func[:]
    x.append(0)
    return max(abs((0.5*x[kk]-3)*x[kk]-1+x[kk-1]+2*x[kk+1]) for kk in range (1,dim))
```

- Test29'daki Problem 22

```
def func71(x_func):
    x=x_func[:]
    x.append(0)
    return max(abs(2*x[kk]+(1/(2*math.pow(dim,2)))*math.pow(x[kk]+(kk/dim)+1,3)-
x[kk-1]-x[kk+1]) for kk in range (1,dim))
```

- Test29'daki Problem 24

```
def func72(x_func):
    x=x_func[:]
    x.append(1)
    return max(abs(2*x[kk]+(10/math.pow(dim,2))*math.sinh(10*x[kk]*math.pi/180)-
x[kk-1]-x[kk+1]) for kk in range (1,dim))
```

- DC Maxl

```
def func73(x):
    return (dim-1)*max(abs(x[kk]) for kk in range (1,dim))-sum(abs(x[kk]) for kk in range
(1,dim))
```

- DC Maxq

```
def func74(x):
    return dim*max(math.pow(x[kk],2) for kk in range (1,dim))- sum(math.pow(x[kk],2)
for kk in range (1,dim))
```

- Problem 6

```
def func75(x):
    return 10*max(abs(sum((x[kk]-(1/(dim-1)))*math.pow(0.1*jj,kk-1) for kk in range
(1,dim))) for jj in range (1,11))-sum(abs(sum((x[kk]-(1/(dim-1)))*math.pow(0.1*jj,kk-1)
for kk in range (1,dim))) for jj in range (1,11))
```

- Problem 7

```
def func76(x):
    return 10*max(abs(sum(abs(x[kk]-(1/(dim-1)))*math.pow(0.1*jj,kk-1) for kk in range
(1,dim))) for jj in range (1,11))-sum(abs(sum(abs(x[kk]-(1/(dim-1)))*math.pow(0.1*jj,kk-1) for kk in range (1,dim))) for jj in range (1,11))
```

EK-2 Küçük Boyutlu Test Problemleri

İsim	n	Amaç Fonksiyonu	f^*	x^*	x^1
Crescent ¹	2	$\max\{x_1^2 + (x_2 - 1)^2 + x_2 - 1, -x_1 - (x_2 - 1)^2 + x_2 + 1\}$	0	(0,0)	(-1.5,2)
Mifflin 2 ²	2	$-x_1 + 2(x_1^2 + x_2^2 - 1) + 1.75 x_1^2 + x_2^2 - 1 $	-1	(1,0)	(-1,-1)
WF ³	2	$\max\left\{\frac{1}{2}\left(x_1 + \frac{10x_1}{x_1+10} + 2x_2^2\right), \frac{1}{2}\left(-x_1 + \frac{10x_1}{x_1+10} + 2x_2^2\right), \frac{1}{2}\left(x_1 - \frac{10x_1}{x_1+10} + 2x_2^2\right)\right\}$	0	(0,0)	(3,1)
SPIRAL ³	2	$\max\{f_1(x), f_2(x)\}$ $f_1(x) = \left(x_1 - \sqrt{x_1^2 + x_2^2} \cos \sqrt{x_1^2 + x_2^2}\right)^2 + 0.005(x_1^2 + x_2^2)$ $f_2(x) = \left(x_2 - \sqrt{x_1^2 + x_2^2} \sin \sqrt{x_1^2 + x_2^2}\right)^2 + 0.005(x_1^2 + x_2^2)$	0	(0,0)	(1.411831, -4.79462)
EVD52 ³	3	$f(x) = \max_{1 \leq i \leq 6} f_i(x)$ $f_1(x) = x_1^2 + x_2^2 + x_3^2 - 1$ $f_2(x) = x_1^2 + x_2^2 + (x_3 - 2)^2$ $f_3(x) = x_1 + x_2 + x_3 - 1$ $f_4(x) = x_1 + x_2 - x_3 - 1$ $f_5(x) = 2x_1^3 + 6x_2^2 + 2(5x_3 - x_1 + 1)^2$ $f_6(x) = x_1^2 - 9x_3$	3.5991193	(0.3283,0,0.1313)	(1,1,1)
PBC3 ³	3	$f(x) = \max_{1 \leq i \leq 21} f_i(x) $ $f_i(x) = \frac{x_3}{x_2} e^{-t_i x_1} \sin(t_i x_2) - y_i$ $y_i = \frac{3}{20} e^{-t_i} + \frac{1}{52} e^{-5t_i} - \frac{1}{65} e^{-2t_i} (3 \sin(2t_i) + 11 \cos(2t_i))$ $t_i = 10(i - 1)/20$	$0.42021427 \cdot 10^{-2}$	(0.9516,0.8761,0.1623)	(1,1,1)
Bard ³	3	$f(x) = \max_{1 \leq i \leq 15} f_i(x) $	$0.50816327 \cdot 10^{-1}$	(0.0535,1.5106,1.9894)	(1,1,1)

İsim	n	Amaç Fonksiyonu	f^*	x^*	x^1
		$f_i(x) = y_i - x_1 - \frac{i}{(16-i)x_2 + u_i x_3}$ $u = (1,2,3,4,5,6,7,8,7,6,5,4,3,2,1)$ $y =$ $(0.14,0.18,0.22,0.25,0.29,0.32,0.35,0.39,0.37,0.58,0.73,0.96,1.34,2.10,4.39)$			
Polak 6 ⁴	4	$f(x) = \max_{1 \leq i \leq 4} f_i(x)$ $f_1(x) = (x_1 - (x_4 + 1)^4 + (x_2 - (x_1 - (x_4 + 1)^4)^4)^2 + 2x_3^2 + x_4^2 - 5(x_1 - (x_4 + 1)^4)$ $f_2(x) = f_1(x) + 10((x_1 - (x_4 + 1)^4)^2 + (x_2 - (x_1 - (x_4 + 1)^4)^4)^2 + x_3^2 + x_4^2 + (x_1 - (x_4 + 1)^4) - (x_2 - (x_1 - (x_4 + 1)^4)^4) + x_3 - x_4 - 8)$ $f_3(x) = f_1(x) + 10((x_1 - (x_4 + 1)^4)^2 + 2(x_2 - (x_1 - (x_4 + 1)^4)^4)^2 + x_3^2 + 2x_4^2 - (x_1 - (x_4 + 1)^4) - x_4 - 10)$ $f_4(x) = f_1(x) + 10((x_1 - (x_4 + 1)^4)^2 + (x_2 - (x_1 - (x_4 + 1)^4)^4)^2 + x_3^2 + 2(x_1 - (x_4 + 1)^4) - (x_2 - (x_1 - (x_4 + 1)^4)^4) - x_4 - 5)$	-44	(0,1,2,-1)	(0,0,0,0)
El-Attar ³	6	$f(x) = \sum_{i=1}^{50} x_1 e^{x_2 l_i} \cos(x_3 3t_i + x_4) + x_5 e^{-x_6 t_i} - y_i $ $y_i = 0.5e^{t_i} - e^{-2t_i} + 0.5e^{-3t_i} + 1.5e^{-1.5t_i} \sin(7t_i)$ $t_i = (i - 1)/10$	0.5598131	(2.2407,1.8577,6.7701,-1.6449,0.1659,0.7423)	(2,2,7,0,-2,1)
Gill ³	10	$f(x) = \max\{f_1(x), f_2(x), f_3(x)\}$ $f_1(x) = \sum_{i=1}^{10} (x_i - 1)^2 + 10^{-3} \sum_{i=1}^{10} \left(x_i^2 - \frac{1}{4}\right)^2$	9.7857721	(-0.6022,0.4907,0.3096,0.1416,0.0542,0.0287,0.0197,0.0137,0.0087,0.0045)	(-0.1,-0.1,...,-0.1)

İsim	n	Amaç Fonksiyonu	f^*	x^*	x^1
		$f_2(x) = \sum_{i=2}^{30} \left[\sum_{j=2}^{10} x_j(j-1) \left(\frac{i-1}{29} \right)^{j-2} - \left(\sum_{j=1}^{10} x_j \left(\frac{i-1}{29} \right)^{j-2} \right)^2 - 1 \right]^2 + x_1^2 + (x_2 - x_1^2 - 1)^2$ $f_3(x) = \sum_{i=2}^{10} [100(x_i - x_{i-1}^2)^2 + (1 - x_i)^2]$			
Problem 1 ⁵	2	$f(x) = \max_{1 \leq i \leq 3} f_i(x) + \min_{4 \leq i \leq 6} f_i(x)$ $f_1(x) = x_1^4 + x_2^2$ $f_2(x) = (2 - x_1)^2 + (2 - x_2)^2$ $f_3(x) = 2e^{-x_1+x_2}$ $f_4(x) = x_1^2 - 2x_1 + x_2^2 - 4x_2 + 4$ $f_5(x) = 2x_1^2 + 5x_1 + x_2^2 - 2x_2 + 4$ $f_6(x) = x_1^2 + 2x_2^2 - 4x_2 + 1$	2	(1,1)	(2,2)
Rosenbrock fonksiyonu nun L1 versiyonu ⁵	2	$f(x) = x_1 - 1 + 100 x_2 - x_1 $	0	(1,1)	(-1.2,1)
Wood fonksiyonu nun L1 versiyonu ⁵	4	$f(x) = x_1 - 1 + 100 x_2 - x_1 + 90 x_4 - x_3 + x_3 - 1 + 10.1(x_2 - 1 + x_4 - 1) + 4.95(x_2 + x_4 - 2 - x_2 - x_4)$	0	(1,1,1,1)	(1,3,3,1)
EXP ³	5	$f(x) = \max_{1 \leq i \leq 21} f_i(x)$ $f_i(x) = \frac{x_1 + x_2 t_i}{1 + x_3 t_i + x_4 t_i^2 + x_5 t_i^3} - e^{t_i}$ $t_i = -1 + (i - 1)/10$	0,12237125.10 ⁻³	(0.9999,0.2536,-0.7466,0.2452,-0.0375)	(0.5,0,0,0,0)

İsim	n	Amaç Fonksiyonu	f^*	x^*	x^1
Kowalik-Osborne ³	4	$f(x) = \max_{1 \leq i \leq 11} f_i(x) $ $f_i(x) = \frac{x_1(u_i^2 + x_2 u_i)}{u_i^2 + x_3 u_i + x_4} - y_i$ $u = (4, 2, 1, 0.5, 0.25, 0.167, 0.125)$ $y = (0.1957, 0.1947, 0.1735, 0.16, 0.0844, 0.0627, 0.0456, 0.0342, 0.0323, 0.0235, 0.02456)$	$0.80843684 \cdot 10^{-2}$	(0.1846, 0.1052, 0.0196, 0.1118)	(0.25, 0.39, 0.415, 0.39)
OET5 ³	4	$f(x) = \max_{1 \leq i \leq 21} f_i(x) $ $f_i(x) = x_4 - (x_1 t_i^2 + x_2 t_i + x_3)^2 - \sqrt{t_i}$ $t_i = 0.25 + 0.75(i - 1)/20$	$0.26359735 \cdot 10^{-2}$	(0.0876, -0.497, 1.1155, 1.4963)	(1, 1, 1, 1)
OET6 ³	4	$f(x) = \max_{1 \leq i \leq 11} f_i(x) $ $f_i(x) = x_1 e^{x_3 t_i} + x_2 e^{x_4 t_i} - \frac{1}{1 + t_i}$ $t_i = -0.5 + (i - 1)/20$	$0.20160753 \cdot 10^{-2}$	(0.0987, 0.9009, -4.0619, -0.6477)	(1, 1, -3, -1)
PBC1 ³	5	$f(x) = \max_{1 \leq i \leq 21} f_i(x) $ $f_i(x) = \frac{x_1 + x_2 t_i + x_3 t_i^2}{1 + x_4 t_i + x_5 t_i^2} - \frac{\sqrt{(8t_i - 1)^2 + 1} \arctan(8t_i)}{8t_i}$ $t_i = -1 + 2(i - 1)/29$	$0.22340496 \cdot 10^{-1}$	(1.4136, -10.5797, 40.7117, -4.0213, 27.6150)	(0, -1, 10, 1, 10)
EVD61 ³	6	$f(x) = \max_{1 \leq i \leq 51} f_i(x) $ $f_i(x) = x_1 e^{-x_2 t_i} \cos(x_3 t_i + x_4) + x_5 e^{-x_6 t_i} - y_i$ $y_i = 0.5 e^{-t_i} - e^{-2t_i} + 0.5 e^{-3t_i} + 1.5 e^{-1.5t_i} \sin(7t_i) - e^{-2.5t_i} \sin(5t_i)$ $t_i = (i - 1)/10$	$0.34904926 \cdot 10^{-1}$	(2.2759, 1.8993, 6.8482, -1.6503, 0.1457, 0.517)	(2, 2, 7, 0, -2, 1)

Kaynak: ¹(Kiwiel, 1985), ²(Mäkelä ve Neittaanmäki, 1992), ³(Lukšan ve Vlček, 2000), ⁴(Polak vd., 1991), ⁵(Bagirov, 2002)

EK-3 Büyük Boyutlu Test Problemleri

İsim	n	Amaç Fonksiyonu	f^*	x^*	x^1
Aktif yüzlerin sayısı ¹	-	$f(x) = \max_{1 \leq i \leq n} \{g(-\sum_{i=1}^n x_i), g(x_i)\}$ $g(y) = \ln(y + 1)$	0	(0,0,...,0)	(1,1,...,1)
Brown fonksiyonunun differansiyellenemeyen genelleştirilmesi 2 ²	-	$f(x) = \sum_{i=1}^{n-1} (x_i ^{x_{i+1}^2+1} + x_{i+1} ^{x_i^2+1})$	0	(0,0,...,0)	$\text{mod}(i, 2) = 0$ ise $x_i^1 = 1$ $\text{mod}(i, 2) = 1$ ise $x_i^1 = -1$
Chained mifflin 2 ²	50	$f(x) = \sum_{i=1}^{n-1} (-x_i + 2(x_i^2 + x_{i+1}^2 - 1) + 1.75 x_i^2 + x_{i+1}^2 - 1)$	-34.795	-	(1,1,...,1)
	200		-140.86	-	
Chained crescent I ²	-	$f(x) = \max\{\sum_{i=1}^{n-1} (x_i^2 + (x_{i+1} - 1)^2 + x_{i+1} - 1), \sum_{i=1}^{n-1} (-x_i^2 - (x_{i+1} - 1)^2 + x_{i+1} + 1)\}$	0	(0,0,...,0)	$\text{mod}(i, 2) = 0$ ise $x_i^1 = 2$ $\text{mod}(i, 2) = 1$ ise $x_i^1 = -1.5$
Chained crescent II ²	-	$f(x) = \sum_{i=1}^{n-1} \max\{(x_i^2 + (x_{i+1} - 1)^2 + x_{i+1} - 1), (-x_i^2 - (x_{i+1} - 1)^2 + x_{i+1} + 1)\}$	0	(0,0,...,0)	$\text{mod}(i, 2) = 0$ ise $x_i^1 = 2$ $\text{mod}(i, 2) = 1$ ise $x_i^1 = -1.5$ $\forall i = 1, \dots, n$
Test29'daki Problem 6 ³	-	$f(x) = \max_{1 \leq i \leq n} (3 - 2x_i) + 1 - x_{i-1} - x_{i+1} $ $x_0 = x_{n+1} = 0$	0	-	(-1,-1,...,-1)
Test29'daki Problem 17 ³	-	$f(x) = \max_{1 \leq i \leq n} 5 - (j+1)(1 - \cos x_i) - \sin x_i - \sum_{k=5j+1}^{5j+5} \cos x_k $ $j = \text{div}(i - 1, 5)$	0	(0,0,...,0)	(1/n, 1/n, ..., 1/n)
Test29'daki Problem 19 ³	-	$f(x) = \max_{1 \leq i \leq n} ((3 - 2x_i)x_i - x_{i-1} - 2x_{i+1} + 1)^2$ $x_0 = x_{n+1} = 0$	0	-	(-1,-1,...,-1)
Test29'daki Problem 20 ³	-	$f(x) = \max_{1 \leq i \leq n} (0.5x_i - 3)x_i - 1 + x_{i-1} + 2x_{i+1} $	0	-	(-1,-1,...,-1)

İsim	n	Amaç Fonksiyonu	f^*	x^*	x^1
		$x_0 = x_{n+1} = 0$			
Test29'daki Problem 22 ³	-	$f(x) = \max_{1 \leq i \leq n} \left 2x_i + \frac{1}{2(n+1)^2} \left(x_i + \frac{i}{n+1} + 1 \right)^3 - x_{i-1} - x_{i+1} \right $ $x_0 = x_{n+1} = 0$	0	-	$x_i^1 = \frac{i}{n+1} \left(\frac{i}{n+1} - 1 \right) \forall i = 1, \dots, n$
Test29'daki Problem 24 ³	-	$f(x) = \max_{1 \leq i \leq n} \left 2x_i + \frac{10}{(n+1)^2} \sinh(10x_i) - x_{i-1} - x_{i+1} \right $ $x_0 = x_{n+1} = 0$	0	-	(1,1,...,1)
DC Maxl ⁴	-	$f(x) = n \max_{1 \leq i \leq n} x_i - \sum_{i=1}^n x_i $	0	$(\alpha, \alpha, \dots, \alpha)$ $\alpha \in R$	$x_i^1 = i \forall i = 1, \dots, n/2$ $x_i^1 = -i \forall i = \frac{n}{2} + 1, \dots, n$
DC Maxlq ⁵	-	$f(x) = (n+1) \max_{1 \leq i \leq n} x_i^2 - \sum_{i=1}^n x_i^2$	0	(0,0,...,0)	-
⁵ 'deki Problem 6	-	$f(x) = 10 \max_{1 \leq j \leq 10} \left\{ \left \sum_{i=1}^n (x_i - x_i^*) t_j^{i-1} \right \right\} - \sum_{j=1}^{10} \left \sum_{i=1}^n (x_i - x_i^*) t_j^{i-1} \right $ $t_j = \max_{1 \leq j \leq 10} \{0.001, 0.1j\}$	0	(1/n,...,1/n)	-
⁵ 'deki Problem 7	-	$f(x) = 10 \max_{1 \leq j \leq 10} \left\{ \left \sum_{i=1}^n x_i - x_i^* t_j^{i-1} \right \right\} - \sum_{j=1}^{10} \left \sum_{i=1}^n x_i - x_i^* t_j^{i-1} \right $ $t_j = \max_{1 \leq j \leq 10} \{0.001, 0.1j\}$	0	(1/n,...,1/n)	-

Kaynak: ¹(Grothey, 2001), ²(Haarala vd., 2004), ³(Lukšan vd., 2002), ⁴(Bagirov, 2002), ⁵(Bagirov ve Ugon, 2011)

- Herhangi bir değişken sayısı