Zeki Optimizasyon Teknikleri

Ara sınav	- 25%
Ödev (Haftalık)	- 10%
Ödev Sunumu (Haftalık)	- 5%
Final (Proje Sunumu)	- 60%

- Dönem sonuna kadar bir optimizasyon tekniğiyle uygulama geliştirilecek

(Örn: Zaman çizelgeleme, en kısa yol bulunması, denetleyici tasarımı, network optimizasyonu vb.)

- Hazırlanacak proje önerileri ara sınav tarihine kadar bildirilecek

e-posta : akcayol@gazi.edu.tr
dersin web adresi : http://w3.gazi.edu.tr/~akcayol

Zeki Optimizasyon Teknikleri

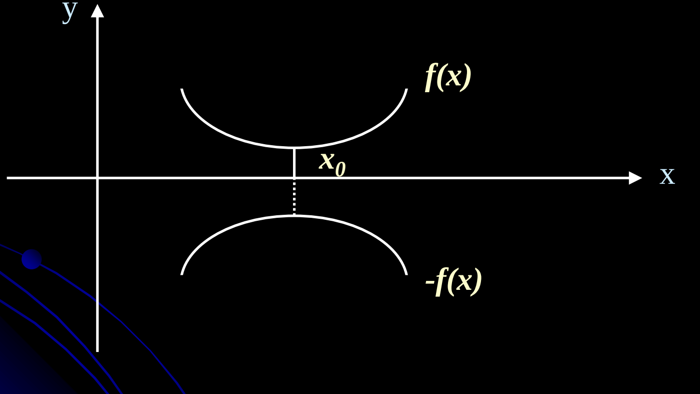
- Optimizasyona Giriş
- Geleneksel Optimizasyon Yöntemleri
- Karınca Algoritması
- Tavlama Benzetimi
- Tabu Arama
- Genetik Algoritma
- Yapay Sinir Ağları

Optimizasyon

- Optimizasyon, verilen şartlar altında en iyi sonucun elde edilmesi işidir.
- Optimizasyon alanındaki en önemli gelişmeler 18.yy'da Newton ve Lagrange tarafından yapılmıştır.
- Bir sistemin planlanmasında hedef, istenen karı maksimize yada gerekli çabayı minimize etmektir.
- İstenen kar veya gerekli çaba, karar değişkenlerinin bir fonksiyonu olarak ifade edilir.
- Optimizasyon sürecinde bu fonksiyonun minimum veya maksimum değerini oluşturan şartlar bulunur.

Minimum, Maksimum

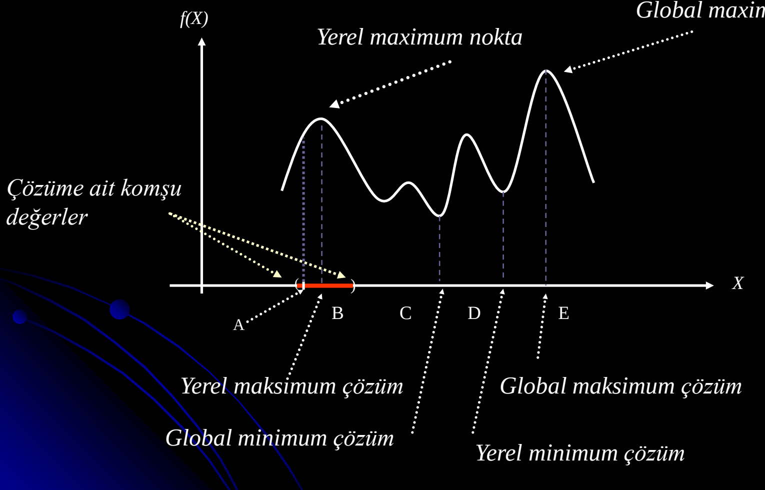
- Bir x_0 noktası $f(x)$ fonksiyonunun minimum değeri ise, aynı nokta $-f(x)$ fonksiyonunun maksimum değeridir.



KLASİK OPTİMİZASYON

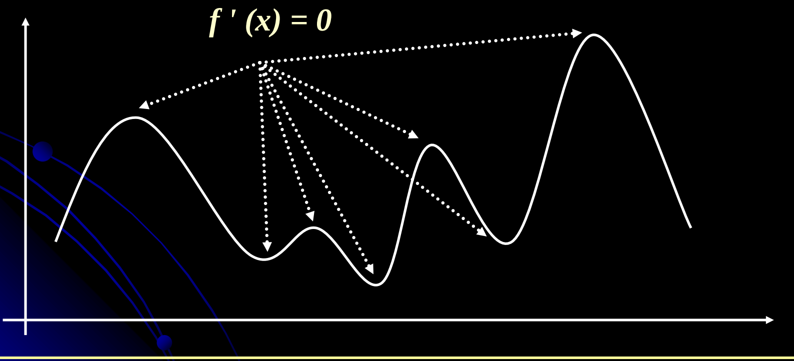
- Fonksiyonlar sürekli ve türevlenebilir olmalıdır.
- Çok küçük pozitif ve negatif bütün h değerleri için $f(x_0) \leq f(x_0 + h)$ ise $f(x)$ fonksiyonu x_0'da yerel minimuma sahiptir. $f(x_0) \geq f(x_0 + h)$ ise $f(x)$ fonksiyonu x_0'da yerel maksimuma sahiptir.
- $f(x)$ fonksiyonu tanımlı olduğu bölgede bütün x'ler için $f(x_0) \leq f(x)$ ise, $f(x)$ fonksiyonu x_0'da mutlak veya tanımlı olduğu bölgede minimuma sahiptir.
- Eğer $f(x_0) \geq f(x)$ ise, $f(x)$ fonksiyonu x_0'da mutlak veya bölgesel maksimuma sahiptir.

KLASİK OPTİMİZASYON

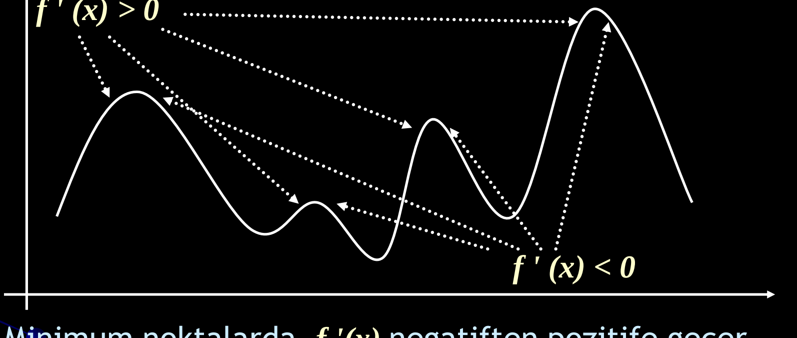


TEK DEĞİŞKENLİ FONKSİYONLAR

- $\delta > 0$ ve $|x - x_0| < \delta$ iken $f(x) \geq f(x_0)$ ise x_0 yerel minimumdur. δ bölge aralığını ifade etmektedir.
- Bütün x 'ler için $f(x) \geq f(x_0)$ ise x_0 global minimumdur.
- Minimum ve maksimumların bulunmasında 1. ve 2. derece türevler kullanılır.



TEK DEĞİŞKENLİ FONKSİYONLAR



- Minimum noktalarda $f'(x)$ negatiften pozitife geçer.
- Maksimum noktalarda $f'(x)$ pozitiften negatife geçer.
- Minimum ve maksimum noktalarda $f'(x) = 0$ olur.
- $f''(x) > 0$ ise bulunulan nokta minimum $f''(x) < 0$ ise bulunulan nokta maksimumdur.

ÖRNEK

$f(x) = x^3 - 2x^2 + x + 1$ fonksiyonunun minimum ve maksimum noktalarını bulunuz.

$f'(x) = 3x^2 - 4x + 1$

$f'(x) = 0$

$3x^2 - 4x + 1 = 0$

$(3x-1)(x-1) = 0$

$x = 1/3, \quad x = 1$

$f''(x) = 6x - 4$

$f''(1/3) = -2$ (yerel maksimum)

$f''(1) = 2$ (yerel minimum)

ÇOK DEĞİŞKENLİ FONKSİYONLAR

- $f(x_1, x_2, \dots, x_n) = f(x)$ şeklinde ifade edilir.
- Gradyan fonksiyonu $\nabla f(x) = (\partial f / \partial x_1, \partial f / \partial x_2, \dots, \partial f / \partial x_n)$ şeklinde gösterilir.
- Hessian matrisi ise aşağıdaki gibidir.

$$H\Big|_{x_0} = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1 x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_1 x_n} \\ \dfrac{\partial^2 f}{\partial x_2 x_1} & \dfrac{\partial^2 f}{\partial x_2^2} & \cdots & \dfrac{\partial^2 f}{\partial x_2 x_n} \\ \vdots & \vdots & & \vdots \\ \dfrac{\partial^2 f}{\partial x_n x_1} & \dfrac{\partial^2 f}{\partial x_n x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

ÇOK DEĞİŞKENLİ FONKSİYONLAR

- Hessian matrisin tüm minor determinantlarının pozitif olması minimum için yeterlidir.

$$H = \begin{bmatrix} h11 & h12 & h13 \\ h21 & h22 & h23 \\ h31 & h32 & h33 \end{bmatrix}$$

$$H_1 = h11 \qquad H_2 = \begin{bmatrix} h11 & h12 \\ h21 & h22 \end{bmatrix} \qquad H_3 = \begin{bmatrix} h11 & h12 & h13 \\ h21 & h22 & h23 \\ h31 & h32 & h33 \end{bmatrix}$$

ÖRNEK

$f(x_1, x_2, x_3) = x_1^2 - x_2^2 + x_3^2 - x_2 x_3 - 2x_1 x_3 + 4x_1 + 10$

fonksiyonunun minimum olduğu noktaları bulunuz.

$\nabla f(x_0) = 0$

$\partial f / \partial x_1 = 0$	$\partial f / \partial x_2 = 0$	$\partial f / \partial x_3 = 0$
$2x_1 - 2x_3 + 4 = 0$	$-2x_2 - x_3 = 0$	$2x_3 - x_2 - 2x_1 = 0$
$x_1 = -10$	$x_2 = 4$	$x_3 = -8$

$$H = \begin{bmatrix} 2 & 0 & -2 \\ 0 & -2 & -1 \\ -2 & -1 & 2 \end{bmatrix} \qquad \begin{aligned} H_1 &= 2 \\ H_2 &= -4 \\ H_3 &= -2 \end{aligned}$$

Minor determinantlarından görüldüğü gibi $f(-10, 4, -8)$ geçiş noktasıdır.

ÖRNEK

$f(x) = x_1^3 + x_2^2 + x_3^2 - 12x_1 - 8x_2 - 12x_3 + 100$
fonksiyonunun minimum olduğu noktaları bulunuz.

$\nabla f(x_0) = 0$

$\partial f / \partial x_1 = 0$	$\partial f / \partial x_2 = 0$	$\partial f / \partial x_3 = 0$
$3x_1^2 - 12 = 0$	$2x_2 - 8 = 0$	$2x_3 - 12 = 0$
$x_1 = 2, -2$	$x_2 = 4$	$x_3 = 6$

$$H\Big|_{(2,4,6)} = \begin{bmatrix} 6x_1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

$H_1 = 6x_1 = 12(2), -12(-2)$

$H_2 = 6x_1 . 2 = 24(2), -24(-2)$

$H_3 = 6x_1 . 2 . 2 = 48(2), -48(-2)$

Tüm minor determinantlar pozitif olduğu için
$f(2, 4, 6)$ minimum noktadır.

DOĞRUSAL PROGRAMLAMA (DP)

- DP, amaç fonksiyonu ve kısıtlayıcıları karar değişkenlerinin doğrusal fonksiyonu biçiminde olan problemlerle uğraşan optimizasyon tekniğidir.
- İlk olarak 1947 yılında George Danting tarafından ortaya atıldı.

DOĞRUSAL PROGRAMLAMA (DP)

- Genel bir doğrusal programlama örneği aşağıdaki gibidir;

$$\begin{array}{ll} \textit{Min} & f(x_1, x_2, \ldots, x_n) = c_1x_1 - c_2x_2 + \ldots + c_nx_n \\ \textit{Kısıtlar} & a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \geq b_1 \\ & a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \geq b_2 \\ & \ldots \quad \ldots \quad \ldots \\ & a_{k1}x_1 + a_{k2}x_2 + \ldots + a_{kn}x_n \geq b_k \\ & x_1 \geq 0,\ x_2 \geq 0, \ldots, x_n \geq 0, \end{array}$$

- Burada $c_1x_1 - c_2x_2 + \ldots + c_nx_n$ amaç fonksiyonu (objective or criterion function) olup minimize edilmek istenmektedir ve z ile gösterilir.
- $c_1, c_2, \ldots, c_n$ maliyet katsayılarıdır (cost coefficients)
- $x_1, x_2, \ldots, x_n$ ise karar değişkenleridir (decision variables).

G. Ü. Bilgisayar Mühendisliği Bölümü

DOĞRUSAL PROGRAMLAMA (DP)

- $\sum_{j=1}^{n} a_{ij} \cdot x_j > b_i$ i.kısıtı ifade eder.
- $x_1, x_2, \ldots, x_n \geq 0$ negatif olmayan kısıtlardır.
- Bütün şartları sağlayan $x_1, x_2, \ldots, x_n$ uygun çözüm noktaları veya uygun çözüm vektörüdür.
- Uygun çözüm noktalarının tamamına uygun çözüm alanı veya uygun çözüm uzayı denir.

G. Ü. Bilgisayar Mühendisliği Bölümü

ÖRNEK

Min $2x_1 + 5x_2$

Kısıtlar $x_1 + x_2 \geq 6$

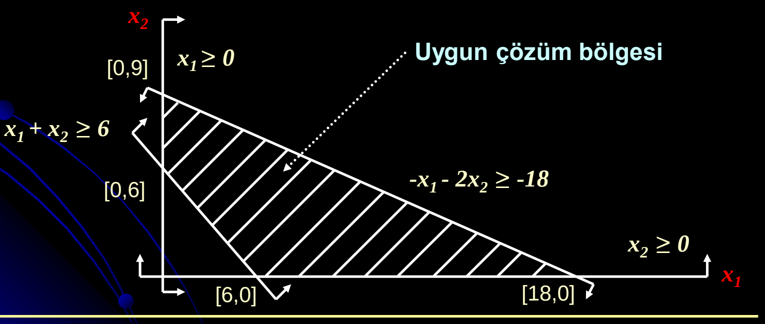
$-x_1 - 2x_2 \geq -18$

$x_1, x_2 \geq 0$

- Optimal çözüm uç noktalardadır.

- Optimal çözüm amaç fonksiyonun katsayılarının yönü kullanılarak bulunur.

- Katsayı vektörü uygun çözüm yönünde değilse amaç fonksiyonu –katsayı yönünde hareket ettirilir, değilse katsayı yönünde hareket ettirilir



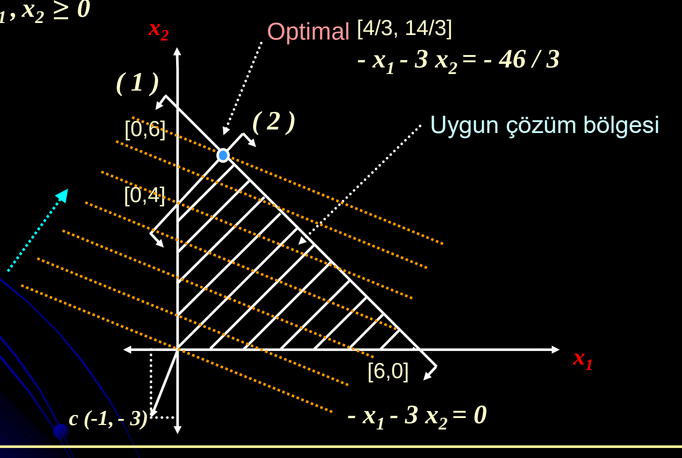
ÖRNEK

Min $-x_1 - 3x_2$

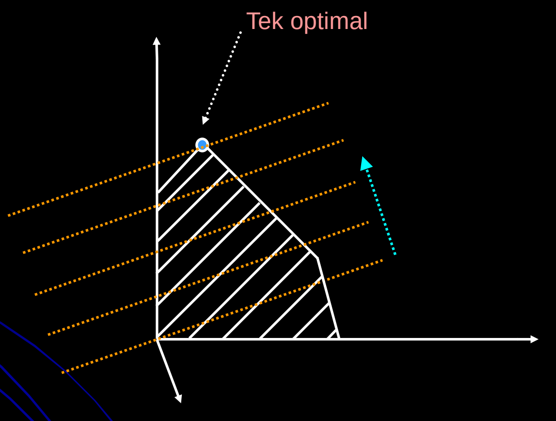
Kısıtlar $x_1 + x_2 \leq 6$ *(1)*

$-x_1 + 2x_2 \leq 8$ *(2)*

$x_1, x_2 \geq 0$



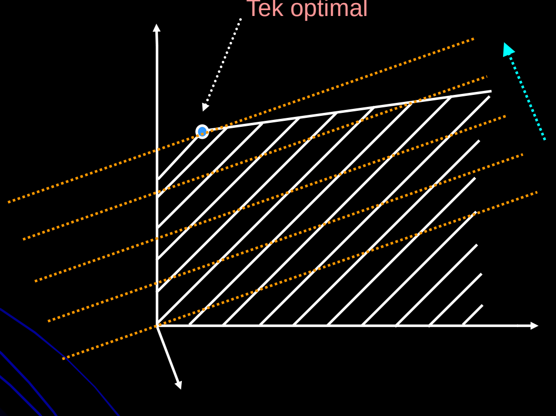
DOĞRUSAL PROGRAMLAMA (DP)



Kapalı bölge

G. Ü. Bilgisayar Mühendisliği Bölümü

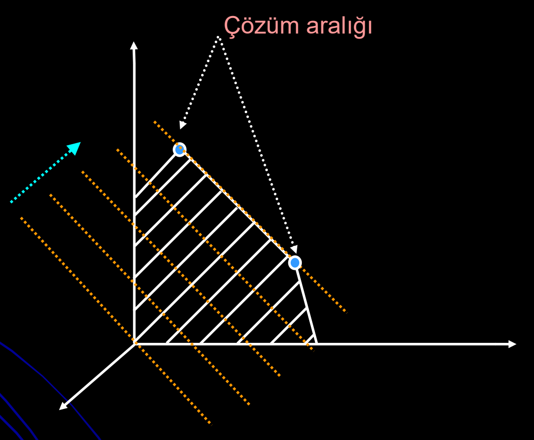
DOĞRUSAL PROGRAMLAMA (DP)



Açık bölge

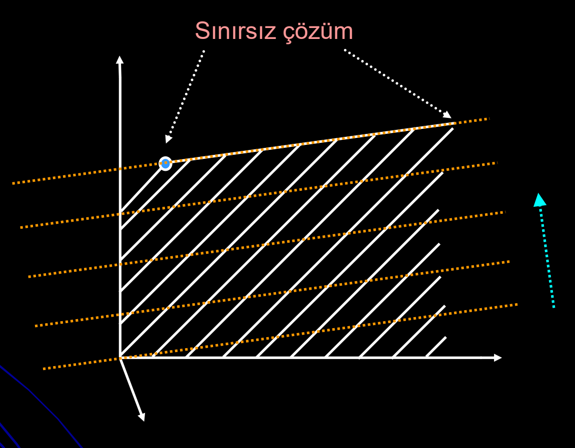
G. Ü. Bilgisayar Mühendisliği Bölümü

DOĞRUSAL PROGRAMLAMA (DP)



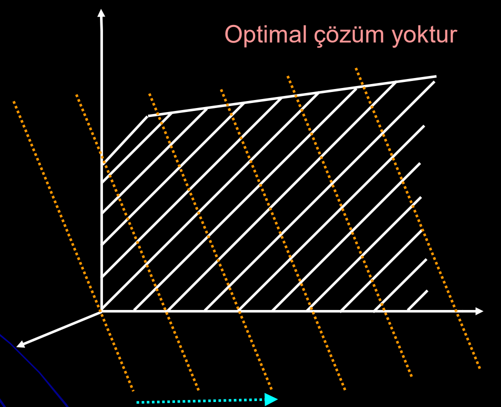
Kapalı bölge

DOĞRUSAL PROGRAMLAMA (DP)



Açık bölge

DOĞRUSAL PROGRAMLAMA (DP)



DOĞRUSAL PROGRAMLAMA (DP)

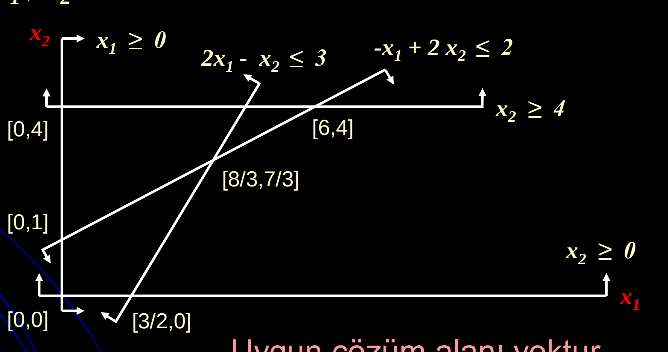
Min $-x_1 - 3x_2$

Kısıtlar $-x_1 + 2x_2 \leq 2$

$2x_1 - x_2 \leq 3$

$x_2 \geq 4$

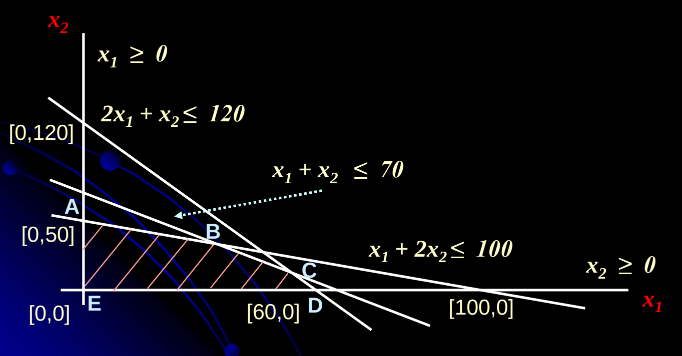
$x_1, x_2 \geq 0$



Uygun çözüm alanı yoktur

ÖRNEK

Maks	$0.3\,x_1 + 0.2\,x_2$	
Kısıtlar	$x_1 , x_2 \geq 0$	(1,2)
	$x_1 + x_2 \leq 70$	(3)
	$x_1 + 2x_2 \leq 100$	(4)
	$2x_1 + x_2 \leq 120$	(5)

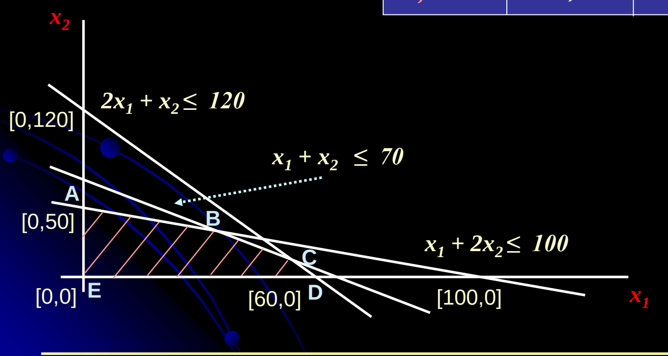


A	10
B	18
*C	19
D	18
E	0

ÖRNEK (devam)

Maks	$0.3\,x_1 + 0.2\,x_2$	
Kısıtlar	$x_1 , x_2 \geq 0$	(1,2)
	$x_1 + x_2 \leq 70$	(3)
	$x_1 + 2x_2 \leq 100$	(4)
	$2x_1 + x_2 \leq 120$	(5)

KÖŞE	KOOR.	KÖŞE	KOOR.
1 , 2 +	0 , 0	2 , 5 +	60 , 0
1 , 3	0 , 70	3 , 4 +	40 , 30
1 , 4 +	0 , 50	3 , 5 + *	50 , 20
1 , 5	0 , 120	4 , 5	140/3,80/3
2 , 3	70 , 0		
2 , 4	100 , 0		



1,2	0
1,4	10
2,5	18
3,4	18
*3,5	19

ÖRNEK

Maks $P = 4x + 3y + 7z$

Kısıtlar

$x + 3y + 2z \geq 120$ (1)

$2x + y + 3z \geq 120$ (2)

$x \geq 0$ (3)

$y \geq 0$ (4)

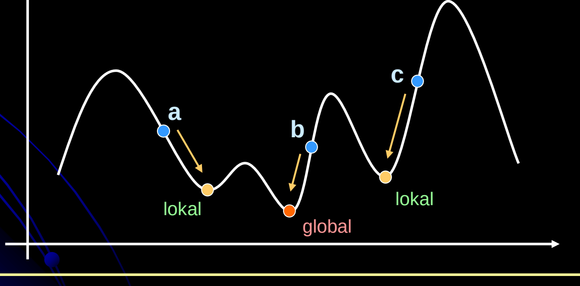
$z \geq 0$ (5)

KÖŞE	KOOR.	P = 4x + 3y + 7z
1 , 2 , 3 +*	0,120/7, 240/7	2040/7 = 291.4
1 , 2 , 4	-120 , 0 , 120	Uygun değil
1 , 2 , 5 +	48 , 24 , 0	264
1 , 3 , 4	0 , 0 , 60	Uygun değil
1 , 3 , 5 +	0 , 40 , 0	120
1 , 4 , 5	120 , 0 , 0	Uygun değil
2 , 3 , 4 +	6 , 0 , 40	280
2 , 3 , 5	0 , 120 , 0	Uygun değil
2 , 4 , 5 +	60 , 0 , 0	240
3 , 4 , 5 +	0 , 0 , 0	0

Optimal çözüm = P (0,120/7, 240/7) = 291.4

HILL CLIMBING

1. İterasyon tekniği kullanılır.
2. Arama alanında tek bir noktadan başlar.
3. Herbir iterasyonda mevcut çözüme komşu çözümler arasından bir tanesi seçilir.
4. Eğer seçilen nokta mevcut çözümden daha iyiyse yeni çözüm olarak alınır, değilse başka bir komşu çözüm aranır.
5. Algoritma daha iyi bir çözüm bulunamayınca veya daha önceden belirlenmiş iterasyon sayısına ulaşınca sonlandırılır.



HILL CLIMBING

- Lokal minimumu bulabilir.
  - Genellikle farklı başlama noktalarından başlanmak zorundadır.

- Başlama noktası seçimi şu şekilde yapılabilir;
  - Rastgele seçilebilir
  - Daha önceden belirlenmiş noktalara göre seçilebilir
  - Diğer bilgilere göre (Örn: Önceki çözümler, Uzman bilgileri, Ölçümler, vb.)

HILL CLIMBING

Örnek:
30 karakterlik bir binary karakter dizisi problemin çözüm uzayını oluşturmaktadır.

f(bd) = | 20 * bir_sayisi(bd) – 100 | fonksiyonunu maksimum yapan binary dizisini (bd) bulunuz.

bir_sayisi(bd) fonksiyonu bd dizisindeki birlerin sayısını vermektedir.

Örn: bd1 = (110111101111011101101111010101)
f(bd1) = | 10 * 22 – 100 | = 120

HILL CLIMBING

```
procedure HillClimbing
begin
 iterasyon ← 0
 repeat
  lokal ← FALSE
  rastgele bir bd dizisi seç
  bd dizisini değerlendir
  repeat
   bd dizisindeki bitlerin değerini değiştirip yeni 30 komşu çözüm üret
   Yeni komşu çözüm kümesinde f fonksiyonunu maks. yapanı seç
   if f(bd) < f(bd_yeni) then bd ← bd_yeni
    else lokal ← TRUE
  until lokal
  iterasyon ← iterasyon + 1
 until iterasyon = MAXITERASYON
end
```


HILL CLIMBING

Zayıf yanları:

- Genellikle lokal optimum olan çözümlerde sonlanmaktadır.
- Elde edilen optimum başlama noktasına bağlıdır.
- Genellikle istenen sonucu elde etmede hesaplama zamanı için bir üst limit yoktur.

Avantajları:

- Uygulamak çok kolaydır.
- Sadece problemin gösterimi, değerlendirme fonksiyonu ve komşu çözüm üretme fonksiyonu yeterlidir.

Zeki Optimizasyon Teknikleri

Haftalık Ödev:

Hill Climbing konusunda bir makale incelenerek elde edilen sonuçları içeren bir rapor hazırlanacaktır. İncelenen makalede Hill climbing kullanılmasının gerekçeleri, uygulamanın sonuçları değerlendirilecektir.

- İncelenen makale son 5 yılda yayınlanmış olacaktır.

- Makale Yurtdışında SCI'te taranan bir dergide yayınlanmış olacaktır.

- SCI'te tarandığını gösterir bilgi ödeve eklenecektir.

- Hazırlanan rapora makalenin tam metnide eklenecektir.

Zeki Optimizasyon Teknikleri

Gelecek Hafta

Karınca Algoritması

(Ant Algorithms)