Métodos Computacionais e Optimização
Computational Methods and Optimization
(MCO)

Bibliography:

- Bazaraa, Sherali and Shetty, Nonlinear Programming: Theory and Algorithms, 3rd edition, Wiley, 2006.

Lecture Notes MCO_1&2

OUTLINE:

- Optimisation Problem.
- Minimum and maximum of a function.
- Local and global minimum (maximum)
- Necessary and Sufficient conditions:

Unconstrained Problems.

Constrained Problems (KKT Conditions)

Problem

$$\underset{(x,y)}{Min} \ \{-[(x-4)^2 + (y-6)^2]\}$$

Subjcet to:

$$g_1 = x + y - 12 \leq 0$$

$$g_2 = x - 8 \leq 0$$

$$g_3 = -x \leq 0 \ (x \geq 0)$$

$$g_4 = -y \leq 0 \ (y \geq 0)$$

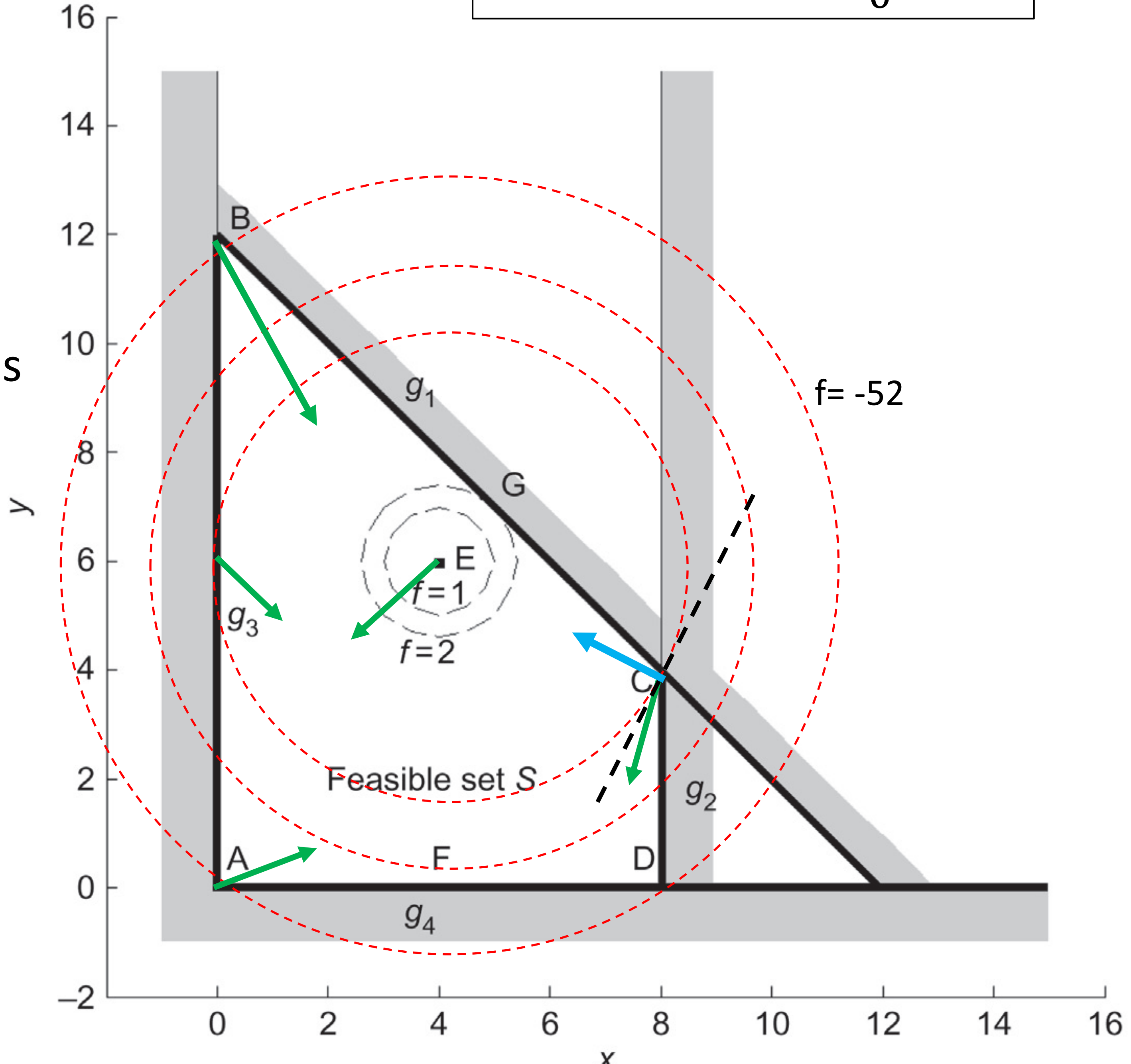
Admissible directions:

x* belongs to S

d is an admissible direction
if **x**= **x*** + ε**d** belongs to S for $0<\varepsilon<\delta$, for $\delta>0$

D is the set of admissible directions

Nec Cond D ∩ F_0 = ∅



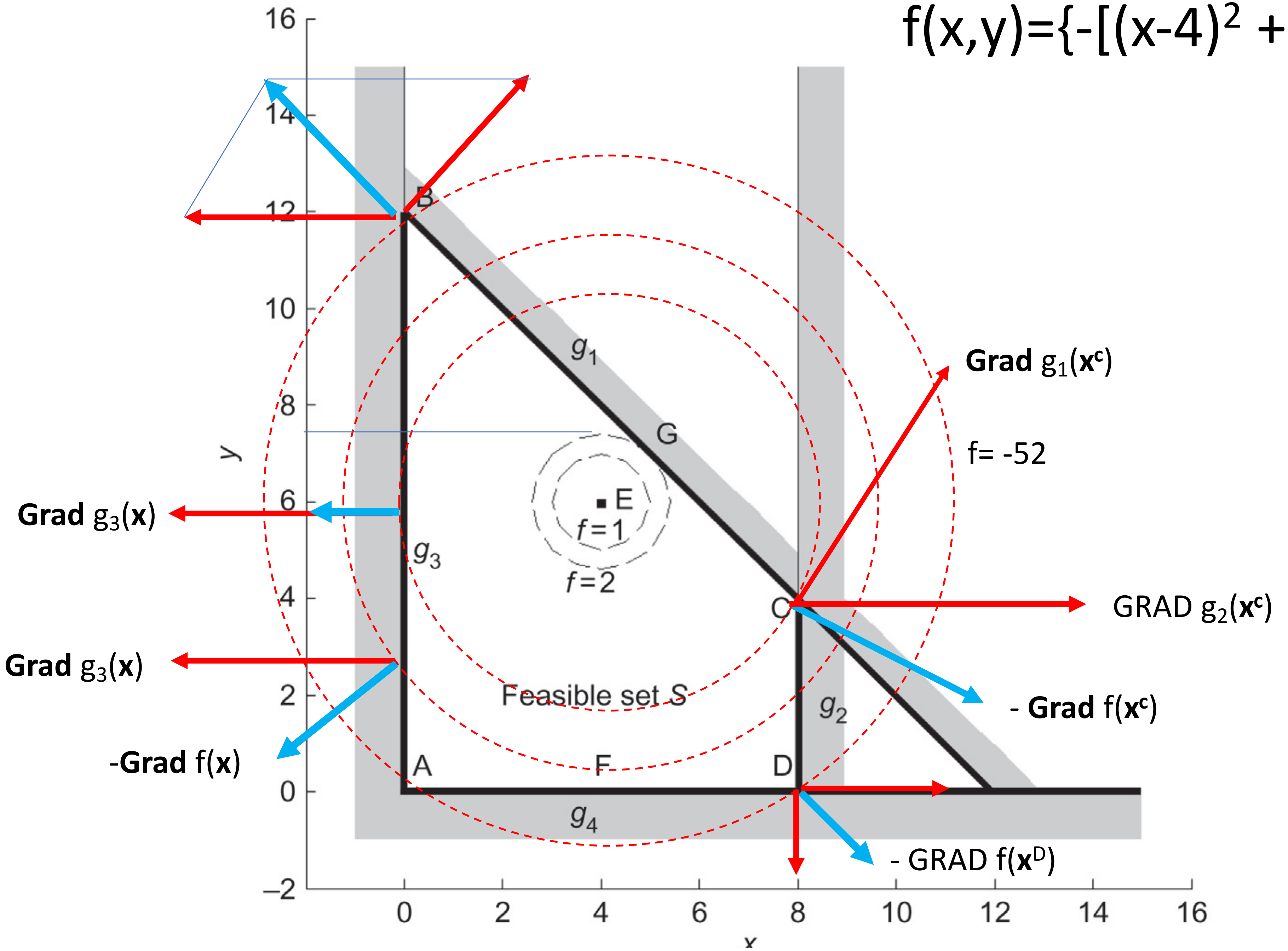
Descent directions (at **x***)

∇ f(x*). **d** < 0

F_0 is the set of descent directions

Question (true or false):
If x* is a minimum point then
∇ f(x*). **d** ≥ 0 ,
<u>for all admissible directions d</u> **?**

$$f(x,y)=\{-[(x-4)^2 + (y-6)^2]\}$$



Grad $g_1(\mathbf{x}^c)$
f= -52
GRAD $g_2(\mathbf{x}^c)$
- Grad f($\mathbf{x}^c$)
- GRAD f($\mathbf{x}^D$)
Grad $g_3(\mathbf{x})$
Grad $g_3(\mathbf{x})$
-Grad f($\mathbf{x}$)
B
G
E
f=1
f=2
C
D
A
F
g_1
g_2
g_3
g_4
Feasible set S
y
x
16
14
12
10
8
6
4
2
0
–2
0
2
4
6
8
10
12
14
16

5.1.1 Definition

Let S be a nonempty set in R^n, and let $\bar{\mathbf{x}} \in \mathrm{cl}\, S$. The *cone of tangents* of S at $\bar{\mathbf{x}}$, denoted by T, is the set of all directions $\mathbf{d}$ such that $\mathbf{d} = \lim_{k\to\infty} \lambda_k(\mathbf{x}_k - \bar{\mathbf{x}})$, where $\lambda_k > 0$, $\mathbf{x}_k \in S$ for each k, and $\mathbf{x}_k \to \bar{\mathbf{x}}$.

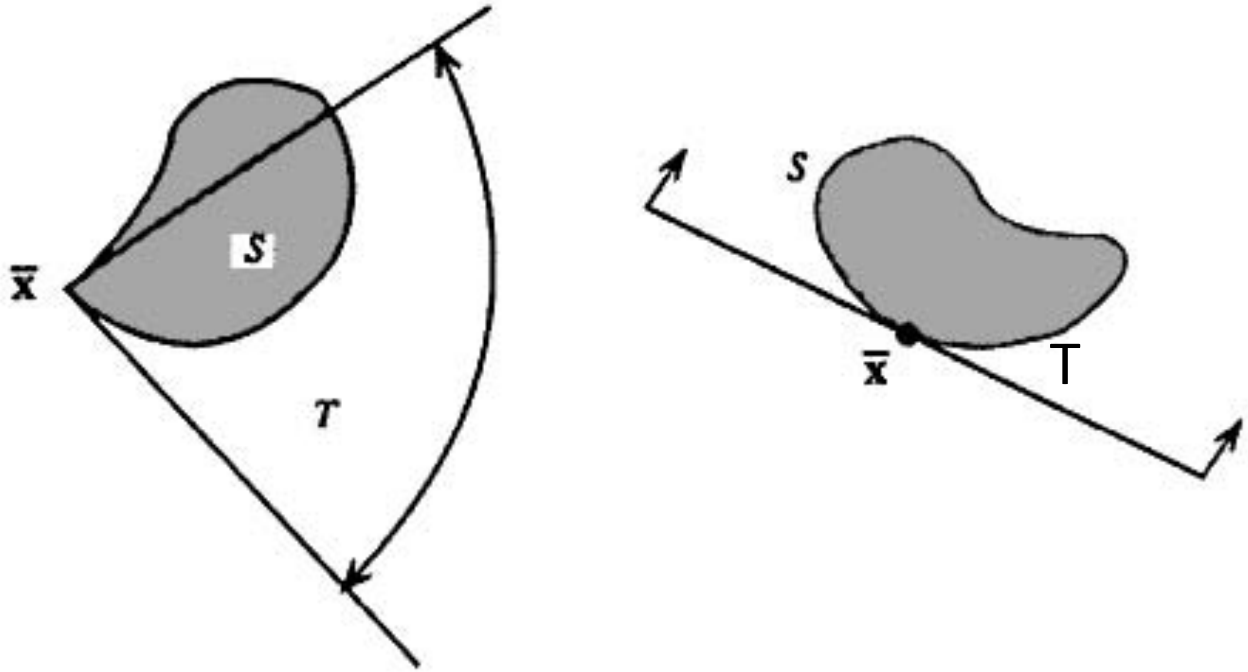


Figure 5.1 Cone of tangents.

5.1.2 Theorem

Let S be a nonempty set in R^n, and let $\bar{\mathbf{x}} \in S$. Furthermore, suppose that f: $R^n \to R$ is differentiable at $\bar{\mathbf{x}}$. If $\bar{\mathbf{x}}$ locally solves the problem to minimize $f(\mathbf{x})$ subject to $\mathbf{x} \in S$, $F_0 \cap T = \varnothing$, where $F_0 = \{\mathbf{d} : \nabla f(\bar{\mathbf{x}})^t \mathbf{d} < 0\}$ and T is the cone of tangents of S at $\bar{\mathbf{x}}$.

See proof in Bazaraa et al.

At the candidate point **x*** f(x) and g_i differentiable

$$I = \{i : \ g_i(\boldsymbol{x}^*) = 0\}$$

$$G' = \{\boldsymbol{\nabla} g_i(\boldsymbol{x}^*).\mathbf{d} \leq 0 \, , i \in I\}$$

Abadie Constraint Qualification (CQ)

G´ = T

So Necessary Condition is: G´∩ F_0 = ∅

Note : Linear independence of $\boldsymbol{\nabla} g_i(\boldsymbol{x}^*)$ for i ∈ I implies Abadie CQ (see Bazaraa et al. chapter 5 for details).

2.4.5 *Theorem* (*Farkas's Theorem*)

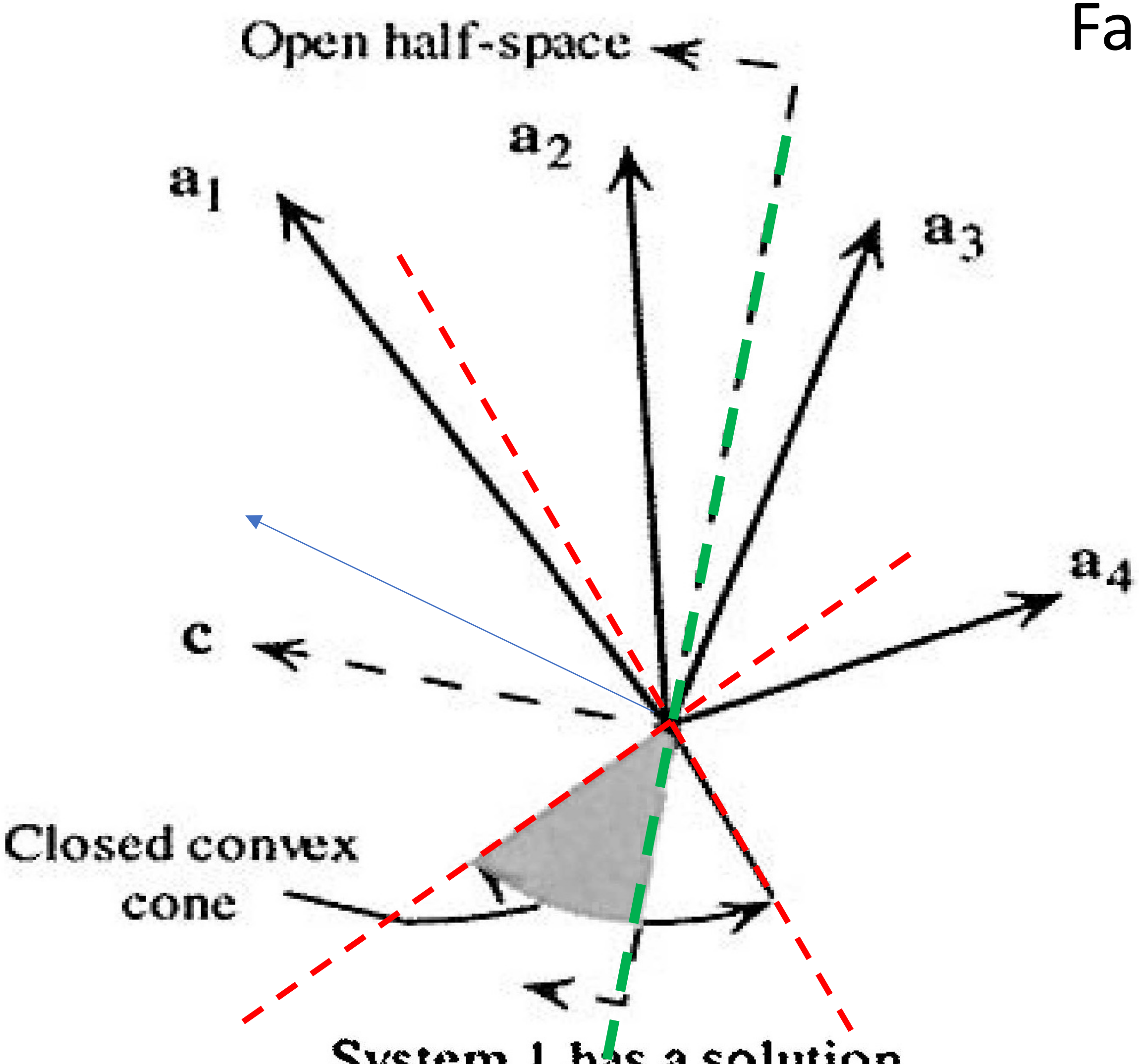
Let $\mathbf{A}$ be an $m \times n$ matrix and $\mathbf{c}$ be an n-vector. Then exactly one of the following two systems has a solution:

System 1: $\mathbf{Ax} \le \mathbf{0}$ and $\mathbf{c}^t\mathbf{x} > 0$ for some $\mathbf{x} \in R^n$.

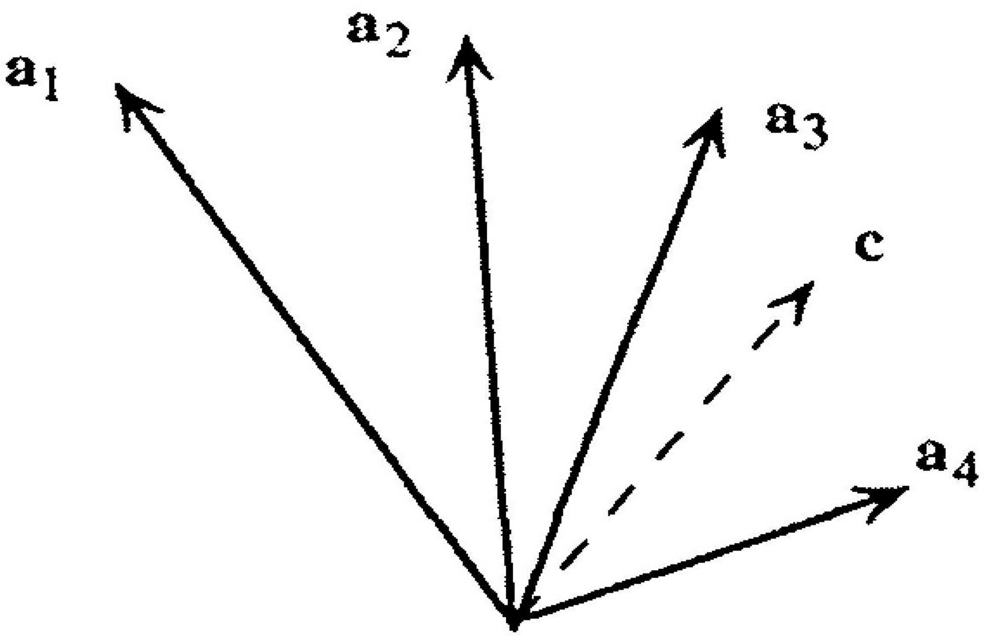
System 2: $\mathbf{A}^t\mathbf{y} = \mathbf{c}$ and $\mathbf{y} \ge \mathbf{0}$ for some $\mathbf{y} \in R^m$.

See Proof in Bazaraa et al.

Farkas Theorem (Geometric interpretation)



System 1 has a solution



System 2 has a solution

***System* 1**: $\mathbf{Ax} \le \mathbf{0},\ \mathbf{x} \ge \mathbf{0},\ \mathbf{c}^t\mathbf{x} > 0$ for some $\mathbf{x} \in R^n$.

***System* 2**: $\mathbf{A}^t\mathbf{y} = \mathbf{c}$ and $\mathbf{y} \ge \mathbf{0}$ for some $\mathbf{y} \in R^m$.

$Min\ f(\boldsymbol{x}),$

Subject to:

$g_i(\boldsymbol{x}) \leq 0, \quad \mathrm{i} = 1 \ldots . \mathrm{m}$

***c*= - ∇ *f*(*x*)**

I={i: $g_i(\boldsymbol{x}) = 0$}

(row) $A^{(i)} = \nabla g_i(\mathbf{x})$

$[A] = \begin{bmatrix} \nabla g_2(\mathbf{x}) \\ \\ \nabla g_p(\mathbf{x}) \end{bmatrix}$

Note: at point x* G´ = T (Abadie CQ)

System 1 has solution **x*** ⟹ $[A(\mathbf{x}^*)]\{d\} = \{\leq 0\}$ and $\nabla f(\mathbf{x}^*).\ \mathbf{d} < \mathbf{0}$

System 1 no solution ⟹ $-\nabla f(\mathbf{x}^*) = \sum_{i \in I} y_i \nabla g_i(\mathbf{x}^*)$

$y_i \geq 0$

$$Min\ f(\mathbf{x}),$$

Subject to:

$$g_i\,(\boldsymbol{x}) \leq 0, \quad \mathrm{i} = 1 \dots . \mathrm{m}$$

KKT Conditions:

$$\nabla f(\mathbf{x^*}) + \sum_{i \in I} \lambda_i \nabla g_i(\mathbf{x^*}) = 0$$

$$\lambda_i \geq 0, \qquad g_i\,(\boldsymbol{x}^*) = 0, i \in I$$

KKT Conditions (alternative form):

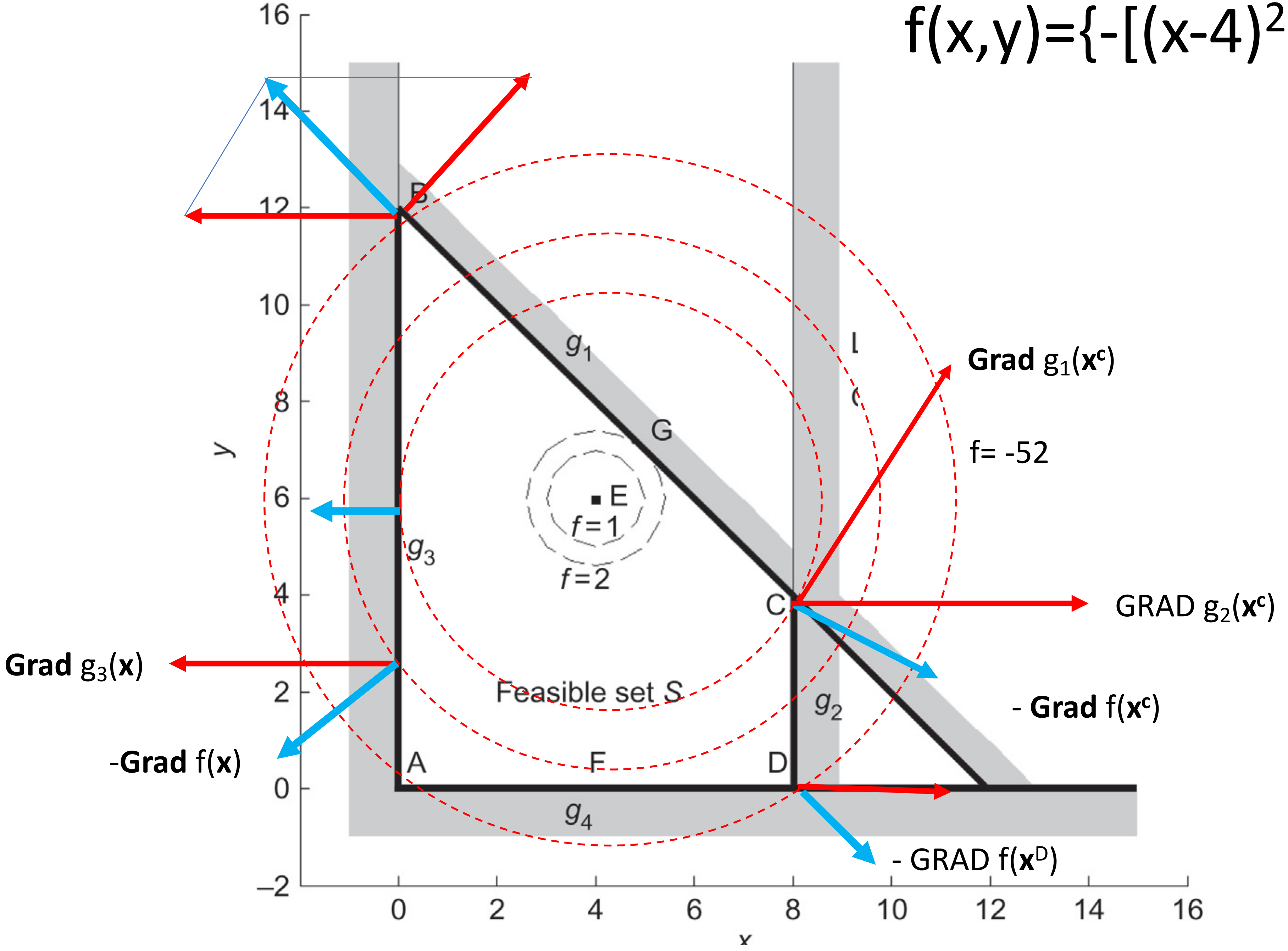
$$\boxed{\lambda_i - \text{Lagrange Multipliers}}$$

$$\nabla f(\mathbf{x^*}) + \sum_{i = 1, m} \lambda_i \nabla g_i(\mathbf{x^*}) = 0$$

$$\lambda_i \geq 0, \qquad \lambda_i\, g_i\,(\mathbf{x^*}) = 0, \quad \mathrm{i} = 1 \dots . \mathrm{m}$$

$$g_i\,(\mathbf{x^*}) \leq 0, \quad \mathrm{i} = 1 \dots . \mathrm{m}$$

$$f(x,y)=\{-[(x-4)^2 + (y-6)^2]\}$$


Grad g1(xc)
f= -52
GRAD g2(xc)
- Grad f(xc)
- GRAD f(xD)
Grad g3(x)
-Grad f(x)
B
G
E
f=1
f=2
C
D
A
F
g1
g2
g3
g4
Feasible set S
y
x
16
14
12
10
8
6
4
2
0
–2
0
2
4
6
8
10
12
14
16

Corollary 3

Let **A** be an $m \times n$ matrix, **B** be an $\ell \times n$ matrix, and **c** be an n-vector. Then exactly one of the following two systems has a solution:

System 1: $\mathbf{Ax} \le \mathbf{0}$, $\mathbf{Bx} = \mathbf{0}$, $\mathbf{c}^t\mathbf{x} > 0$ for some $\mathbf{x} \in R^n$.

System 2: $\mathbf{A}^t\mathbf{y} + \mathbf{B}^t\mathbf{z} = \mathbf{c}$, $\mathbf{y} \ge \mathbf{0}$ for some $\mathbf{y} \in R^m$ and $\mathbf{z} \in R^\ell$.

Proof

The result follows by writing $\mathbf{z} = \mathbf{z}_1 - \mathbf{z}_2$, where $\mathbf{z}_1 \ge \mathbf{0}$ and $\mathbf{z}_2 \ge \mathbf{0}$ in System 2 and, accordingly, replacing $\mathbf{A}^t$ in the theorem by $[\mathbf{A}^t, \mathbf{B}^t, -\mathbf{B}^t]$.

From Bazaraa et al.

At the candidate point **x*** f(x) and h_j, g_i are differentiable

$$I = \{i : \; g_i(\boldsymbol{x}^*) = 0\}$$

$$G' = \{\boldsymbol{\nabla} g_i(\boldsymbol{x}^*).\mathbf{d} \leq 0 \, , i \in I\}$$

$$H_0 \;\; = \left\{\boldsymbol{\nabla} h_j(\boldsymbol{x}^*).\mathbf{d} = 0 \, , j = 1, \ldots\ldots, p\right\}$$

Abadie Constraint Qualification (CQ)

$G' \cap H_0 \; = T$

So Necessary Condition is: $G' \cap H_0 \;\; \cap F_0 \;\; = \emptyset$

Note : Linear independence of $\boldsymbol{\nabla} g_i(\boldsymbol{x}^*)$ for i ∈ I and $\boldsymbol{\nabla} h_j(\boldsymbol{x}^*)$ implies Abadie CQ (see Bazaraa et al. chapter 5 for details)

$$Min\ f(\mathbf{x}),$$

Subject to:

$g_i\,(\boldsymbol{x}) \leq 0, \quad \mathrm{i} = 1 \ldots . \mathrm{m}$

$h_j\,(\boldsymbol{x}) = 0, \quad \mathrm{j} = 1 \ldots . \mathrm{p}$

KKT Conditions:

$$\nabla f(\mathbf{x}^*) + \sum_{i \in I} \lambda_i \nabla g_i(\mathbf{x}^*) + \sum \mu_j \nabla h_j(\mathbf{x}^*) = 0$$

$\lambda_i \geq 0, \quad g_i\,(\boldsymbol{x}^*) = 0, \mathrm{i} \in I$

$h_j\,(\boldsymbol{x}^*) = 0, \quad \mathrm{j} = 1 \ldots . \mathrm{p}$

KKT Conditions (alternative form):

$$\nabla f(\mathbf{x}^*) + \sum_{i = 1, m} \lambda_i \nabla g_i(\mathbf{x}^*) + \sum_{j = 1, p} \mu_j \nabla h_j(\mathbf{x}^*) = 0$$

$\lambda_i \geq 0, \quad \lambda_i g_i\,(\mathbf{x}^*) = 0, \quad \mathrm{i} = 1 \ldots . \mathrm{m} \qquad h_j\,(\boldsymbol{x}) = 0, \quad \mathrm{j} = 1 \ldots . \mathrm{p}$

$g_i\,(\mathbf{x}^*) \leq 0, \quad \mathrm{i} = 1 \ldots . \mathrm{m}$

Lecture Notes MCO_3

Outline:

Unconstrained minimisation

Basic Minimization Algorithm.

Line search

Algorithms (implementation procedures taken from Bazaraa)*

* Bazaraa, Sherali and Shetty, Nonlinear Programming: Theory and Algorithms, 3rd edition, Wiley, 2006.

Unconstrained Minimization

Find $\mathbf{x}^*$ in R^n

that solves (local):

Min $f(\mathbf{x})$

Necessary Condition 1st order:

$\nabla f(\mathbf{x}^*) = 0.$ (1)

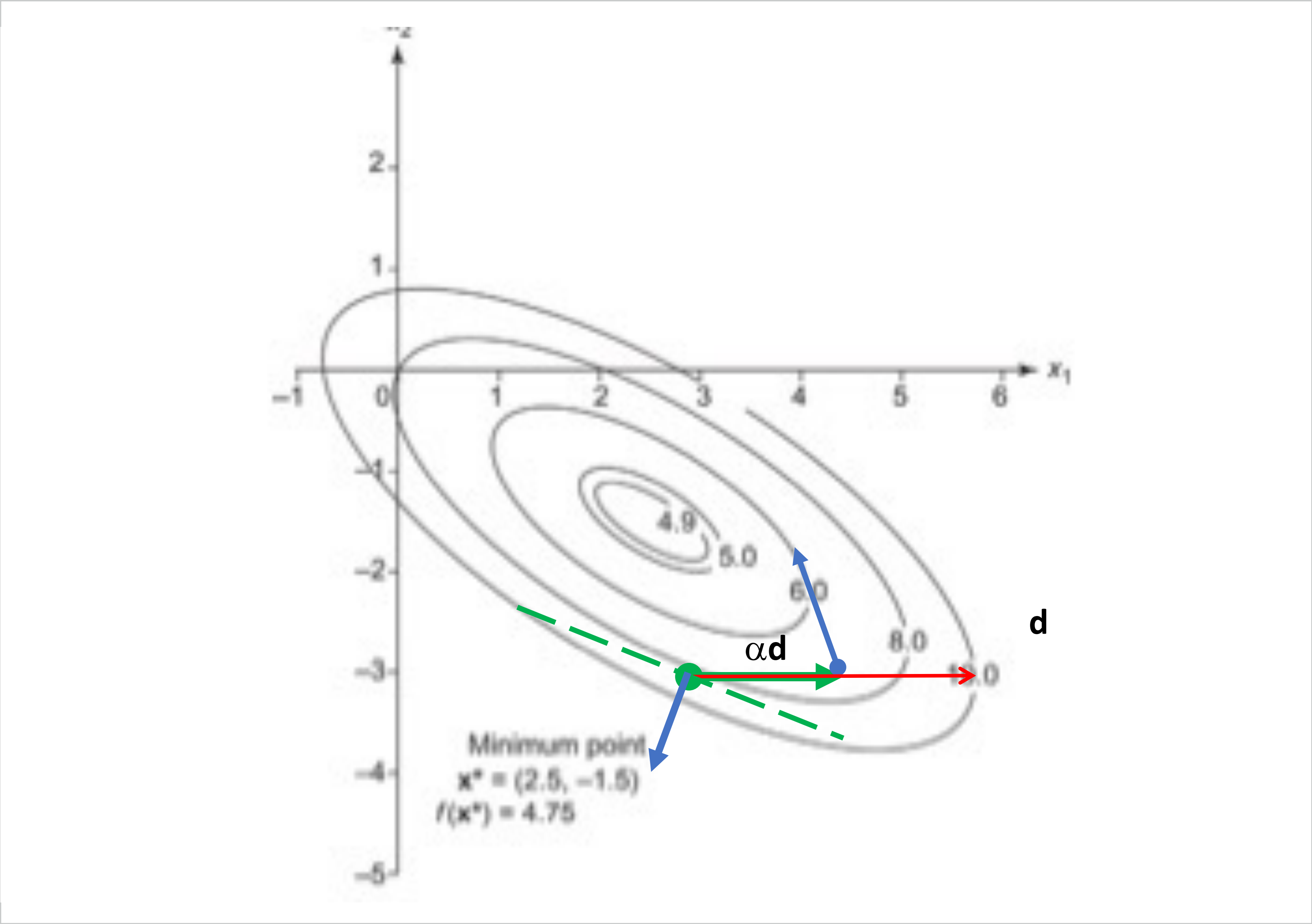
At $\mathbf{x}^*$ satisfying (1):

Necessary Condition 2nd order:

$\mathbf{H}(\mathbf{x}^*) \geq 0$ (semi-positiveve definite)

Sufficient Condition:

$\mathbf{H}(\mathbf{x}^*) > 0$ (positive definite)



Basic Mathematical Programming Algorithm

Basic Mathematical Programming Algorithm

Find x^* in R^n
Min f(x)

1 - Actual Point $\mathbf{x}^k$

2 - Define direction $\mathbf{d}^k$ of descent, $\nabla f(\mathbf{x}^k) . \mathbf{d}^k < 0$

3 - Solve min $(f(\mathbf{x}^k + \alpha \mathbf{d}^k)$ to find α^* "> 0 if $\mathbf{d}^k$ descent" (Line search)

4 - New point $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^* \mathbf{d}^k$

5 - Repeat until convergence (or limit of iterations)

Line Search (step length)

$\phi(\alpha) = f(\mathbf{x}^k + \alpha\, \mathbf{d}^k)$

$$\min_{\alpha > 0} \phi(\alpha) \xrightarrow{\text{Solve}} \boxed{\nabla f(\mathbf{x}^k + \alpha\, \mathbf{d}^k).\, \mathbf{d}^k = 0}$$

“Exact” line search algorithms:

- Equal Interval search
- Dichotomous search
- Golden section
- Bisecting Method

……….

Non-exact line search: Armijo Rule (see Bazaraa et al.)

Summary of the Dichotomous Search Method

Following is a summary of the dichotomous method for minimizing a strictly quasiconvex function θ over the interval $[a_1, b_1]$.

Initialization Step Choose the distinguishability constant, $2\varepsilon > 0$, and the allowable final length of uncertainty, $\ell > 0$. Let $[a_1, b_1]$ be the initial interval of uncertainty, let $k = 1$, and go to the Main Step.

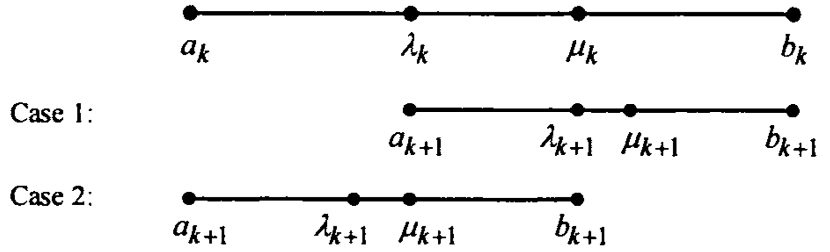
Main Step

1. If $b_k - a_k < \ell$, stop; the minimum point lies in the interval $[a_k, b_k]$. Otherwise, consider λ_k and μ_k defined below, and go to Step 2.

$$\lambda_k = \frac{a_k + b_k}{2} - \varepsilon, \qquad \mu_k = \frac{a_k + b_k}{2} + \varepsilon.$$

2. If $\theta(\lambda_k) < \theta(\mu_k)$, let $a_{k+1} = a_k$ and $b_{k+1} = \mu_k$. Otherwise, let $a_{k+1} = \lambda_k$ and $b_{k+1} = b_k$. Replace k by $k + 1$, and go to Step 1.

From: Bazaraa, Sherali and Shetty, Nonlinear Programming: Theory and Algorithms, 3rd edition, Wiley, 2006.



Summary of the Golden Section Method

Following is a summary of the golden section method for minimizing a strictly quasiconvex function over the interval $[a_1, b_1]$.

Initialization Step Choose an allowable final length of uncertainty $\ell > 0$. Let $[a_1, b_1]$ be the initial interval of uncertainty, and let $\lambda_1 = a_1 + (1-\alpha)(b_1 - a_1)$ and $\mu_1 = a_1 + \alpha(b_1 - a_1)$, where $\alpha = 0.618$. Evaluate $\theta(\lambda_1)$ and $\theta(\mu_1)$, let $k = 1$, and go to the Main Step.

Main Step

1. If $b_k - a_k < \ell$, stop; the optimal solution lies in the interval $[a_k, b_k]$. Otherwise, if $\theta(\lambda_k) > \theta(\mu_k)$, go to Step 2; and if $\theta(\lambda_k) \leq \theta(\mu_k)$, go to Step 3.
2. Let $a_{k+1} = \lambda_k$ and $b_{k+1} = b_k$. Furthermore, let $\lambda_{k+1} = \mu_k$, and let $\mu_{k+1} = a_{k+1} + \alpha(b_{k+1} - a_{k+1})$. Evaluate $\theta(\mu_{k+1})$ and go to Step 4.
3. Let $a_{k+1} = a_k$ and $b_{k+1} = \mu_k$. Furthermore, let $\mu_{k+1} = \lambda_k$, and let $\lambda_{k+1} = a_{k+1} + (1-\alpha)(b_{k+1} - a_{k+1})$. Evaluate $\theta(\lambda_{k+1})$ and go to Step 4.
4. Replace k by $k + 1$ and go to Step 1.

From: Bazaraa, Sherali and Shetty, Nonlinear Programming: Theory and Algorithms, 3rd edition, Wiley, 2006.

Armijo Rule (from Bazaraa et al.)

$$\hat{\theta}(\lambda) = \theta(0) + \lambda \varepsilon \theta'(0) \qquad \text{for } \lambda \geq 0.$$

$\bar{\lambda}$ acceptable if:

$$\theta(\bar{\lambda}) \leq \hat{\theta}(\bar{\lambda}).$$

$$\theta(\alpha\bar{\lambda}) > \hat{\theta}(\alpha\bar{\lambda})$$

Typical Values:

$\alpha=[2, 10]$ and $\varepsilon=0.2$

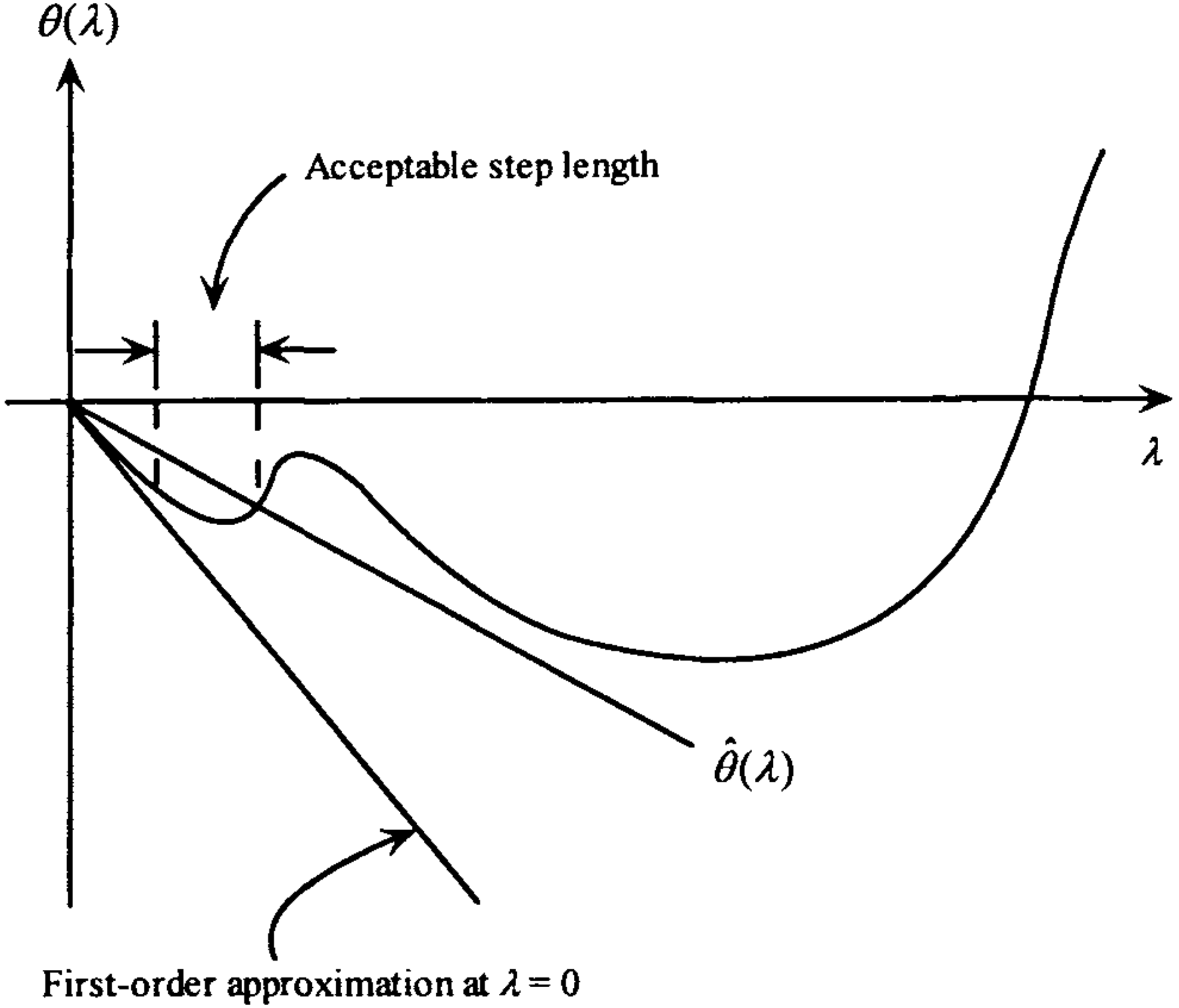


Figure 8.6 Armijo's rule.

Methods Without Gradient Information:

- Cyclic coordinate Method
- Hooke – Jeeves
- Rosenbrock

Other methods not discussed here (e.g. Nelder-Mead Method….etc.)

Examples shown in the graphics are for:

$\min_{\mathbf{x}} (x_1-2)^4 + (x_1-2x_2)^2$

From Bazaraa et al.

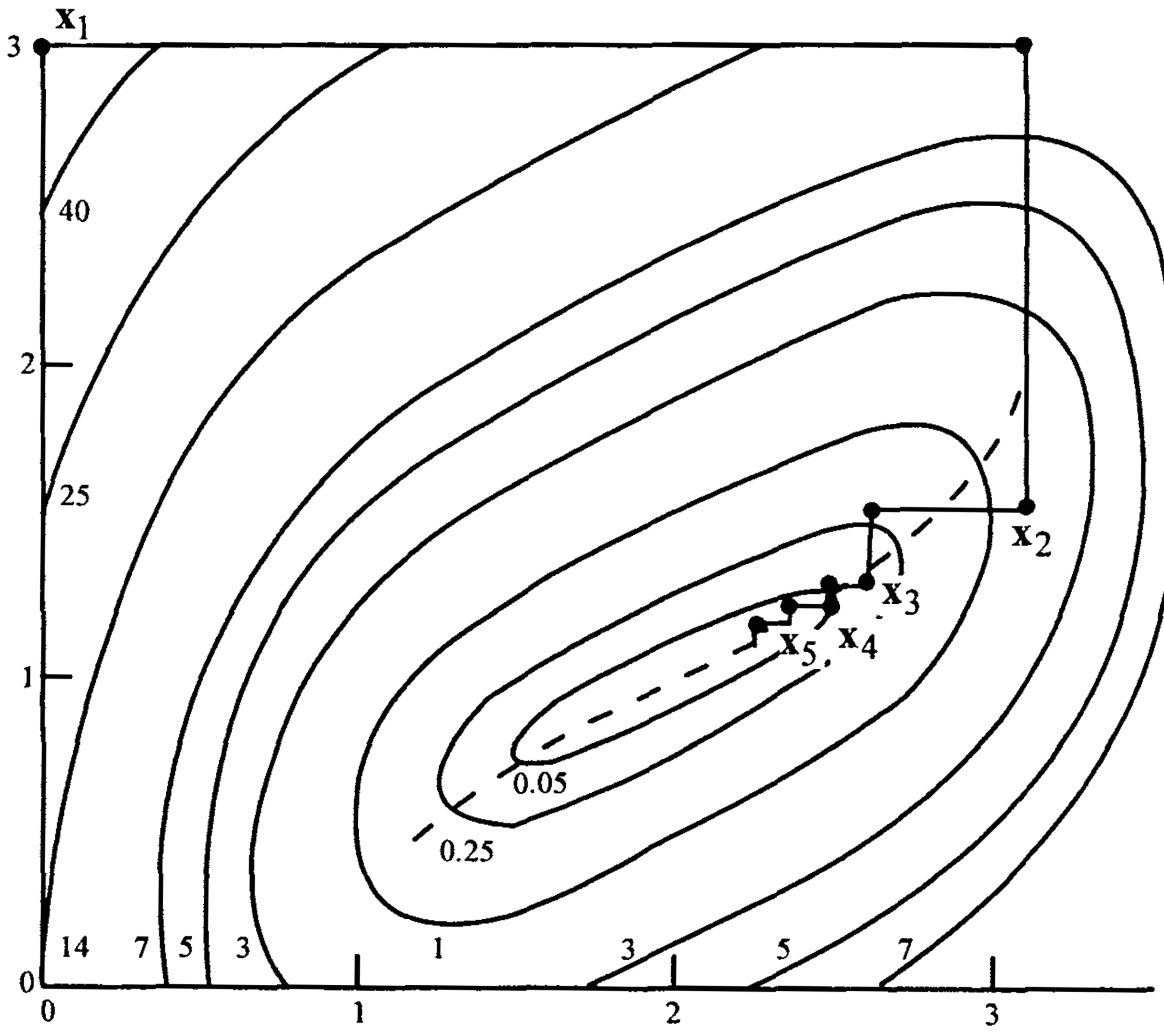
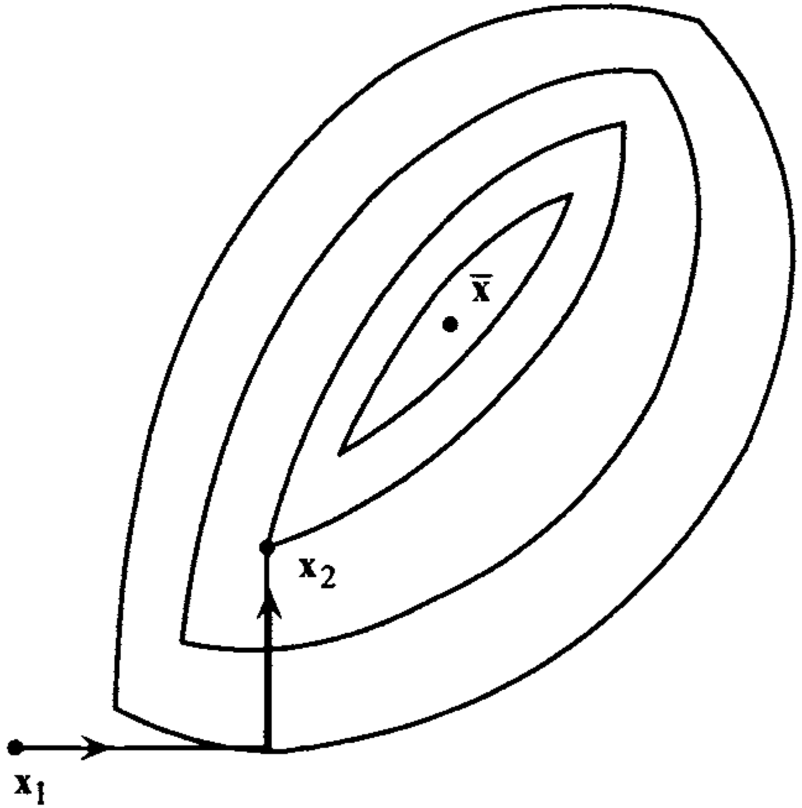


Figure 8.7 Cyclic coordinate method.

• HOOKE-JEEVES



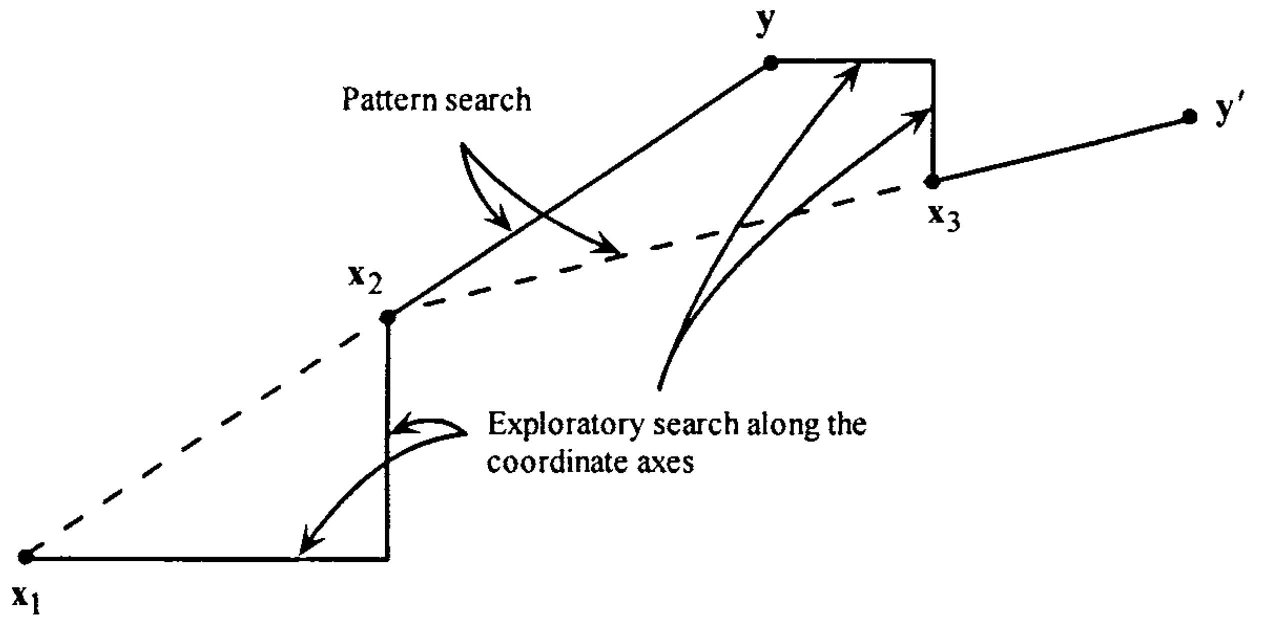


Figure 8.9 Method of Hooke and Jeeves.

From Bazaraa et al.

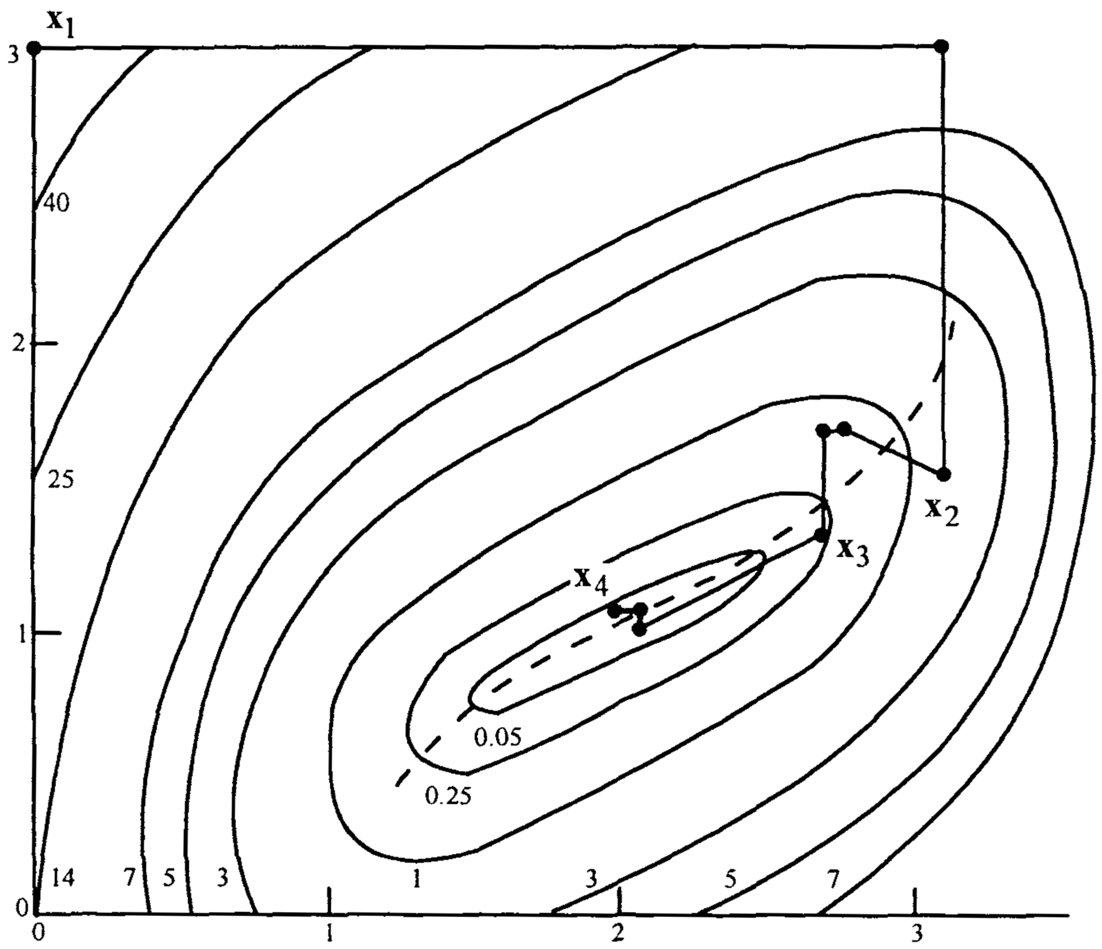
Hooke Jeeves(from Bazaraa et al.)

Initialization Step Choose a scalar $\varepsilon > 0$ to be used in terminating the algorithm. Choose a starting point $\mathbf{x}_1$, let $\mathbf{y}_1 = \mathbf{x}_1$, let $k = j = 1$, and go to the Main Step.

Main Step

1. Let λ_j be an optimal solution to the problem to minimize $f(\mathbf{y}_j + \lambda \mathbf{d}_j)$ subject to $\lambda \in R$, and let $\mathbf{y}_{j+1} = \mathbf{y}_j + \lambda_j \mathbf{d}_j$. If $j < n$, replace j by $j + 1$, and repeat Step 1. Otherwise, if $j = n$, let $\mathbf{x}_{k+1} = \mathbf{y}_{n+1}$. If $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < \varepsilon$, stop; otherwise, go to Step 2.

2. Let $\mathbf{d} = \mathbf{x}_{k+1} - \mathbf{x}_k$, and let $\hat{\lambda}$ be an optimal solution to the problem to minimize $f(\mathbf{x}_{k+1} + \lambda \mathbf{d})$ subject to $\lambda \in R$. Let $\mathbf{y}_1 = \mathbf{x}_{k+1} + \hat{\lambda}\mathbf{d}$, let $j = 1$, replace k by $k + 1$, and go to Step 1.



From: Bazaraa, Sherali and Shetty, Nonlinear Programming: Theory and Algorithms, 3rd edition, Wiley, 2006.

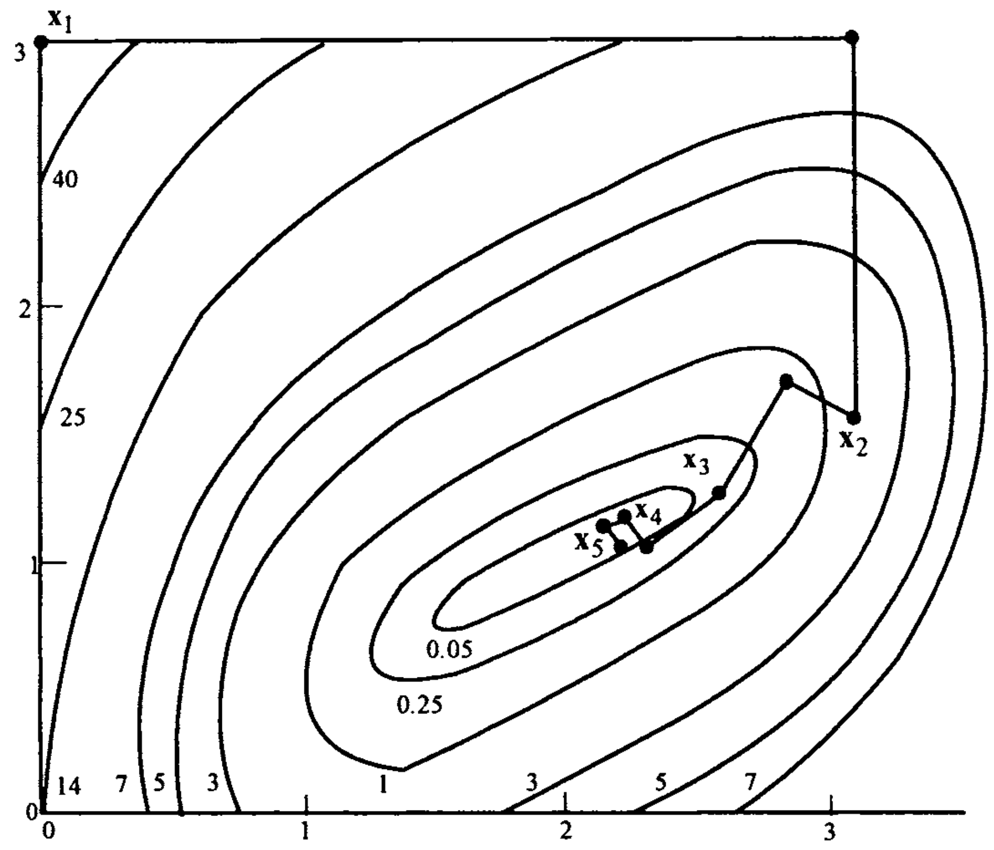
Rosenbrock (from Bazaraa et al.)

Initialization Step Let $\varepsilon > 0$ be the termination scalar. Choose $\mathbf{d}_1,\ldots,\mathbf{d}_n$ as the coordinate directions. Choose a starting point $\mathbf{x}_1$, let $\mathbf{y}_1 = \mathbf{x}_1$, $k = j = 1$, and go to the Main Step.

Main Step

1. Let λ_j be an optimal solution to the problem to minimize $f(\mathbf{y}_j + \lambda\mathbf{d}_j)$ subject to $\lambda \in R$, and let $\mathbf{y}_{j+1} = \mathbf{y}_j + \lambda_j\mathbf{d}_j$. If $j < n$, replace j by $j + 1$, and repeat Step 1. Otherwise, go to Step 2.
2. Let $\mathbf{x}_{k+1} = \mathbf{y}_{n+1}$. If $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < \varepsilon$, then stop; otherwise, let $\mathbf{y}_1 = \mathbf{x}_{k+1}$, replace k by $k + 1$, let $j = 1$, and go to Step 3.
3. Form a new set of linearly independent orthogonal search directions by (8.9). Denote these new directions by $\mathbf{d}_1,\ldots,\mathbf{d}_n$ and go to Step 1.

New directions created using a Gram-Schmidt Procedure
(see Bazaraa et al.)



The new collection of directions $\bar{\mathbf{d}}_1, \ldots, \bar{\mathbf{d}}_n$ is formed by the *Gram-Schmidt procedure*, or *orthogonalization procedure*, as follows:

$$\mathbf{a}_j = \begin{cases} \mathbf{d}_j & \text{if } \lambda_j = 0 \\ \sum_{i=j}^{n} \lambda_i \mathbf{d}_i & \text{if } \lambda_j \neq 0 \end{cases}$$

$$\mathbf{b}_j = \begin{cases} \mathbf{a}_j, & j = 1 \\ \mathbf{a}_j - \sum_{i=1}^{j-1} (\mathbf{a}_j^t \bar{\mathbf{d}}_i) \bar{\mathbf{d}}_i, & j \geq 2 \end{cases} \tag{8.9}$$

$$\bar{\mathbf{d}}_j = \frac{\mathbf{b}_j}{\|\mathbf{b}_j\|}.$$

From Bazaraa et al.

Methods With Gradient Information

- Steepest Descent
- Newton (see Arora)
- Conjugate Gradient (FR)
- DFP - Davidon-Fletcher-Powell

Important Note:
The following three algorithms are taken from the book : Nonlinear Programming, Bazaraa, Sherali and Shetty, 3nd edition, Wiley
They are for exclusive and private use of IST MCO course students, 2023/24 academic year.

Summary of the Steepest Descent Algorithm

Given a point $\mathbf{x}$, the steepest descent algorithm proceeds by performing a line search along the direction $-\nabla f(\mathbf{x})/\|\nabla f(\mathbf{x})\|$ or, equivalently, along the direction $-\nabla f(\mathbf{x})$. A summary of the method is given below.

Initialization Step Let $\varepsilon > 0$ be the termination scalar. Choose a starting point $\mathbf{x}_1$, let $k = 1$, and go to the Main Step.

Main Step

If $\|\nabla f(\mathbf{x}_k)\| < \varepsilon$, stop; otherwise, let $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$, and let λ_k be an optimal solution to the problem to minimize $f(\mathbf{x}_k + \lambda \mathbf{d}_k)$ subject to $\lambda \geq 0$. Let $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$, replace k by $k+1$, and repeat the Main Step.

Conjugate Gradient –Fletcher - Reeves

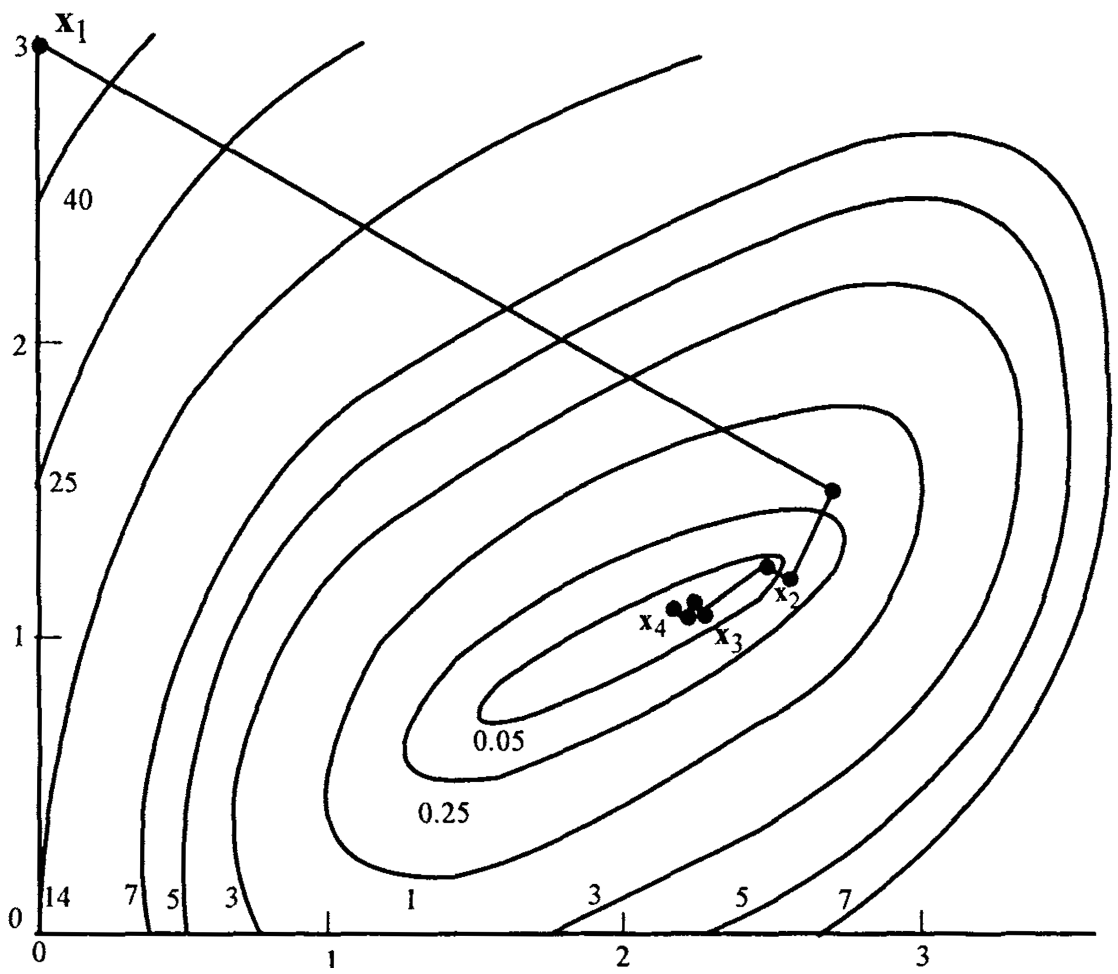
Initialization Step Choose a termination scalar $\varepsilon > 0$ and an initial point $\mathbf{x}_1$. Let $\mathbf{y}_1 = \mathbf{x}_1$, $\mathbf{d}_1 = -\nabla f(\mathbf{y}_j)$, $k = j = 1$, and go to the Main Step.

Main Step

1. If $\|\nabla f(\mathbf{y}_j)\| < \varepsilon$, stop. Otherwise, let λ_j be an optimal solution to the problem to minimize $f(\mathbf{y}_j + \lambda \mathbf{d}_j)$ subject to $\lambda \geq 0$, and let $\mathbf{y}_{j+1} = \mathbf{y}_j + \lambda_j \mathbf{d}_j$. If $j < n$, go to Step 2; otherwise, go to Step 3.
2. Let $\mathbf{d}_{j+1} = -\nabla f(\mathbf{y}_{j+1}) + \alpha_j \mathbf{d}_j$, where

$$\alpha_j = \frac{\|\nabla f(\mathbf{y}_{j+1})\|^2}{\|\nabla f(\mathbf{y}_j)\|^2}.$$

   Replace j by $j + 1$, and go to Step 1.
3. Let $\mathbf{y}_1 = \mathbf{x}_{k+1} = \mathbf{y}_{n+1}$, and let $\mathbf{d}_1 = -\nabla f(\mathbf{y}_1)$. Let $j = 1$, replace k by $k + 1$, and go to Step 1.



See Bazaraa et al.

Quasi – Newton Methods:

DFP

BFGS*

*See Bazaraa et al. for BFGS family of Methods

DFP

Initialization Step Let $\varepsilon > 0$ be a termination tolerance. Choose an initial point $\mathbf{x}_1$ and an initial symmetric positive definite matrix $\mathbf{D}_1$. Let $\mathbf{y}_1 = \mathbf{x}_1$, let $k = j = 1$, and go to the Main Step.

Main Step

1. If $\|\nabla f(\mathbf{y}_j)\| < \varepsilon$, stop; otherwise, let $\mathbf{d}_j = -\mathbf{D}_j \nabla f(\mathbf{y}_j)$ and let λ_j be an optimal solution to the problem to minimize $f(\mathbf{y}_j + \lambda \mathbf{d}_j)$ subject to $\lambda \geq 0$. Let $\mathbf{y}_{j+1} = \mathbf{y}_j + \lambda_j \mathbf{d}_j$. If $j < n$, go to Step 2. If $j = n$, let $\mathbf{y}_1 = \mathbf{x}_{k+1} = \mathbf{y}_{n+1}$, replace k by $k + 1$, let $j = 1$, and repeat Step 1.
2. Construct $\mathbf{D}_{j+1}$ as follows:

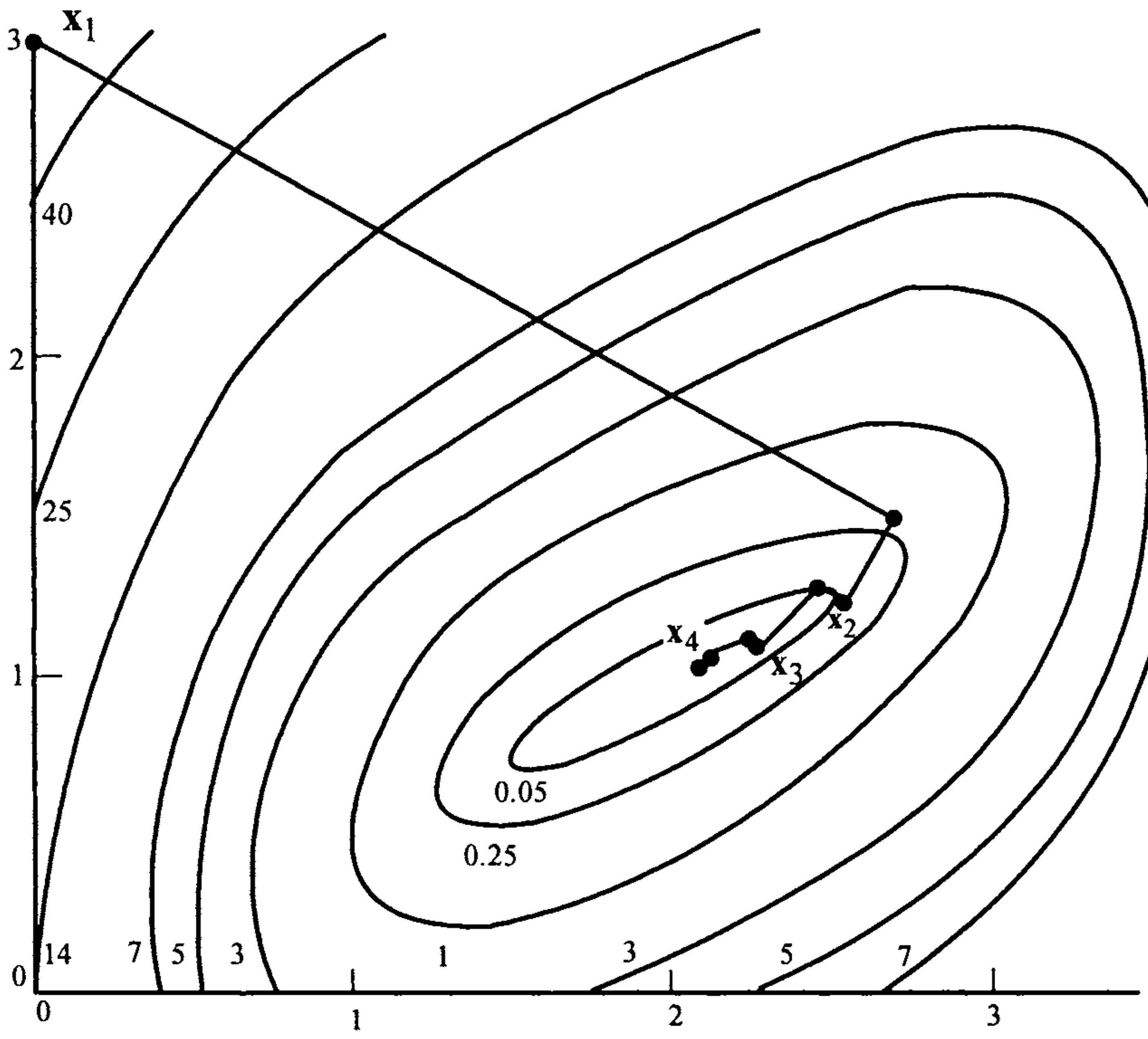
$$\mathbf{D}_{j+1} = \mathbf{D}_j + \frac{\mathbf{p}_j \mathbf{p}_j^t}{\mathbf{p}_j^t \mathbf{q}_j} - \frac{\mathbf{D}_j \mathbf{q}_j \mathbf{q}_j^t \mathbf{D}_j}{\mathbf{q}_j^t \mathbf{D}_j \mathbf{q}_j}, \tag{8.30}$$

where

$$\mathbf{p}_j \equiv \lambda_j \mathbf{d}_j = \mathbf{y}_{j+1} - \mathbf{y}_j \tag{8.31}$$

$$\mathbf{q}_j \equiv \nabla f(\mathbf{y}_{j+1}) - \nabla f(\mathbf{y}_j). \tag{8.32}$$

Replace j by $j + 1$, and go to Step 1.



From Bazaraa et al.

Lecture Notes MCO 4

Outline:

Constrained Minimisation: Algorithms

- (exterior) Penalty Method
- Augmented Lagrangian Method

(*) implementation procedures taken from Bazaraa et al.)*

Penalty Method (P1)

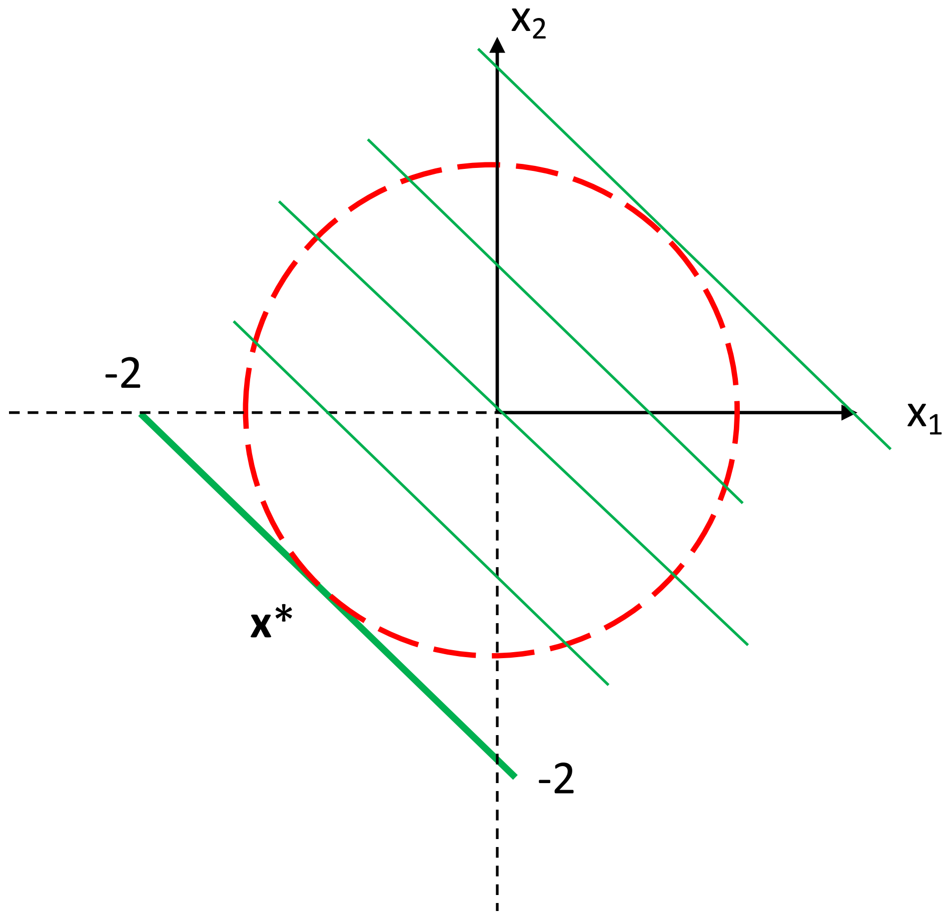
Original Problem (P1)

$$\min\ x_1 + x_2 \qquad \text{subject to } x_1^2 + x_2^2 - 2 = 0,$$

Solution: (-1,-1)

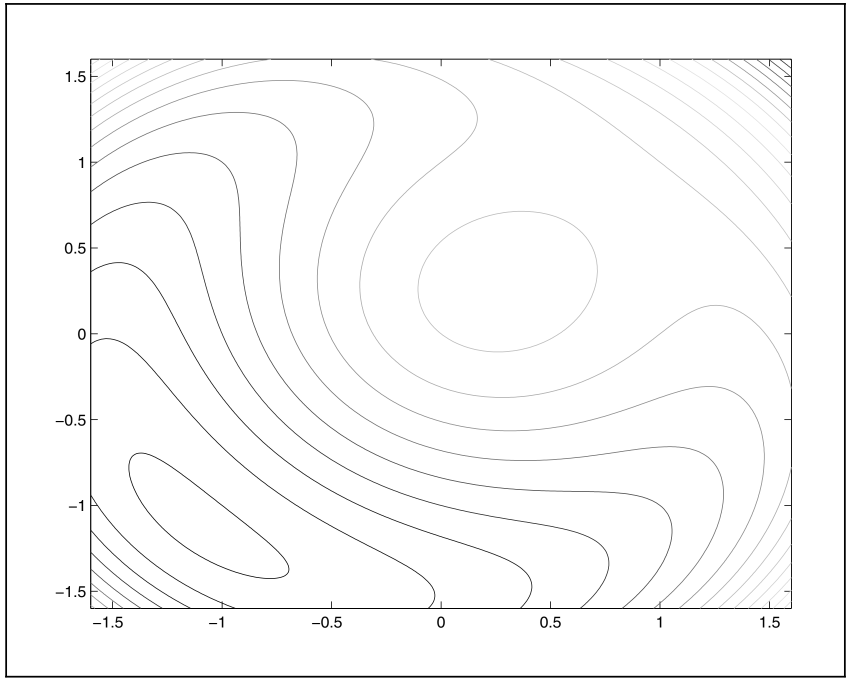
Penalized Function (P2)

$$\Phi\,(x;\ \mathrm{r}\,) = x_1 + x_2 + \frac{\mathrm{r}}{2}\left(x_1^2 + x_2^2 - 2\right)^2.$$



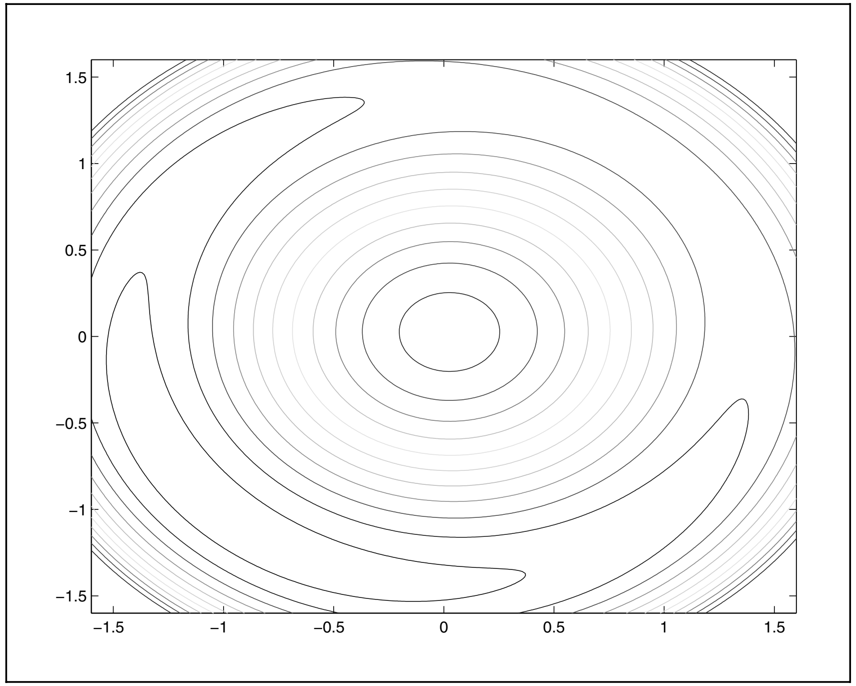
x2
x1
-2
-2
x*

r=1



1.5
1
0.5
0
−0.5
−1
−1.5
−1.5
−1
−0.5
0
0.5
1
1.5

r=10



1.5
1
0.5
0
−0.5
−1
−1.5
−1.5
−1
−0.5
0
0.5
1
1.5

Penalty Method

min $f = f(\mathbf{x})$

$h_i(\mathbf{x}) = 0; \quad i = 1 \text{ to } p$

$g_i(\mathbf{x}) \le 0; \quad i = 1 \text{ to } m$

Penalty Formulation

$$\Phi(\mathbf{x}, \mathbf{r}) = f(\mathbf{x}) + P\big(\mathbf{h}(\mathbf{x}), \mathbf{g}(\mathbf{x}), \mathbf{r}\big)$$

$$P\big(\mathbf{h}(\mathbf{x}), \mathbf{g}(\mathbf{x}), r\big) = r\left\{\sum_{i=1}^{p}[h_i(\mathbf{x})]^2 + \sum_{i=1}^{m}[g_i^+(\mathbf{x})]^2\right\}; \quad g_i^+(\mathbf{x}) = \max\big(0, g_i(\mathbf{x})\big)$$

Basic Penalty Algorithm

1 – set k=1, choose $\mathbf{x}^{(1)}$, penalty factor $r^{(1)}$ e $\beta>1$ and tolerance ε.
2 – min $\Phi(\mathbf{x}, r^{(k)})$ (e.g. DFP, CG…) to obtain $\mathbf{x}^{*(k+1)}$
3 – check $||\mathbf{x}^{*(k+1)} - \mathbf{x}^{*(k)}|| \le \varepsilon$ (stop if true)
4 – set $\mathbf{x}^{(k)} = \mathbf{x}^{*(k+1)}$, $r^{(k+1)} = \beta r^{(k)}$,
5 – set k=k+1, go to 2.

Augmented Lagrangean

P1
$$\begin{aligned} &\text{minimize} \quad f(\mathbf{x}) \\ &\text{subject to} \quad \mathbf{h}(\mathbf{x}) = \mathbf{0} \end{aligned}$$

Equivalent to:

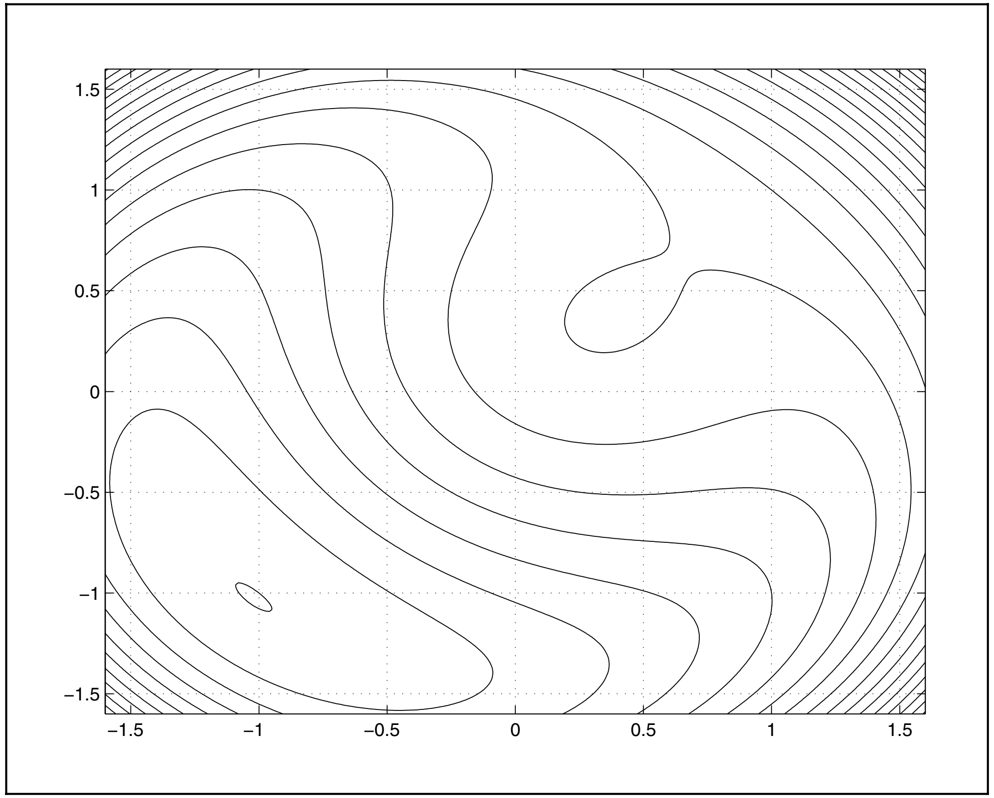
P2
$$\begin{aligned} &\text{minimize} \quad f(\mathbf{x}) + \frac{1}{2} c |\mathbf{h}(\mathbf{x})|^2 \\ &\text{subject to} \quad \mathbf{h}(\mathbf{x}) = \mathbf{0} \end{aligned}$$

Augmented Lagrangean :

$$\phi(\boldsymbol{\lambda}) = \min\{f(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{h}(\mathbf{x}) + \frac{1}{2} c |\mathbf{h}(\mathbf{x})|^2\}$$

KKT necessary conditions (in x) for P1 and P2 lead to:

$$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + c\mathbf{h}(\mathbf{x}(\boldsymbol{\lambda}_k))$$



c=1, $\lambda = -0.4$

Augmented Lagrangean

P1

Grad f(**x***) + λ^* ***Grad*** $h(\mathbf{x}^*)=0$

$h(\mathbf{x}^*)=0$

P2

Grad f(**x***) + $(\lambda^* + ch(\mathbf{x}*))$ ***Grad*** $h(\mathbf{x}^*)=0$

$h(\mathbf{x}^*)=0$

Comparing KKT necessary conditions for P1 and P2 lead to:

$$\lambda^* = \lambda^* + ch(\mathbf{x}^*)$$

Basic Method: For fixed λ solve for **x**

$$\min\{f(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{h}(\mathbf{x}) + \frac{1}{2} c|\mathbf{h}(\mathbf{x})|^2\}$$

Updating formula for multipliers:

$$\lambda_i^{(k+1)} = \lambda_i^{(k)} + ch_i(\mathbf{x}^*)$$

Inequality Constraints

$$\begin{aligned} &\text{minimize} \quad f(\mathbf{x}) \\ &\text{subject to} \quad \mathbf{g}(\mathbf{x}) \le \mathbf{0}, \end{aligned}$$

Equivalent to:

$$\begin{aligned} &\text{minimize} \quad f(\mathbf{x}) \\ &\text{subject to} \quad g_j(x) + z_j^2 = 0, \quad j = 1, 2, \ldots, p. \end{aligned}$$

Note: The min now is for **x** and **z**

Define $v_j = z_j^2$

$$\phi(\boldsymbol{\mu}) = \min_{\mathbf{v} \ge \mathbf{0}, \mathbf{x}} \{f(\mathbf{x}) + \boldsymbol{\mu}^T[\mathbf{g}(\mathbf{x}) + \mathbf{v}] + \frac{1}{2}c|\mathbf{g}(\mathbf{x}) + \mathbf{v}|^2\}.$$

$$P_j = \mu_j[g_j(\mathbf{x}) + v_j] + \frac{1}{2}c[g_j(\mathbf{x}) + v_j]^2.$$

min $P_j(v_j)$ with respect to v_j to obtain

$$P_j = \frac{1}{2c}\{[\max\ (0, \mu_j + cg_j(\mathbf{x}))]^2 - \mu_j^2\}.$$

Augmented Lagrangean :

$$\phi(\mu) = \min_{\mathbf{x}} \left\{ f(\mathbf{x}) + \sum_{j=1}^{p} P_c(g_j(\mathbf{x}), \mu_j) \right\}$$

1 – choose $\mathbf{x}^{(1)}$ and $\boldsymbol{\mu}^{(1)}$, k=1

2 – at iteration k

Solve for **x** (inner min) to obtain $\mathbf{x}^{(\mathbf{k})*}$ with $\boldsymbol{\mu}^{(k)}$ fixed

3 – fixing $\mathbf{x}=\mathbf{x}^{(\mathbf{k})*}$ now Max $\phi(\boldsymbol{\mu})$ for $\boldsymbol{\mu}$, and obtain updating for $\boldsymbol{\mu}^{k+1}$:

$\boldsymbol{\mu}^{k+1} = \max(0, \mu^k + cg(\mathbf{x}^{(\mathbf{k})*}))$ $\qquad$ $\boldsymbol{\mu}^{k+1} = \mu^k + \max(-\mu^k, cg(\mathbf{x}^{(\mathbf{k})*}))$

4 – Verify convergence and go to 2