Θεωρία Βελτιστοποίησης

Βασίλειος Μαχαιράς

Πολιτικός Μηχανικός Ph.D.

Μη γραμμικός προγραμματισμός: βελτιστοποίηση με περιορισμούς

Πανεπιστήμιο Θεσσαλίας
Σχολή Θετικών Επιστημών
Τμήμα Πληροφορικής

Διάλεξη 9-10η/2017

Τι παρουσιάστηκε έως σήμερα

- Αναλυτικές μέθοδοι επίλυσης προβλημάτων βελτιστοποίησης.
- Μέθοδοι γραμμικού προγραμματισμού για την επίλυση προβλημάτων με γραμμική αντικειμενική συνάρτηση και γραμμικούς περιορισμούς ισότητας ή ανισότητας.
- Μέθοδοι επίλυσης μονοδιάστατων προβλημάτων βελτιστοποίησης, οι οποίες είναι χρήσιμες στην εύρεση του βέλτιστου μήκους βήματος σε προβλήματα επίλυσης με επαναληπτικές μεθόδους.
- Μέθοδοι επίλυσης μη γραμμικών προβλημάτων χωρίς περιορισμούς

Τι θα παρουσιαστεί

- Μέθοδοι επίλυσης μη γραμμικών προβλημάτων ΜΕ περιορισμούς

Εισαγωγή

- Η γενική περιγραφή προβλήματος βελτιστοποίησης με περιορισμούς είναι ως εξής:

$$\text{Find } \mathbf{X} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} \text{ which minimizes } f(\mathbf{X})$$

subject to

$$g_j(\mathbf{X}) \leq 0, \quad j = 1, 2, \ldots, m$$

$$h_k(\mathbf{X}) = 0, \quad k = 1, 2, \ldots, p$$

Table 7.1 Constrained Optimization Techniques

Direct methods	Indirect methods
Random search methods	Transformation of variables technique
Heuristic search methods	Sequential unconstrained minimization techniques
Complex method	Interior penalty function method
Objective and constraint approximation methods	Exterior penalty function method
Sequential linear programming method	Augmented Lagrange multiplier method
Sequential quadratic programming method	
Methods of feasible directions	
Zoutendijk's method	
Rosen's gradient projection method	
Generalized reduced gradient method	

Χαρακτηριστικά προβλήματος με περιορισμούς

1. Ο περιορισμός πιθανόν να μην επηρεάζει τη βέλτιστη λύση, οπότε λύνεται σαν να μην έχει περιορισμούς.

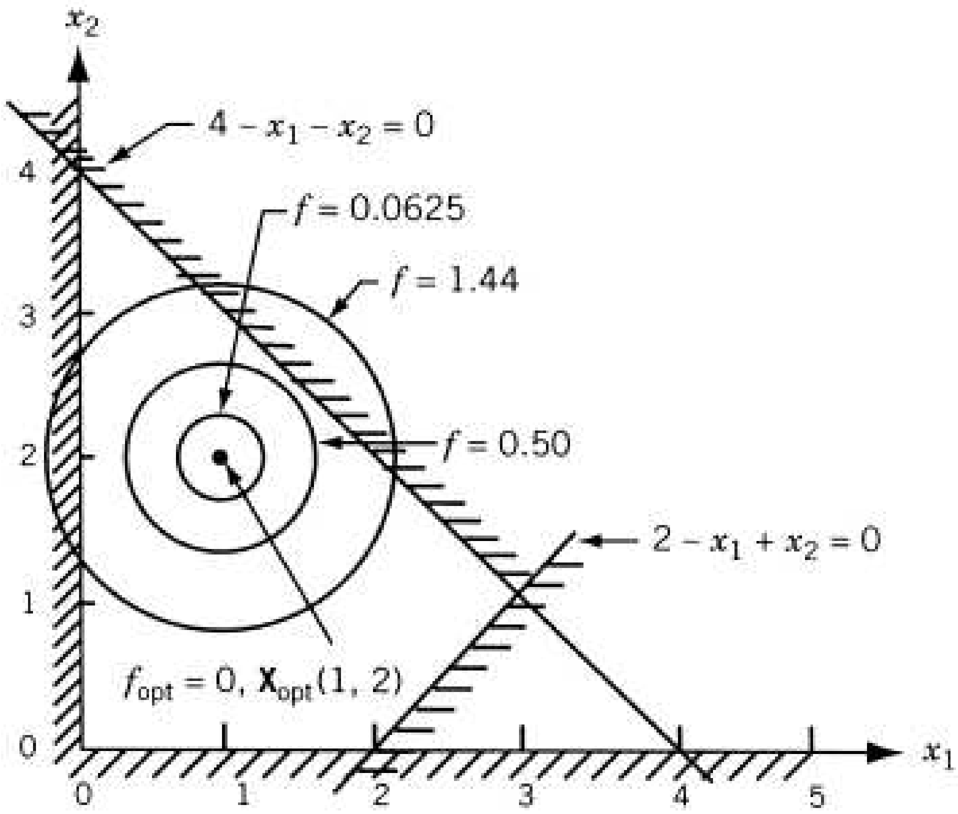


Figure 7.1 Constrained and unconstrained minima are the same (linear constraints).

Χαρακτηριστικά προβλήματος με περιορισμούς

2. Η βέλτιστη (μοναδική) λύση υφίσταται στο όριο του περιορισμού. Η απαραίτητη συνθήκη Kuhn – Tucker υποδείχνει ότι το αρνητικό της κλίσης περιγράφεται ως γραμμικός συνδυασμός των κλίσεων των ενεργών περιορισμών.

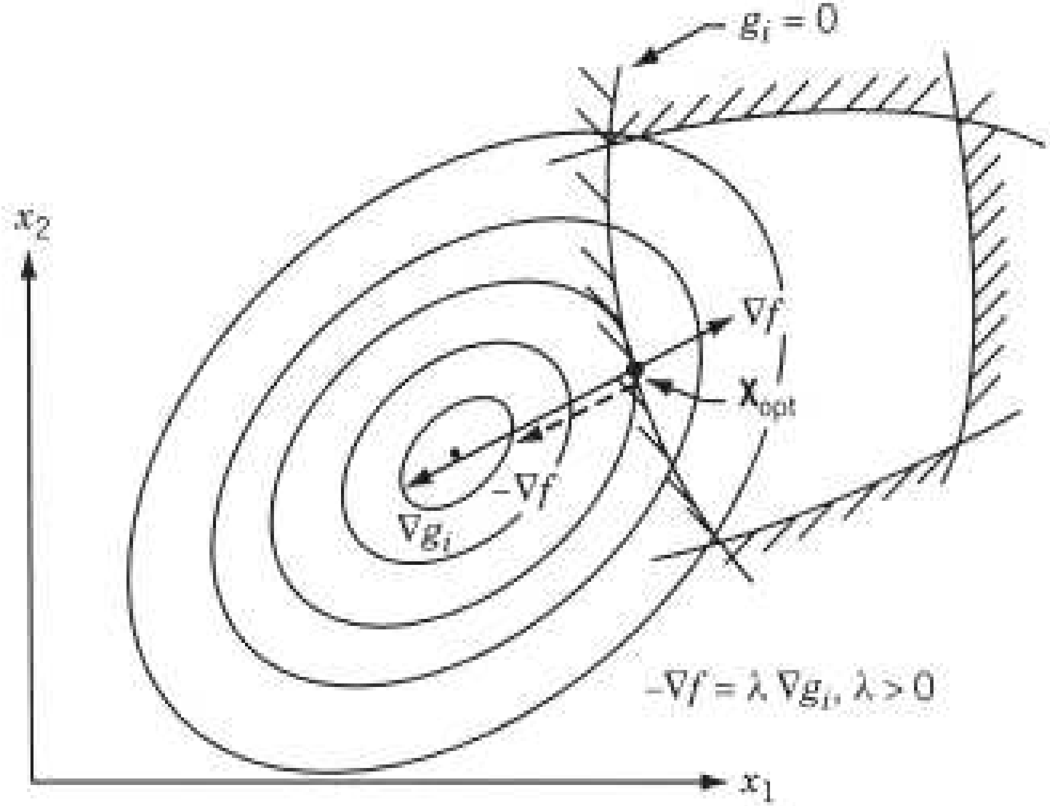


Figure 7.2 Constrained minimum occurring on a nonlinear constraint.

Χαρακτηριστικά προβλήματος με περιορισμούς

3. Αν η αντικειμενική συνάρτηση έχει 2 ή περισσότερα τοπικά ακρότατα τότε το περιορισμένο πρόβλημα ίσως έχει πολλά ακρότατα.

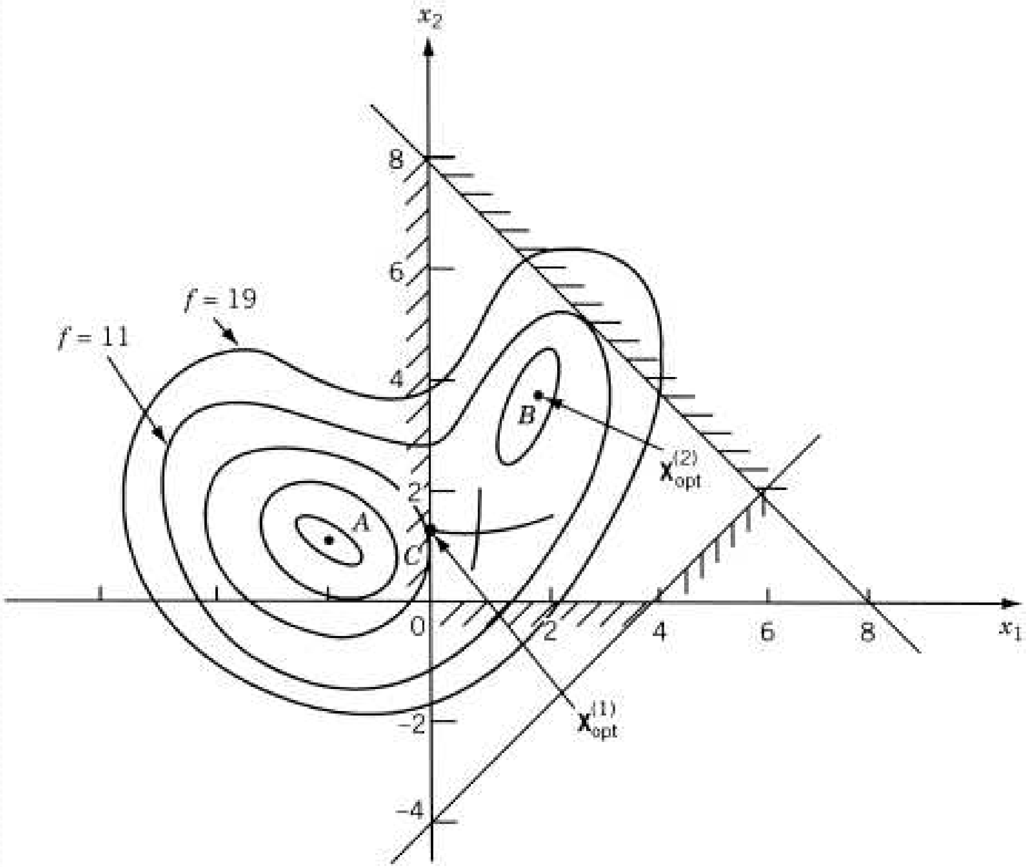


Figure 7.3 Relative minima introduced by objective function.

Χαρακτηριστικά προβλήματος με περιορισμούς

4. Σε μερικές περιπτώσεις η αντικειμενική συνάρτηση έχει μοναδική βέλτιστη λύση, όμως η εισαγωγή περιορισμών αναδεικνύει πολλά τοπικά ακρότατα.

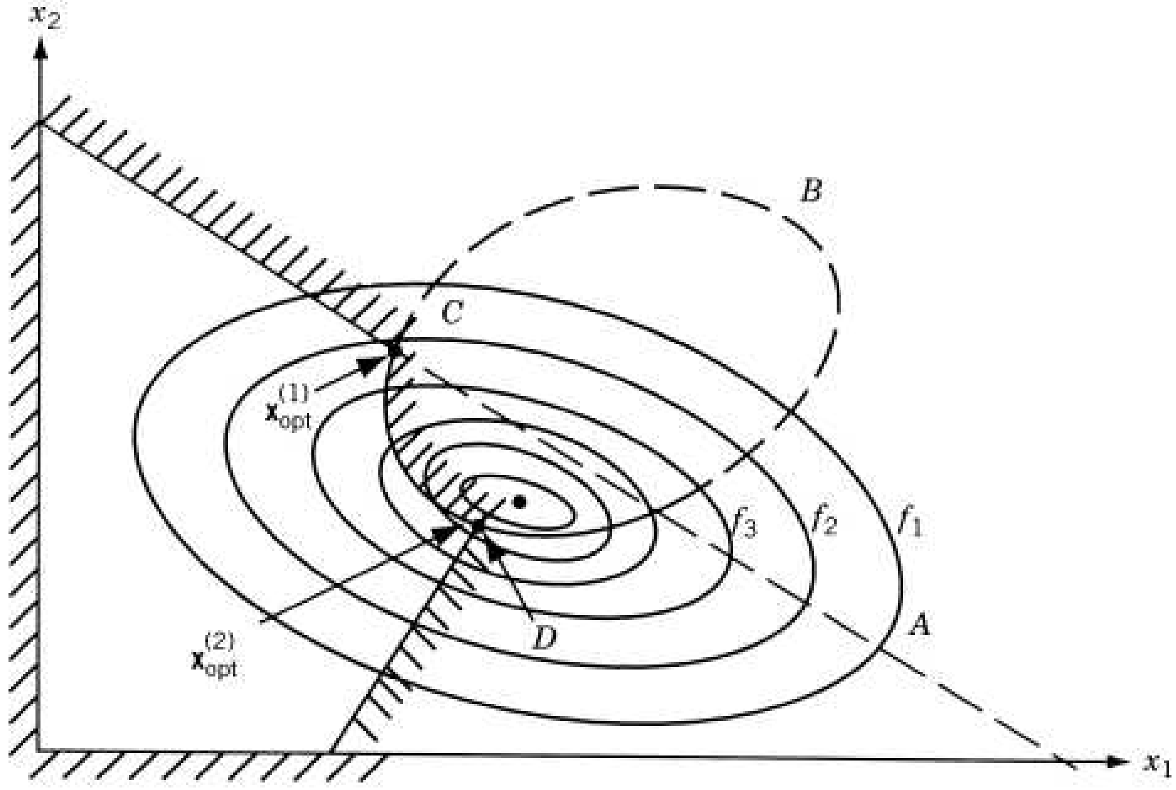
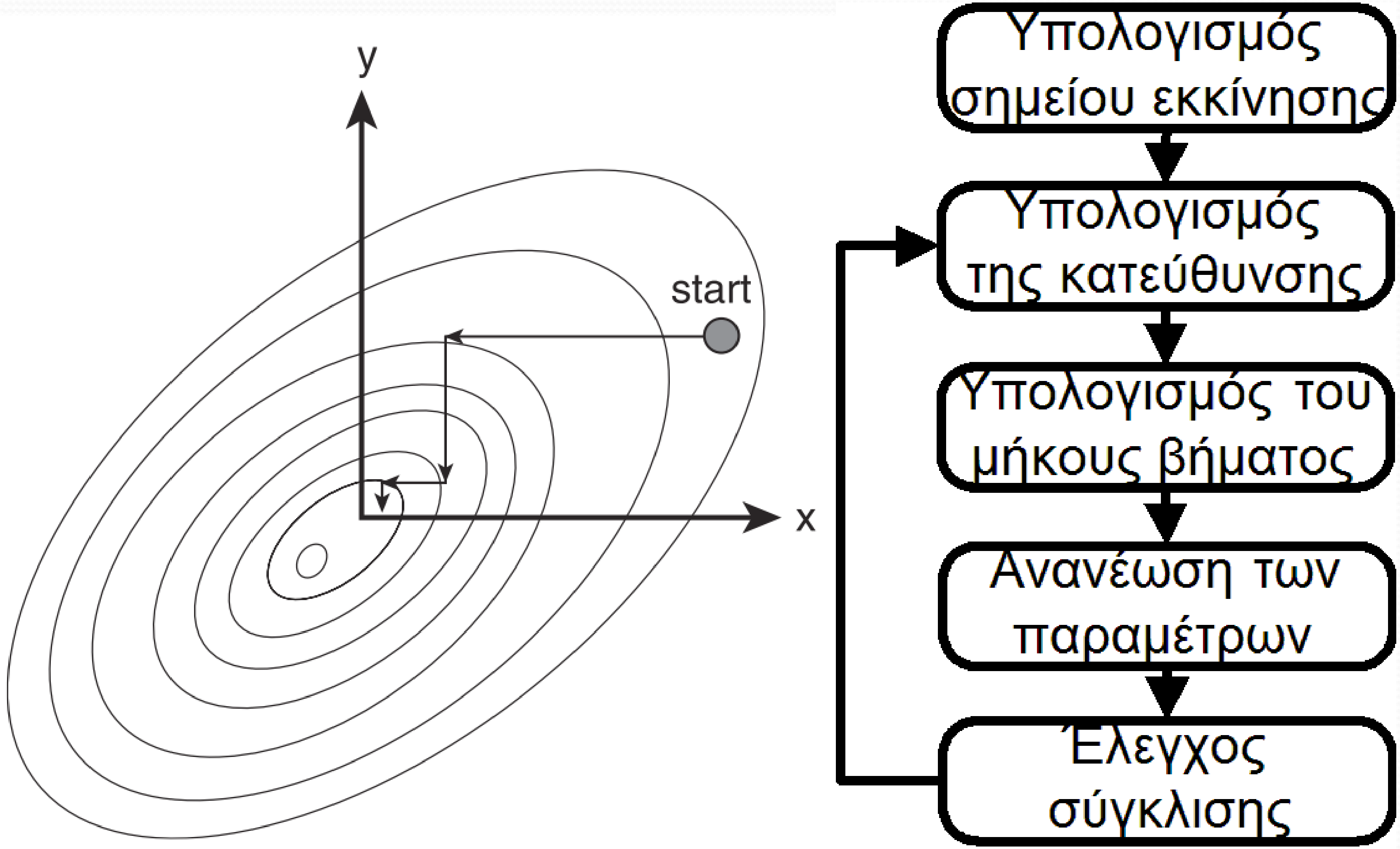


Figure 7.4 Relative minima introduced by constraints.

Μέθοδοι άμεσης αναζήτησης (Direct search methods)



Μέθοδοι τυχαίας αναζήτησης (random search methods)

The random search methods described for unconstrained minimization (Section 6.2) can be used, with minor modifications, to solve a constrained optimization problem. The basic procedure can be described by the following steps:

1. Generate a trial design vector using one random number for each design variable.
2. Verify whether the constraints are satisfied at the trial design vector. Usually, the equality constraints are considered satisfied whenever their magnitudes lie within a specified tolerance. If any constraint is violated, continue generating new trial vectors until a trial vector that satisfies all the constraints is found.
3. If all the constraints are satisfied, retain the current trial vector as the best design if it gives a reduced objective function value compared to the previous best available design. Otherwise, discard the current feasible trial vector and proceed to step 1 to generate a new trial design vector.
4. The best design available at the end of generating a specified maximum number of trial design vectors is taken as the solution of the constrained optimization problem.

It can be seen that several modifications can be made to the basic procedure indicated above. For example, after finding a feasible trial design vector, a feasible direction can be generated (using random numbers) and a one-dimensional search can be conducted along the feasible direction to find an improved feasible design vector.

Complex Method

- O Box επέκτεινε τη μέθοδο simplex, ώστε να μπορεί να επιλύσει προβλήματα με περιορισμούς.
- Αντίστοιχα με τη μέθοδο simplex σχηματίζονται πολυγωνικά σχήματα, ενώ σε κάθε δοκιμή εξετάζεται η εφικτότητα (feasibility) της λύσης.
- Η μέθοδος complex τυπικά επεκτείνεται. Όταν συναντηθεί κάποιο όριο συστέλλεται και «πλαταίνει». Έπειτα επεκτείνεται κατά μήκος του ορίου, εκτός και αν διαφοροποιηθούν οι ισοϋψείς. Η μέθοδος διαχειρίζεται και περισσότερα από ένα όρια και μπορεί να περιστρέφεται στις γωνίες.

Complex Method (συνέχεια)

- Αν η εφικτή περιοχή είναι μη κυρτή δεν εξασφαλίζεται ότι το κεντροΐδές των εφικτών σημείων είναι επίσης εφικτό. Σε τέτοια περίπτωση δεν μπορεί η διαδικασία να βρει επόμενο σημείο.
- Η μέθοδος complex δεν είναι αποδοτική όταν οι μεταβλητές είναι πολλές.
- Δεν μπορεί να χρησιμοποιηθεί για προβλήματα με περιορισμούς ισότητας.
- Απαιτείται η εκκίνηση από σημείο, το οποίο είναι εφικτό.

Διαδοχικός γραμμικός προγραμματισμός (Sequential ή Successive Linear Programming (SLP) ή Cutting Plane Method)

- Η μέθοδος SLP επιλύνει ΜΗ γραμμικά προβλήματα προσεγγίζοντάς τα με γραμμικές συναρτήσεις.
- Η αντικειμενική συνάρτηση και οι συναρτήσεις περιορισμών προσεγγίζονται με σειρά Taylor 1ου βαθμού στο συγκεκριμένο σημείο $\mathbf{X_i}$.
- Το παραγόμενο γραμμικό πρόβλημα λύνεται αποδοτικά με τη μέθοδο simplex για να βρεθεί το επόμενο σημείο $\mathbf{X_{i+1}}$.
- Αν το σημείο $\mathbf{X_{i+1}}$ δεν ικανοποιεί τα κριτήρια σύγκλισης τότε το πρόβλημα «γραμμικοποιείται» ξανά στο νέο σημείο $\mathbf{X_{i+1}}$ και η διαδικασία επαναλαμβάνεται.

Algorithm. The SLP algorithm can be stated as follows:

1. Start with an initial point $\mathbf{X}_1$ and set the iteration number as $i = 1$. The point $\mathbf{X}_1$ need not be feasible.
2. Linearize the objective and constraint functions about the point $\mathbf{X}_i$ as

$$f(\mathbf{X}) \approx f(\mathbf{X}_i) + \nabla f(\mathbf{X}_i)^{\mathrm{T}}(\mathbf{X} - \mathbf{X}_i) \tag{7.14}$$

$$g_j(\mathbf{X}) \approx g_j(\mathbf{X}_i) + \nabla g_j(\mathbf{X}_i)^{\mathrm{T}}(\mathbf{X} - \mathbf{X}_i) \tag{7.15}$$

$$h_k(\mathbf{X}) \approx h_k(\mathbf{X}_i) + \nabla h_k(\mathbf{X}_i)^{\mathrm{T}}(\mathbf{X} - \mathbf{X}_i) \tag{7.16}$$

3. Formulate the approximating linear programming problem as[†]

$$\text{Minimize } f(\mathbf{X}_i) + \nabla f_i^{\mathrm{T}}(\mathbf{X} - \mathbf{X}_i)$$

subject to

$$\begin{aligned} g_j(\mathbf{X}_i) + \nabla g_j(\mathbf{X}_i)^T(\mathbf{X} - \mathbf{X}_i) &\le 0, \quad j = 1, 2, \ldots, m \\ h_k(\mathbf{X}_i) + \nabla h_k(\mathbf{X}_i)^T(\mathbf{X} - \mathbf{X}_i) &= 0, \quad k = 1, 2, \ldots, p \end{aligned} \tag{7.17}$$

4. Solve the approximating LP problem to obtain the solution vector $\mathbf{X}_{i+1}$.

Έπειτα ελέγχονται τα κριτήρια σύγκλισης και αν όχι η διαδικασία επαναλαμβάνεται.

Μέθοδοι των εφικτών κατευθύνσεων

- Επιλέγεται ένα σημείο εκκίνησης και η αναζήτηση κατευθύνεται επαναληπτικά ως εξής: $\mathbf{X_{i+1}} = \mathbf{X_i} + \lambda \mathbf{S_i}$
- Όπου $\mathbf{X_i}$ είναι το σημείο εκκίνησης στην i-στη επανάληψη, $\mathbf{S_i}$ η κατεύθυνση της μετακίνησης, λ η απόσταση της μετακίνησης (μέγεθος βήματος) και $\mathbf{X_{i+1}}$ το σημείο που αναδεικνύει η i-στη επανάληψη.
- Το λ επιλέγεται, ώστε να το $\mathbf{X_{i+1}}$ να βρίσκεται στην εφικτή περιοχή.
- Η κατεύθυνση $\mathbf{S_i}$ επιλέγεται, ώστε α) μια μικρή μετακίνηση σ' αυτή την κατεύθυνση να μην παραβιάζει κανένα περιορισμό και β) η τιμή της αντικειμενικής συνάρτησης να βελτιώνεται προς τα εκεί.
- Μέθοδοι των εφικτών κατευθύνσεων είναι η μέθοδος του Zoutendijk και η gradient projection μέθοδος του Rosen.

Zoutendijk's method of feasible directions

1. Start with an initial feasible point $\mathbf{X}_1$ and small numbers ε_1, ε_2, and ε_3 to test the convergence of the method. Evaluate $f(\mathbf{X}_1)$ and $g_j(\mathbf{X}_1)$, $j = 1, 2, \ldots, m$. Set the iteration number as $i = 1$.
2. If $g_j(\mathbf{X}_i) < 0$, $j = 1, 2, \ldots, m$ (i.e., $\mathbf{X}_i$ is an interior feasible point), set the current search direction as

$$\mathbf{S}_i = -\nabla f(\mathbf{X}_i) \tag{7.29}$$

   Normalize $\mathbf{S}_i$ in a suitable manner and go to step 5. If at least one $g_j(\mathbf{X}_i) = 0$, go to step 3.
3. Find a usable feasible direction $\mathbf{S}$ by solving the direction-finding problem:

$$\text{Minimize} \quad -\alpha \tag{7.30a}$$

subject to

$$\mathbf{S}^{\mathrm{T}}\nabla g_j(\mathbf{X}_i) + \theta_j\alpha \le 0, \quad j = 1, 2, \ldots, p \tag{7.30b}$$

$$\mathbf{S}^{\mathrm{T}}\nabla f + \alpha \le 0 \tag{7.30c}$$

$$-1 \le s_i \le 1, \quad i = 1, 2, \ldots, n \tag{7.30d}$$

where s_i is the ith component of $\mathbf{S}$, the first p constraints have been assumed to be active at the point $\mathbf{X}_i$ (the constraints can always be renumbered to satisfy this requirement), and the values of all θ_j can be taken as unity. Here α can be taken as an additional design variable.

Zoutendijk's method of feasible directions

4. If the value of α^* found in step 3 is very nearly equal to zero, that is, if $\alpha^* \leq \varepsilon_1$, terminate the computation by taking $\mathbf{X}_{\text{opt}} \simeq \mathbf{X}_i$. If $\alpha^* > \varepsilon_1$, go to step 5 by taking $\mathbf{S}_i = \mathbf{S}$.

5. Find a suitable step length λ_i along the direction $\mathbf{S}_i$ and obtain a new point $\mathbf{X}_{i+1}$ as

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \lambda_i \mathbf{S}_i \tag{7.31}$$

The methods of finding the step length λ_i will be considered later.

6. Evaluate the objective function $f(\mathbf{X}_{i+1})$.

7. Test for the convergence of the method. If

$$\left| \frac{f(\mathbf{X}_i) - f(\mathbf{X}_{i+1})}{f(\mathbf{X}_i)} \right| \leq \varepsilon_2 \quad \text{and} \quad \|\mathbf{X}_i - \mathbf{X}_{i+1}\| \leq \varepsilon_3 \tag{7.32}$$

terminate the iteration by taking $\mathbf{X}_{\text{opt}} \simeq \mathbf{X}_{i+1}$. Otherwise, go to step 8.

8. Set the new iteration number as $i = i + 1$, and repeat from step 2 onward.

Generalized Reduced Gradient (GRG)

The basic concept of GRG method entails linearizing the Non-linear objective and constraint functions at a local solution with Taylor expansion equation. Then, the concept of reduced gradient method is employed which divides the variable set into two subsets of basic and non-basic variable and the concept of implicit variable elimination to express the basic variable by the non-basic variable. Finally, the constraints are eliminated and the variable space is deduced to only non-basic variables. The proven efficient method for non-constraints NLP problems are involved to solve the approximated problem and, then, the next optimal solution for the approximated problem should be found. The processes repeat again until it

Generalized Reduced Gradient (GRG)

Algorithm

1. *Specify the design and state variables*. Start with an initial trial vector **X**. Identify the design and state variables (**Y** and **Z**) for the problem using the following guidelines.

   (a) The state variables are to be selected to avoid singularity of the matrix, $[D]$.

   (b) Since the state variables are adjusted during the iterative process to maintain feasibility, any component of **X** that is equal to its lower or upper bound initially is to be designated a design variable.

   (c) Since the slack variables appear as linear terms in the (originally inequality) constraints, they should be designated as state variables. However, if the initial value of any state variable is zero (its lower bound value), it should be designated a design variable.

2. *Compute the generalized reduced gradient*. The GRG is determined using Eq. (7.105). The derivatives involved in Eq. (7.105) can be evaluated numerically, if necessary.

3. *Test for convergence*. If all the components of the GRG are close to zero, the method can be considered to have converged and the current vector **X** can be taken as the optimum solution of the problem. For this, the following test can be used:

$$||\mathbf{G}_R|| \leq \varepsilon$$

   where ε is a small number. If this relation is not satisfied, we go to step 4.

Generalized Reduced Gradient (GRG)

4. *Determine the search direction*. The GRG can be used similar to a gradient of an unconstrained objective function to generate a suitable search direction, **S**. The techniques such as steepest descent, Fletcher–Reeves, Davidon–Fletcher–Powell, or Broydon–Fletcher–Goldfarb–Shanno methods can be used for this purpose. For example, if a steepest descent method is used, the vector **S** is determined as

$$\mathbf{S} = -\mathbf{G}_R \tag{7.110}$$

5. *Find the minimum along the search direction*. Although any of the one -dimensional minimization procedures discussed in Chapter 5 can be used to find a local minimum of f along the search direction **S**, the following procedure can be used conveniently.

   (a) Find an estimate for λ as the distance to the nearest side constraint. When design variables are considered, we have

$$\lambda = \begin{cases} \dfrac{y_i^{(u)} - (y_i)_{\text{old}}}{s_i} & \text{if } s_i > 0 \\[2ex] \dfrac{y_i^{(l)} - (y_i)_{\text{old}}}{s_i} & \text{if } s_i < 0 \end{cases} \tag{7.111}$$

   where s_i is the ith component of **S**. Similarly, when state variables are considered, we have, from Eq. (7.102),

$$d\mathbf{Z} = -[D]^{-1}[C]\,d\mathbf{Y} \tag{7.112}$$

Generalized Reduced Gradient (GRG)

Using $d\mathbf{Y} = \lambda\mathbf{S}$, Eq. (7.112) gives the search direction for the variables $\mathbf{Z}$ as

$$\mathbf{T} = -[D]^{-1}[C]\mathbf{S} \tag{7.113}$$

Thus

$$\lambda = \begin{cases} \dfrac{z_i^{(u)} - (z_i)_{\text{old}}}{t_i} & \text{if } t_i > 0 \\ \dfrac{z_i^{(l)} - (z_i)_{\text{old}}}{t_i} & \text{if } t_i < 0 \end{cases} \tag{7.114}$$

where t_i is the ith component of $\mathbf{T}$.

(b) The minimum value of λ given by Eq. (7.111), λ_1, makes some design variable attain its lower or upper bound. Similarly, the minimum value of λ given by Eq. (7.114), λ_2, will make some state variable attain its lower or upper bound. The smaller of λ_1 or λ_2 can be used as an upper bound on the value of λ for initializing a suitable one-dimensional minimization procedure. The quadratic interpolation method can be used conveniently for finding the optimal step length λ^*.

(c) Find the new vector $\mathbf{X}_{\text{new}}$:

$$\mathbf{X}_{\text{new}} = \begin{Bmatrix} \mathbf{Y}_{\text{old}} + d\mathbf{Y} \\ \mathbf{Z}_{\text{old}} + d\mathbf{Z} \end{Bmatrix} = \begin{Bmatrix} \mathbf{Y}_{\text{old}} + \lambda^*\mathbf{S} \\ \mathbf{Z}_{\text{old}} + \lambda^*\mathrm{T} \end{Bmatrix} \tag{7.115}$$

Generalized Reduced Gradient (GRG)

If the vector $\mathbf{X}_{\text{new}}$ corresponding to λ^* is found infeasible, then $\mathbf{Y}_{\text{new}}$ is held constant and $\mathbf{Z}_{\text{new}}$ is modified using Eq. (7.108) with $d\mathbf{Z} = \mathbf{Z}_{\text{new}} - \mathbf{Z}_{\text{old}}$. Finally, when convergence is achieved with Eq. (7.108), we find that

$$\mathbf{X}_{\text{new}} = \begin{Bmatrix} \mathbf{Y}_{\text{old}} + \Delta\mathbf{Y} \\ \mathbf{Z}_{\text{old}} + \Delta\mathbf{Z} \end{Bmatrix} \tag{7.116}$$

and go to step 1.

Sequential quadratic programming (SQP)

Sequential quadratic programming (SQP)

- Αποτελεί μια από τις καλύτερες μεθόδους βελτιστοποίησης.
- Η θεωρητική της βάση σχετίζεται με α) την επίλυση ενός συνόλου μη γραμμικών εξισώσεων, χρησιμοποιώντας τη μέθοδο Newton και β) την παραγωγή μη γραμμικών εξισώσεων, χρησιμοποιώντας τη συνθήκη Kuhn-Tucker στη Lagrangian του προβλήματος βελτιστοποίησης με περιορισμούς.
- Είναι επαναληπτική μέθοδος.
- Συνάρτηση fmincon του MATLAB (medium scale)

Sequential quadratic programming (SQP)

- Κατασκευάζεται η συνάρτηση Lagrange με προσέγγιση 2ου βαθμού.
- Κατασκευάζονται γραμμικές συναρτήσεις, οι οποίες προσεγγίζουν τους περιορισμούς.
- Επιλύεται το πρόβλημα 2ας τάξης για βρεθεί η κατεύθυνση **S**
- Μονοδιάστατη αναζήτηση 1-D για το μήκος βήματος.
- Ανανέωση της προσεγγιστικής συνάρτησης Lagrange

Algorithm	Relative advantages	Relative disadvantage
SLP	Easy to implement Widely used in practice Rapid convergence when optimum is at a vertex Can handle very large problems Does not attempt to satisfy equalities at each iteration Can benefit from improvements in LP solvers	May converge slowly on problems with nonvertex optima Will usually violate nonlinear constraints until convergence, often by large amounts

Algorithm	Relative advantages	Relative disadvantage
SLP	Easy to implement Widely used in practice Rapid convergence when optimum is at a vertex Can handle very large problems Does not attempt to satisfy equalities at each iteration Can benefit from improvements in LP solvers	May converge slowly on problems with nonvertex optima Will usually violate nonlinear constraints until convergence, often by large amounts
SQP	Usually requires fewest functions and gradient evaluations of all three algorithms (by far) Does not attempt to satisfy equalities at each iteration	Will usually violate nonlinear constraints until convergence, often by large amounts Harder than SLP to implement Requires a good QP solver

Algorithm	Relative advantages	Relative disadvantage
GRG	Probably most robust of all three methods Versatile--especially good for unconstrained or linearly constrained problems but also works well for nonlinear constraints Can utilize existing process simulators employing Newton's method Once it reaches a feasible solution it remains feasible and then can be stopped at any stage with an improved solution	Hardest to implement Needs to satisfy equalities at each step of the algorithm

Indirect Search (Descent) Methods
Μέθοδοι καθόδου ή μέθοδοι κλίσης

Τεχνικές μετασχηματισμού

- Όταν οι περιορισμοί εξαρτώνται από τις μεταβλητές και έχουν συγκεκριμένη απλή μορφή τότε είναι δυνατόν να γίνουν μετασχηματισμοί στις ανεξάρτητες μεταβλητές, ώστε οι περιορισμοί να ικανοποιούνται αυτόματα.
- Έτσι ένα πρόβλημα με περιορισμούς μπορεί να μετασχηματιστεί σε πρόβλημα χωρίς περιορισμούς και να επιλυθεί με τις αντίστοιχες μεθόδους.
- Οι περιορισμοί πρέπει να είναι απλές συναρτήσεις.
- Για μερικούς περιορισμούς μπορεί να μην είναι δυνατή η εύρεση του αναγκαίου μετασχηματισμού.
- Αν δεν εξαλειφθούν όλοι οι περιορισμοί ίσως είναι καλύτερα να μην υλοποιηθεί κανένας μετασχηματισμός. Διότι ο μερικός μετασχηματισμός ίσως παράγει αντικειμενική συνάρτηση, η οποία θα είναι αρκετά πιο δύσκολο να επιλυθεί σε σχέση με την αρχική συνάρτηση.

Τεχνικές μετασχηματισμού (παραδείγματα)

1. If lower and upper bounds on x_i are specified as

$$l_i \le x_i \le u_i \tag{7.148}$$

these can be satisfied by transforming the variable x_i as

$$x_i = l_i + (u_i - l_i)\sin^2 y_i \tag{7.149}$$

where y_i is the new variable, which can take any value.

2. If a variable x_i is restricted to lie in the interval (0, 1), we can use the transformation:

$$x_i = \sin^2 y_i, \quad x_i = \cos^2 y_i$$

$$x_i = \frac{e^{yi}}{e^{yi} + e^{-yi}} \quad \text{or} \quad x_i = \frac{y_i^2}{1 + y_i^2} \tag{7.150}$$

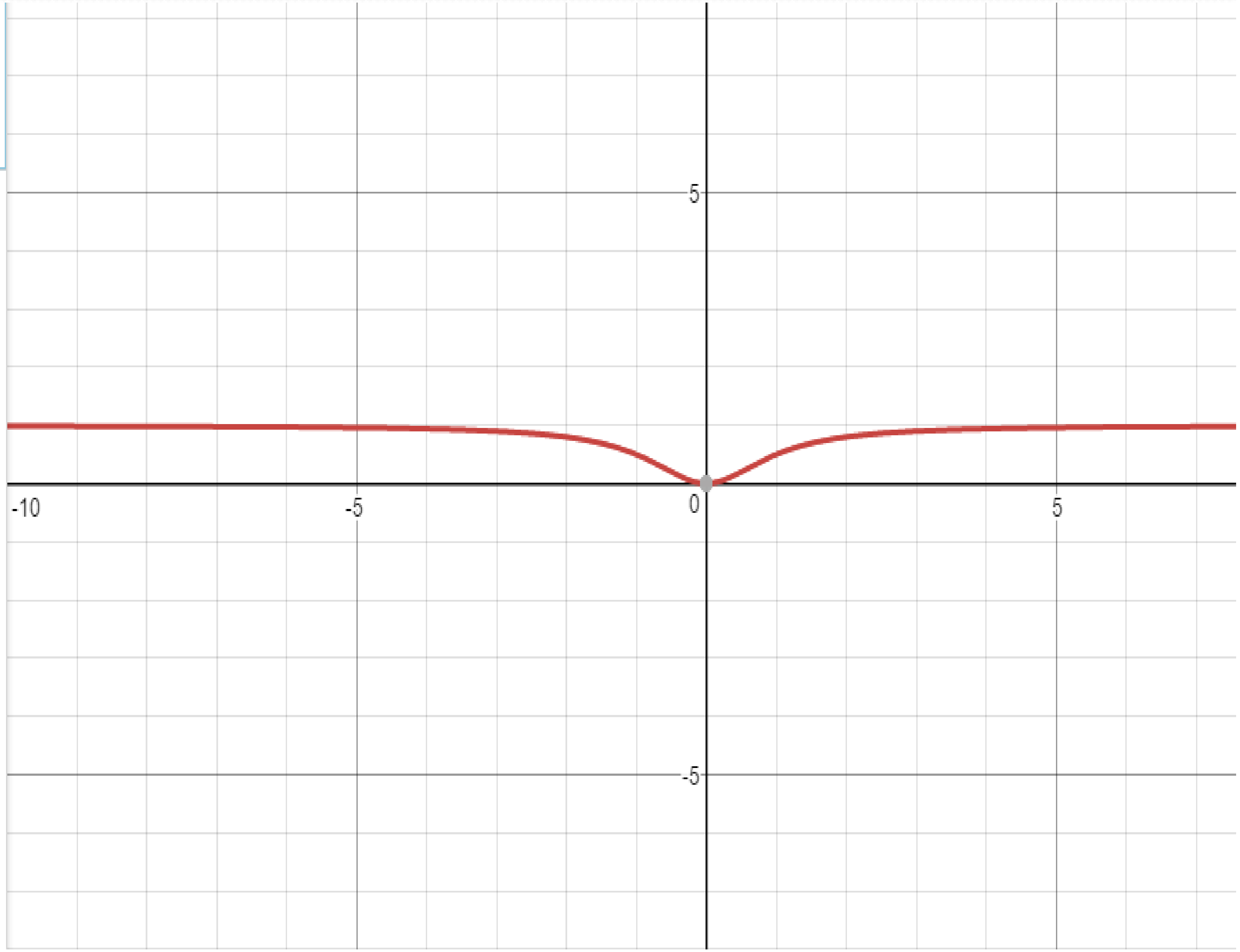
3. If the variable x_i is constrained to take only positive values, the transformation can be

$$x_i = \text{abs}(y_i), \quad x_i = y_i^2 \quad \text{or} \quad x_i = e^{yi} \tag{7.151}$$

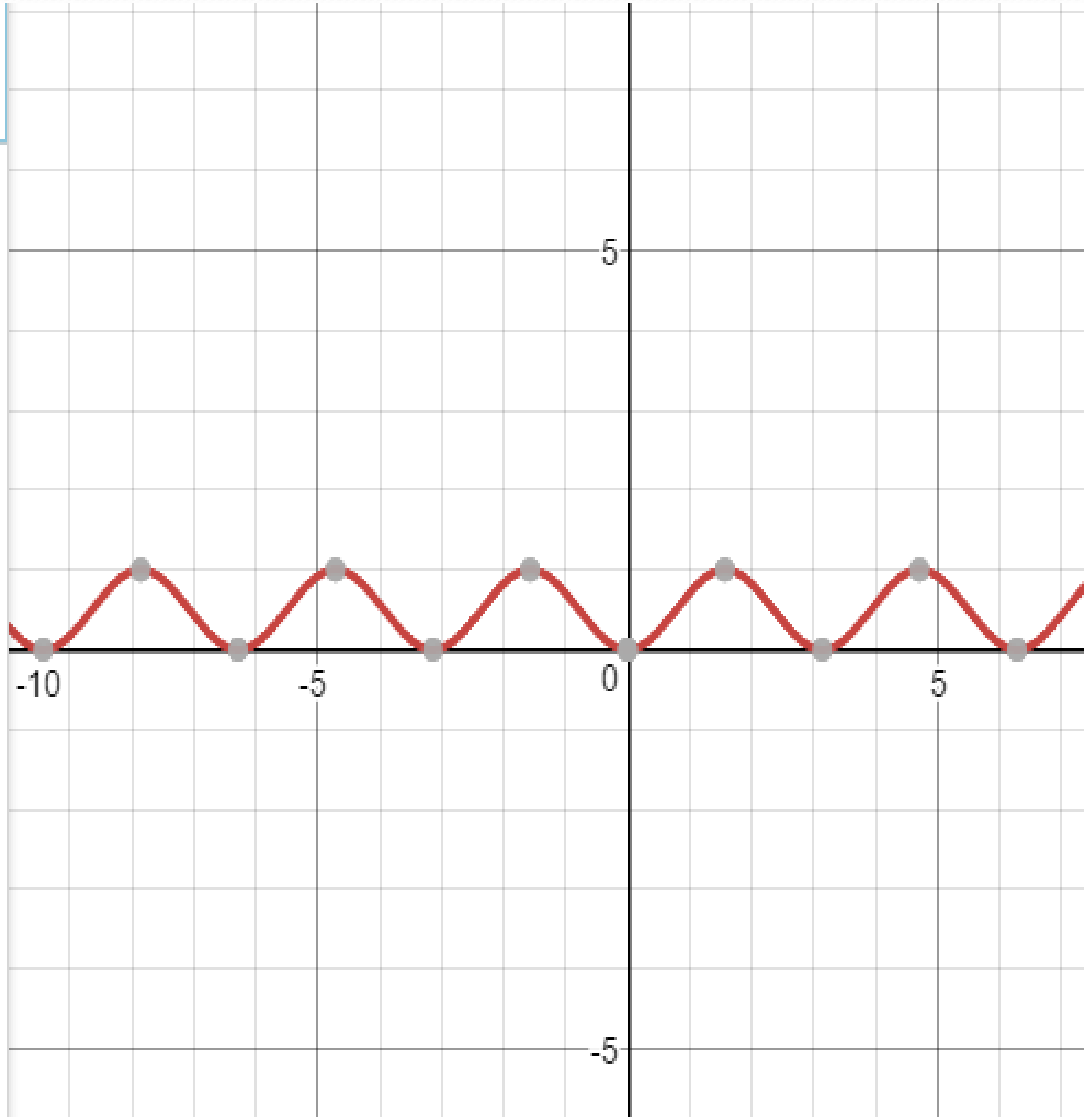
4. If the variable is restricted to take values lying only in between -1 and 1, the transformation can be

$$x_i = \sin y_i, \quad x_i = \cos y_i, \quad \text{or} \quad x_i = \frac{2y_i}{1 + y_i^2} \tag{7.152}$$

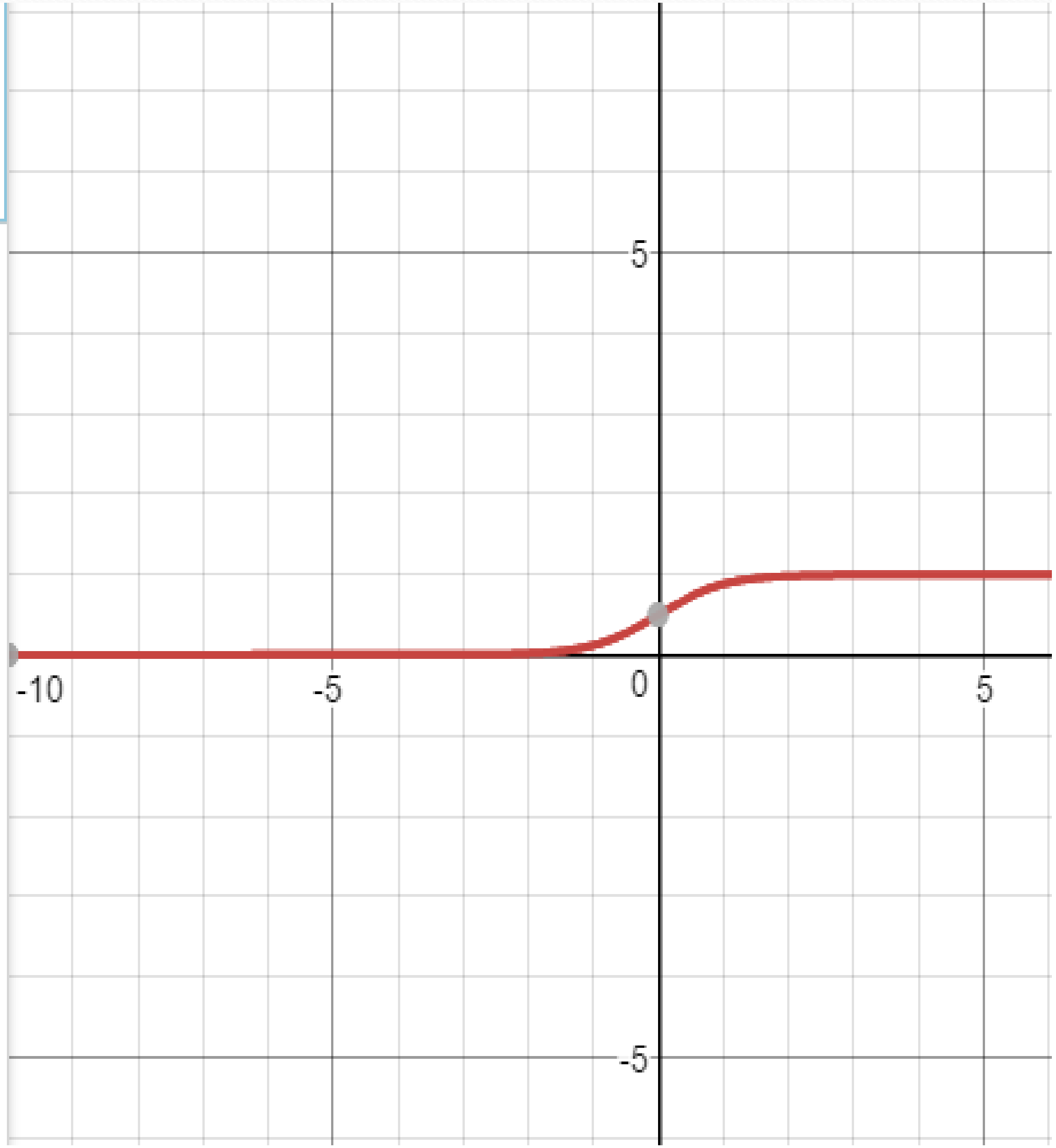
$$y = \frac{x^2}{\left(1 + x^2\right)}$$



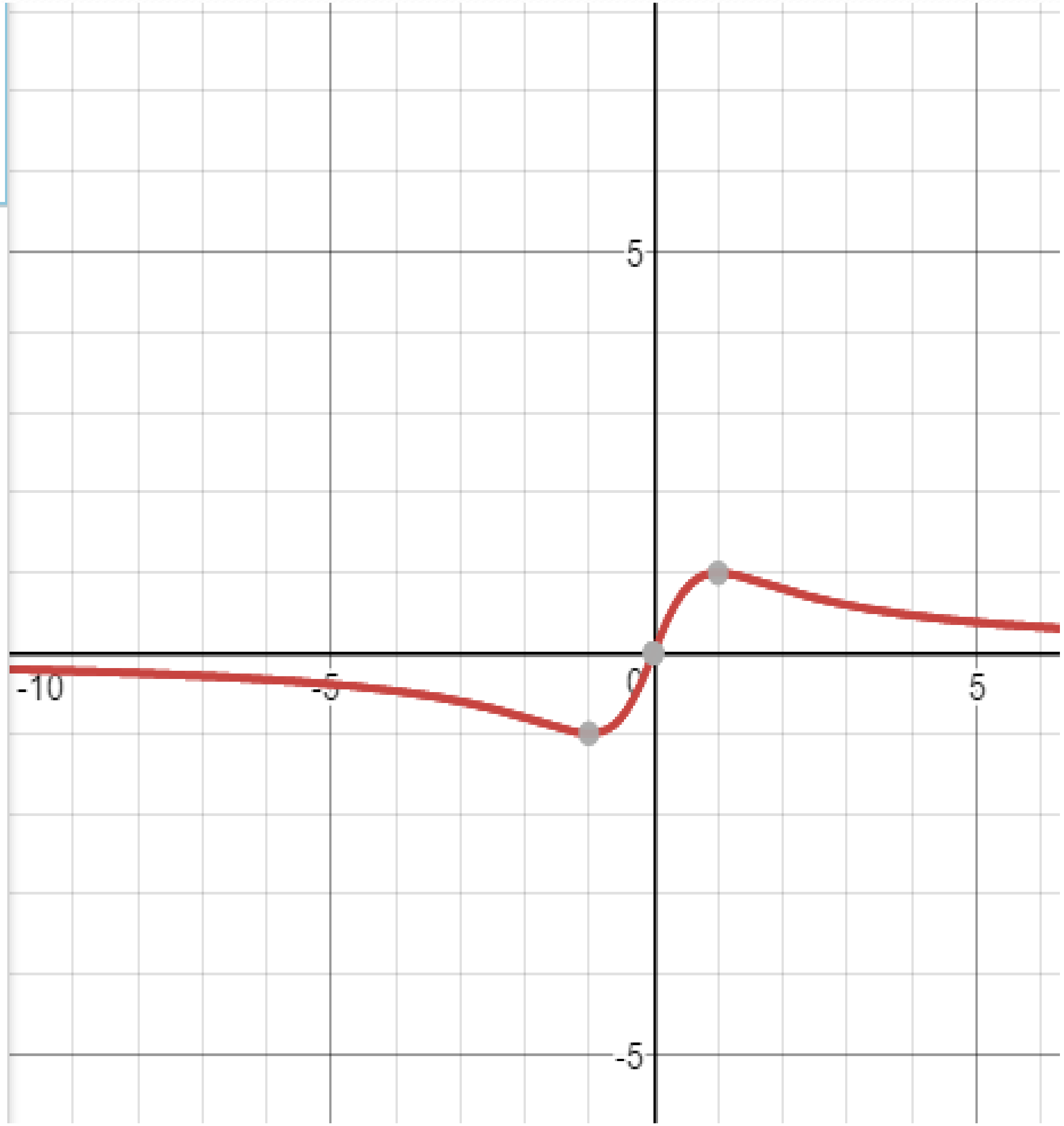
$$y = \sin(x) \cdot \sin(x)$$



$$y = \frac{e^x}{e^x + e^{-x}}$$



$$y = \frac{2 \cdot x}{1 + x^2}$$



Example 7.6 Find the dimensions of a rectangular prism-type box that has the largest volume when the sum of its length, width, and height is limited to a maximum value of 60 in. and its length is restricted to a maximum value of 36 in.

SOLUTION Let x_1, x_2, and x_3 denote the length, width, and height of the box, respectively. The problem can be stated as follows:

$$\text{Maximize } f(x_1, x_2, x_3) = x_1 x_2 x_3 \tag{E_1}$$

subject to

$$x_1 + x_2 + x_3 \leq 60 \tag{E_2}$$

$$x_1 \leq 36 \tag{E_3}$$

$$x_i \geq 0, \quad i = 1, 2, 3 \tag{E_4}$$

By introducing new variables as

$$y_1 = x_1, \quad y_2 = x_2, \quad y_3 = x_1 + x_2 + x_3 \tag{E$_5$}$$

or

$$x_1 = y_1, \quad x_2 = y_2, \quad x_3 = y_3 - y_1 - y_2 \tag{E$_6$}$$

the constraints of Eqs. (E_2) to (E_4) can be restated as

$$0 \leq y_1 \leq 36, \quad 0 \leq y_2 \leq 60, \quad 0 \leq y_3 \leq 60 \tag{E$_7$}$$

where the upper bound, for example, on y_2 is obtained by setting $x_1 = x_3 = 0$ in Eq. (E_2). The constraints of Eq. (E_7) will be satisfied automatically if we define new variables z_i, $i = 1, 2, 3$, as

$$y_1 = 36 \sin^2 z_1, \quad y_2 = 60 \sin^2 z_2, \quad y_3 = 60 \sin^2 z_3 \tag{E$_8$}$$

Thus the problem can be stated as an unconstrained problem as follows:

$$\begin{aligned} &\text{Maximize } f(z_1, z_2, z_3) \\ &= y_1 y_2 (y_3 - y_1 - y_2) \\ &= 2160 \sin^2 z_1 \sin^2 z_2 (60 \sin^2 z_3 - 36 \sin^2 z_1 - 60 \sin^2 z_2) \end{aligned} \tag{E$_9$}$$

The necessary conditions of optimality yield the relations

$$\frac{\partial f}{\partial z_1} = 259{,}200 \sin z_1 \cos z_1 \sin^2 z_2 (\sin^2 z_3 - \tfrac{6}{5}\sin^2 z_1 - \sin^2 z_2) = 0 \qquad (E_{10})$$

$$\frac{\partial f}{\partial z_2} = 518{,}400 \sin^2 z_1 \sin z_2 \cos z_2 (\tfrac{1}{2}\sin^2 z_3 - \tfrac{3}{10}\sin^2 z_1 - \sin^2 z_2) = 0 \qquad (E_{11})$$

$$\frac{\partial f}{\partial z_3} = 259{,}200 \sin^2 z_1 \sin^2 z_2 \sin z_3 \cos z_3 = 0 \qquad (E_{12})$$

Equation (E_{12}) gives the nontrivial solution as $\cos z_3 = 0$ or $\sin^2 z_3 = 1$. Hence Eqs. (E_{10}) and (E_{11}) yield $\sin^2 z_1 = \frac{5}{9}$ and $\sin^2 z_2 = \frac{1}{3}$. Thus the optimum solution is given by $x_1^* = 20$ in., $x_2^* = 20$ in., $x_3^* = 20$ in., and the maximum volume $= 8000$ in^3.

Συναρτήσεις ποινής
Penalty functions

Εξασφάλιση περιορισμών μέσω συναρτήσεων ποινής (Penalty functions)

- Η λογική των συναρτήσεων ποινών είναι απλή και αποσκοπεί στην εισαγωγή περιορισμών σε κάποιο πρόβλημα βελτιστοποίησης.
- Έστω ότι έχουμε ένα πρόβλημα ελαχιστοποίησης της αντικειμενικής συνάρτησης f(**X**), στο οποίο απαιτείται η εξασφάλιση κάποιου περιορισμού g(**X**)<0. Ένας απλός τρόπος εισαγωγής του περιορισμού είναι μετασχηματίζοντας την αντικειμενική συνάρτηση ως εξής:

$$f'(\boldsymbol{X}) = \begin{cases} f(\boldsymbol{X}) + \infty, & \alpha\nu\ g(\boldsymbol{X}) < 0 \\ f(\boldsymbol{X}), & \sigma\tau\iota\varsigma\ \acute{\alpha}\lambda\lambda\varepsilon\varsigma\ \pi\varepsilon\rho\iota\pi\tau\acute{\omega}\sigma\varepsilon\iota\varsigma \end{cases}$$

- Η συγκεκριμένη ποινή ονομάζεται «ποινή θανάτου» και δημιουργεί προβλήματα, διότι εισάγει ασυνέχειες στην αντικειμενική συνάρτηση. Υπάρχουν καλύτερες υλοποιήσεις, αλλά βασίζονται στην ίδια λογική.

Εξασφάλιση περιορισμών μέσω συναρτήσεων ποινής (Penalty functions)

- Η χρήση συναρτήσεων ποινών μετασχηματίζει το πρόβλημα με περιορισμούς σε πρόβλημα χωρίς περιορισμούς, οπότε μπορούν να χρησιμοποιηθούν οι αντίστοιχες μέθοδοι βελτιστοποίησης.
- Γενικά η αντικειμενική συνάρτηση $f(x)$ μετασχηματίζεται σε $f'(x) = f(x) + p(x)$
- όπου $p(x)$ μια συνάρτηση η οποία προσθέτει ποινές όταν παραβιάζεται κάποιος περιορισμός.
- Η ποινή μπορεί να εφαρμοστεί και πολλαπλασιαστικά ως εξής (πιο σπάνια περίπτωση): $f'(x) = f(x) \cdot p(x)$

Συναρτήσεις ποινής εξωτερικού σημείου (Exterior point penalty functions)

- Η ποινή εφαρμόζεται μόλις παραβιαστεί κάποιος περιορισμός.
- Ονομάζεται και «χαλαρός περιορισμός», διότι επιτρέπει την αναζήτηση σε περιοχές, στις οποίες παραβιάζονται περιορισμοί.

Συναρτήσεις ποινής εξωτερικού σημείου (Exterior point penalty functions) Παράδειγμα

We will be working with a very simple example:

minimize $f(x) = 100/x$

subject to $x \leq 5$

(With a little thought, you can tell that f(x) will be minimized when x = 5, so we know what answer we should get!)

Before starting, convert any constraints into the form (expression) ≤ 0. In this example, $x \leq 5$ becomes:

$$x - 5 \leq 0$$

Συναρτήσεις ποινής εξωτερικού σημείου (Exterior point penalty functions) Παράδειγμα

With the constraint $x - 5 \leq 0$, we need a penalty that is:

- 0 when $x - 5 \leq 0$ (the constraint is satisfied)
- positive when $x - 5$ is > 0 (the constraint is violated)

This can be done using the operation

$$P(x) = \max(0, x - 5)$$

which returns the maximum of the two values, either 0 or whatever $(x - 5)$ is.

We can make the penalty more severe by using

$$P(x) = \max(0, x - 5)^2.$$

This is known as a quadratic loss function.

Συναρτήσεις ποινής εξωτερικού σημείου (Exterior point penalty functions) Παράδειγμα

It is even easier to convert equality constraints into quadratic loss functions because we don't need to worry about the operation (max, g(x)). We can convert h(x) = c into h(x) – c = 0, then use

$$P(x) = (h(x) - c)^2$$

The lowest value of P(x) will occur when h(x) = c, in which case the penalty P(x) = 0. This is exactly what we want.

Συναρτήσεις ποινής εξωτερικού σημείου (Exterior point penalty functions) Παράδειγμα

Once you have converted your constraints into penalty functions, the basic idea is to add all the penalty functions on to the original objective function and minimize from there:

$$\text{minimize } T(x) = f(x) + P(x)$$

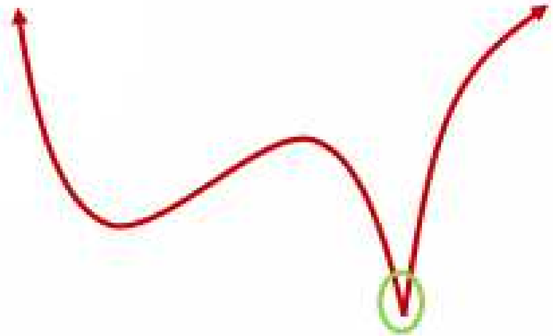
In our example,

$$\text{minimize } T(x) = 100/x + \max(0, x - 5)^2$$

Συναρτήσεις ποινής εξωτερικού σημείου (Exterior point penalty functions)

But… it isn't quite that easy.

The addition of penalty functions can create severe slope changes in the graph at the boundary, which interferes with typical minimization programs.



Fortunately, there are two simple changes that will alleviate this problem.

Συναρτήσεις ποινής εξωτερικού σημείου (Exterior point penalty functions)

The first is to multiply the quadratic loss function by a constant, *r*. This controls how severe the penalty is for violating the constraint.

The accepted method is to start with r = 10, which is a mild penalty. It will not form a very sharp point in the graph, but the minimum point found using r = 10 will not be a very accurate answer because the penalty is not severe enough.

Συναρτήσεις ποινής εξωτερικού σημείου (Exterior point penalty functions)

Then, r is increased to 100 and the function minimized again starting from the minimum point found when r was 10. The higher penalty increases accuracy, and as we narrow in on the solution, the sharpness of the graph is less of a problem.

We continue to increase r values until the solutions converge.

Συναρτήσεις ποινής εξωτερικού σημείου (Exterior point penalty functions)

- Γενική περίπτωση ποινών εξωτερικού σημείου:

In the exterior penalty function method, the ϕ function is generally taken as

$$\phi(\mathbf{X}, r_k) = f(\mathbf{X}) + r_k \sum_{j=1}^{m} \langle g_j(\mathbf{X}) \rangle^q \tag{7.199}$$

where r_k is a positive penalty parameter, the exponent q is a nonnegative constant, and the bracket function $\langle g_j(\mathbf{X}) \rangle$ is defined as

$$\langle g_j(\mathbf{X}) \rangle = \max\langle g_j(\mathbf{X}), 0 \rangle$$
$$= \begin{cases} g_j(\mathbf{X}) & \text{if } g_j(\mathbf{X}) > 0 \\ & \quad \text{(constraint is violated)} \\ 0 & \text{if } g_j(\mathbf{X}) \le 0 \\ & \quad \text{(constraint is satisfied)} \end{cases} \tag{7.200}$$

Συναρτήσεις ποινής εξωτερικού σημείου

1. $q = 0$. Here the ϕ function is given by

$$\phi(\mathbf{X}, r_k) = f(\mathbf{X}) + r_k \sum_{j=1}^{m} \langle g_j(\mathbf{X}) \rangle^0$$

$$= \begin{cases} f(\mathbf{X}) + m r_k & \text{if} \quad \text{all } g_j(\mathbf{X}) > 0 \\ f(\mathbf{X}) & \text{if} \quad \text{all } g_j(\mathbf{X}) \le 0 \end{cases} \tag{7.201}$$

This function is discontinuous on the boundary of the acceptable region as shown in Fig. 7.11 and hence it would be very difficult to minimize this function.

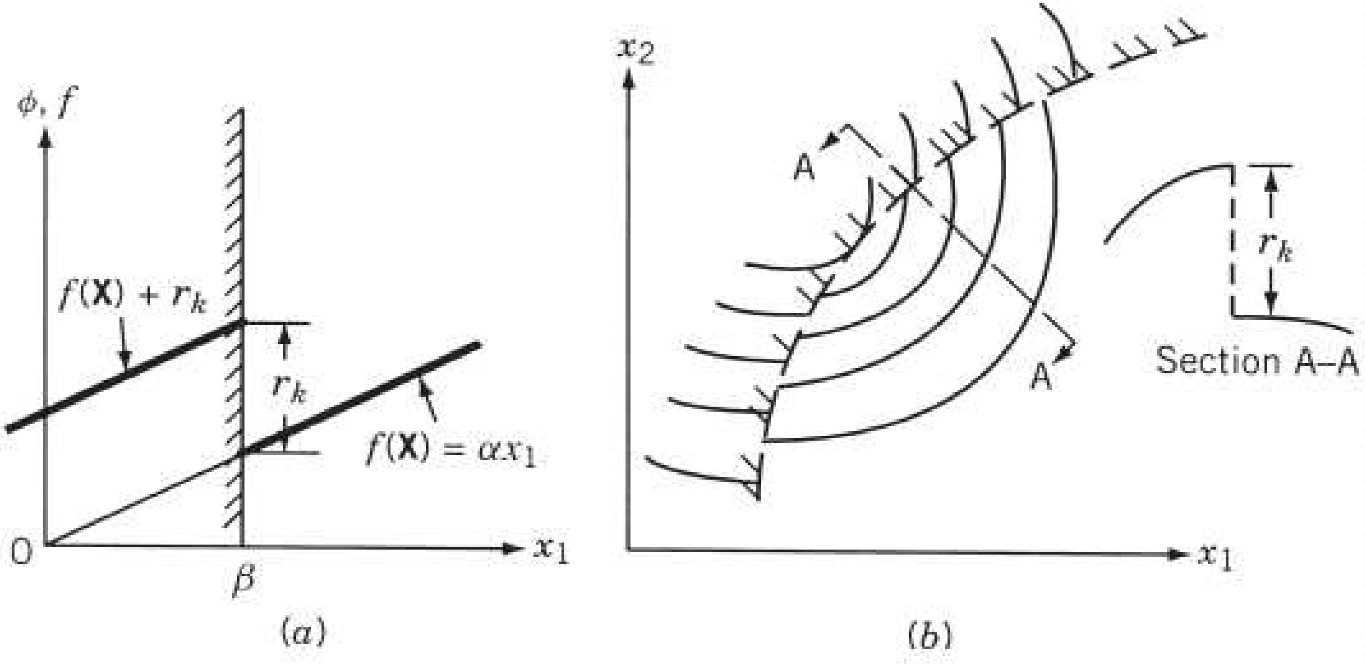


Figure 7.11 A ϕ function discontinuous for $q = 0$.

Συναρτήσεις ποινής εξωτερικού σημείου

2. $0 < q < 1$. Here the ϕ function will be continuous, but the penalty for violating a constraint may be too small. Also, the derivatives of the function are discontinuous along the boundary. Thus it will be difficult to minimize the ϕ function. Typical contours of the ϕ function are shown in Fig. 7.12.

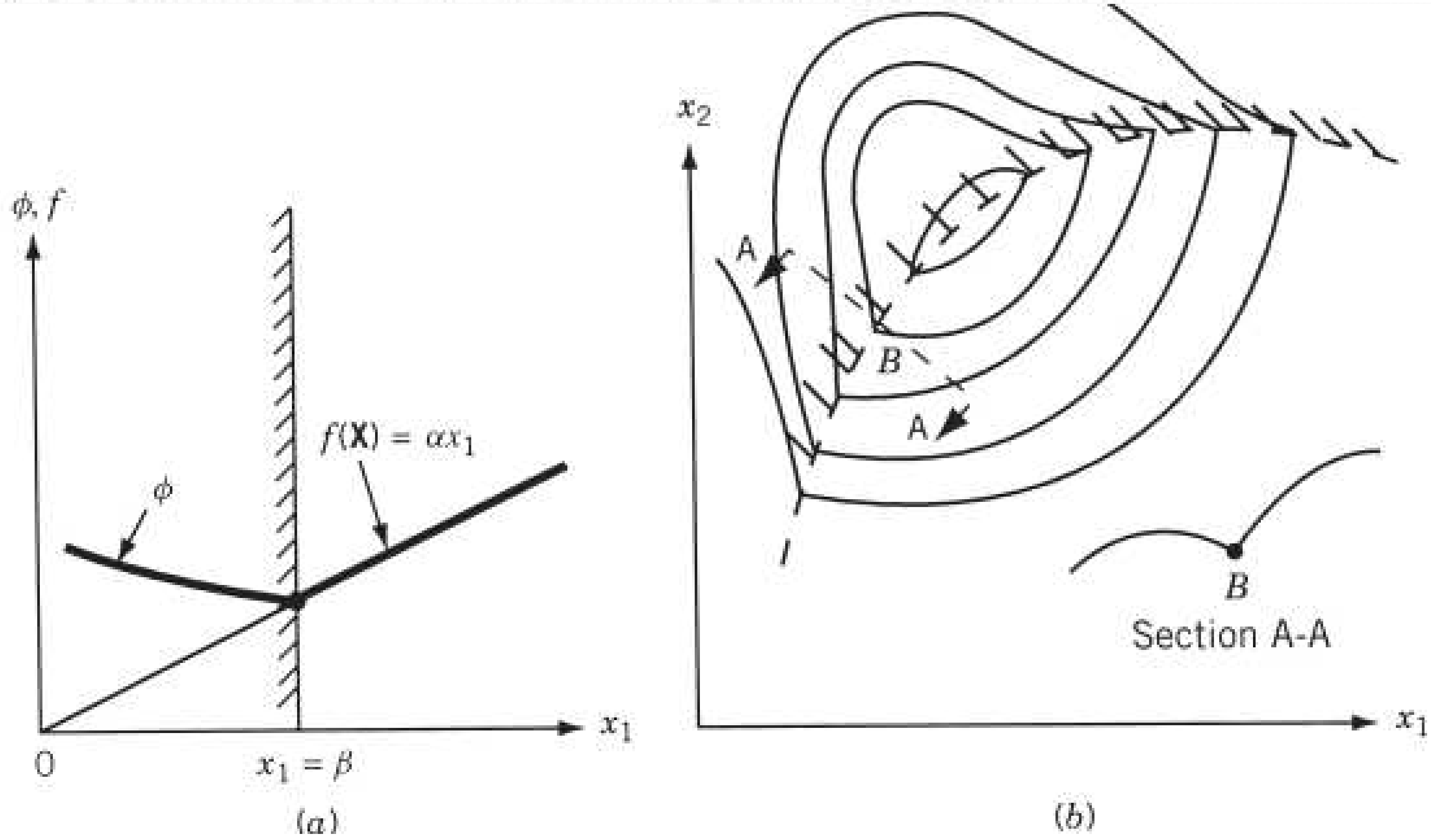


Figure 7.12 Derivatives of a ϕ function discontinuous for $0 < q < 1$.

4. $q > 1$. The ϕ function will have continuous first derivatives in this case as shown in Fig. 7.13. These derivatives are given by

$$\frac{\partial \phi}{\partial x_i} = \frac{\partial f}{\partial x_i} + r_k \sum_{j=1}^{m} q \langle g_j(\mathbf{X}) \rangle^{q-1} \frac{\partial g_j(\mathbf{X})}{\partial x_i} \tag{7.202}$$

Generally, the value of q is chosen as 2 in practical computation. We assume a value of $q > 1$ in subsequent discussion of this method.

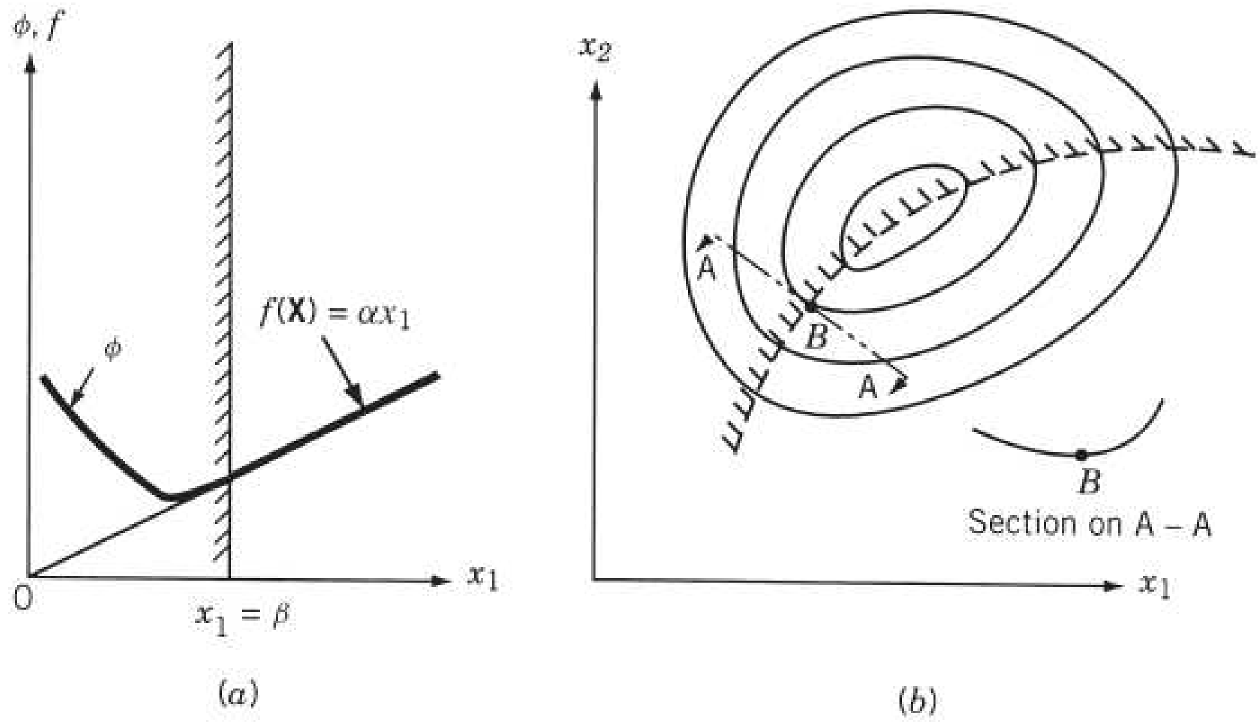


Figure 7.13 A ϕ function for $q > 1$.

Συναρτήσεις ποινής εξωτερικού σημείου

Algorithm. The exterior penalty function method can be stated by the following steps:

1. Start from any design $\mathbf{X}_1$ and a suitable value of r_1. Set $k = 1$.
2. Find the vector $\mathbf{X}_k^*$ that minimizes the function

$$\phi(\mathbf{X}, r_k) = f(\mathbf{X}) + r_k \sum_{j=1}^{m} \langle g_j(\mathbf{X}) \rangle^q$$

3. Test whether the point $\mathbf{X}_k^*$ satisfies all the constraints. If $\mathbf{X}_k^*$ is feasible, it is the desired optimum and hence terminate the procedure. Otherwise, go to step 4.
4. Choose the next value of the penalty parameter that satisfies the relation

$$r_{k+1} > r_k$$

and set the new value of k as original k plus 1 and go to step 2. Usually, the value of r_{k+1} is chosen according to the relation $r_{k+1} = c r_k$, where c is a constant greater than 1.

Συναρτήσεις ποινής εξωτερικού σημείου (Exterior point penalty functions)

A Note About Exterior Penalty Functions

Because exterior penalty functions start outside the feasible region and approach it from the outside, they only find extremes that occur on the boundaries of the feasible region. They will not find interior extremes.

In order to accomplish that, these are often used in combination with interior penalty functions... next lesson!

Συναρτήσεις ποινής

- Γενικά οι συναρτήσεις ποινής σχηματίζονται ως εξής:

Find $\mathbf{X}$ which minimizes $f(\mathbf{X})$

subject to

$$g_j(\mathbf{X}) \leq 0, \quad j = 1, 2, \ldots, m \tag{7.153}$$

This problem is converted into an unconstrained minimization problem by constructing a function of the form

$$\phi_k = \phi(\mathbf{X}, r_k) = f(\mathbf{X}) + r_k \sum_{j=1}^{m} G_j[g_j(\mathbf{X})] \tag{7.154}$$

where G_j is some function of the constraint g_j, and r_k is a positive constant known as the *penalty parameter*. The significance of the second term on the right side of Eq. (7.154), called the *penalty term*, will be seen in Sections 7.13 and 7.15. If the unconstrained minimization of the ϕ function is repeated for a sequence of values of the penalty parameter r_k $(k = 1, 2, \ldots)$, the solution may be brought to converge to that of the original problem stated in Eq. (7.153). This is the reason why the penalty function methods are also known as *sequential unconstrained minimization techniques* (SUMTs).

Τυπικές συναρτήσεις ποινής

The penalty function formulations for inequality constrained problems can be divided into two categories: interior and exterior methods. In the interior formulations, some popularly used forms of G_j are given by

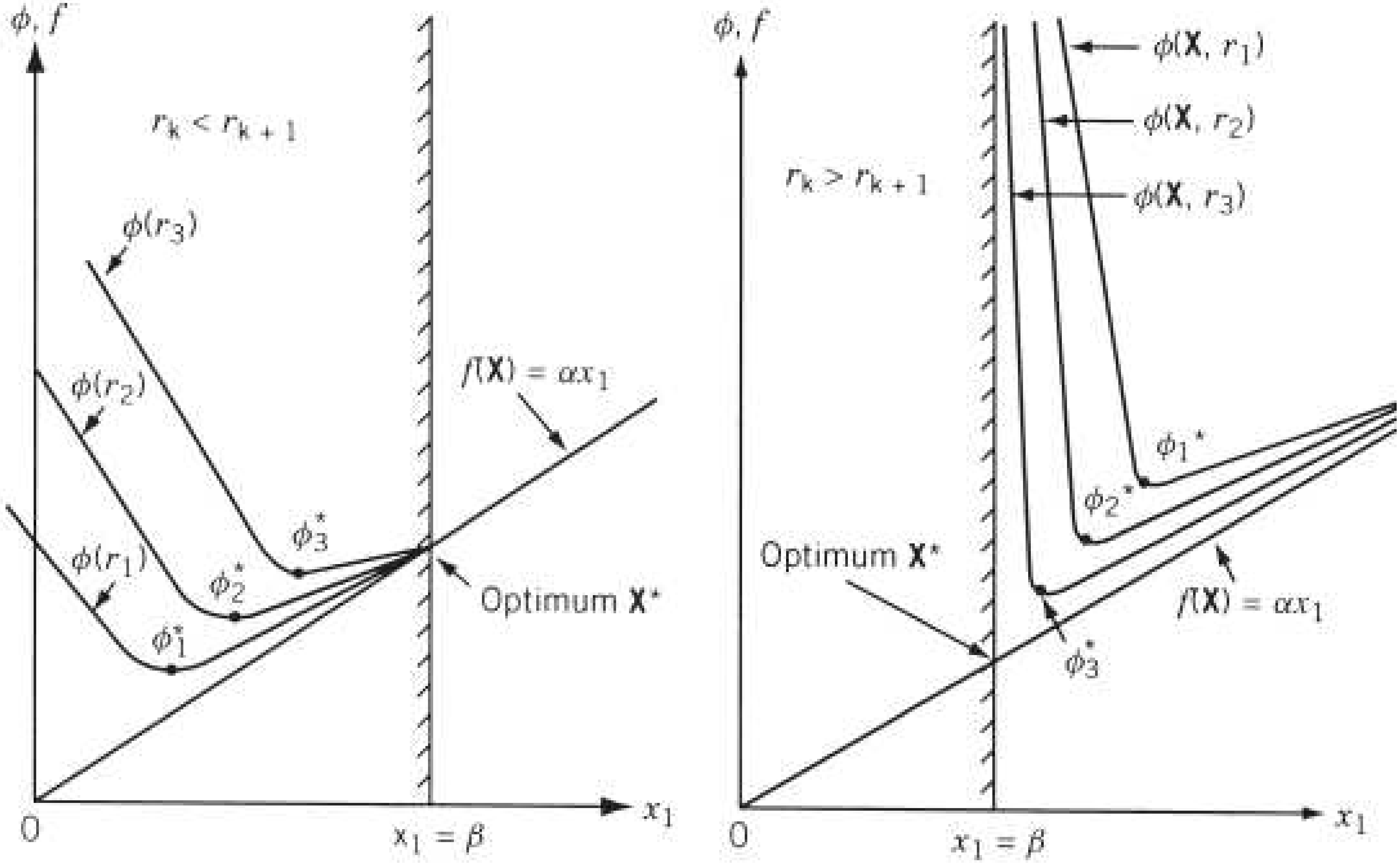
$$G_j = -\frac{1}{g_j(\mathbf{X})} \tag{7.155}$$

$$G_j = \log[-g_j(\mathbf{X})] \tag{7.156}$$

Some commonly used forms of the function G_j in the case of exterior penalty function formulations are

$$G_j = \max[0, g_j(\mathbf{X})] \tag{7.157}$$

$$G_j = \{\max[0, g_i(\mathbf{X})]\}^2 \tag{7.158}$$



$G_j\{g_j(\mathbf{X})\} = \{\max[0, g_j(\mathbf{X})]\}^2$

(*a*)

$G_j\{g_j(\mathbf{X})\} = -1/g_j(\mathbf{X})$

(*b*)

Figure 7.10 Penalty function methods: (*a*) exterior method; (*b*) interior method.

Συναρτήσεις ποινής εσωτερικού σημείου
(Interior point penalty functions ή Barrier Functions)

Συναρτήσεις ποινής εσωτερικού σημείου (Interior point penalty functions ή Barrier Functions)

The simplest interior penalty function, called an "inverse barrier function", involves writing a barrier function for each constraint $g_j(x) \le 0$ as follows:

$$B(x) = -\frac{1}{g_j(x)}$$

This procedure has two effects:

- The negative sign changes the constraint from a negative number (≤ 0) to a positive.
- The fraction ensures that as $g_j(x)$ approaches the constraint boundary at 0, the barrier function gets infinitely large.

Therefore, this barrier creates a positive number that approaches infinity as x nears the boundary.

Συναρτήσεις ποινής εσωτερικού σημείου (Interior point penalty functions ή Barrier Functions)

$$\phi(\mathbf{X}, r_k) = f(\mathbf{X}) - r_k \sum_{j=1}^{m} \frac{1}{g_j(\mathbf{X})} \qquad (7.160)$$

It can be seen that the value of the function ϕ will always be greater than f since $g_j(\mathbf{X})$ is negative for all feasible points $\mathbf{X}$. If any constraint $g_j(\mathbf{X})$ is satisfied critically (with equality sign), the value of ϕ tends to infinity. It is to be noted that the penalty term in Eq. (7.160) is not defined if $\mathbf{X}$ is infeasible. This introduces serious shortcoming while using the Eq. (7.160). Since this equation does not allow any constraint to be violated, it requires a feasible starting point for the search toward the optimum point. However, in many engineering problems, it may not be very difficult to find a point satisfying all the constraints, $g_j(\mathbf{X}) \leq 0$, at the expense of large values of the objective function, $f(\mathbf{X})$. If there is any difficulty in finding a feasible starting point, the method described in the latter part of this section can be used to find a feasible point. Since the initial point as well as each of the subsequent points generated in this method lies inside the acceptable region of the design space, the method is classified as an *interior penalty function formulation*. Since the constraint boundaries act as barriers, the method is also known as a barrier method. The iteration procedure of this method can be summarized as follows.

Συναρτήσεις ποινής εσωτερικού σημείου (Interior point penalty functions ή Barrier Functions)

Iterative Process

1. Start with an initial feasible point $\mathbf{X}_1$ satisfying all the constraints with strict inequality sign, that is, $g_j(\mathbf{X}_1) < 0$ for $j = 1, 2, \ldots, m$, and an initial value of $r_1 > 0$. Set $k = 1$.
2. Minimize $\phi(\mathbf{X}, r_k)$ by using any of the unconstrained minimization methods and obtain the solution $\mathbf{X}_k^*$.
3. Test whether $\mathbf{X}_k^*$ is the optimum solution of the original problem. If $\mathbf{X}_k^*$ is found to be optimum, terminate the process. Otherwise, go to the next step.
4. Find the value of the next penalty parameter, r_{k+1}, as

$$r_{k+1} = c r_k$$

   where $c < 1$.

5. Set the new value of $k = k + 1$, take the new starting point as $\mathbf{X}_1 = \mathbf{X}_k^*$, and go to step 2.

Συναρτήσεις ποινής εσωτερικού σημείου (Interior point penalty functions ή Barrier Functions)

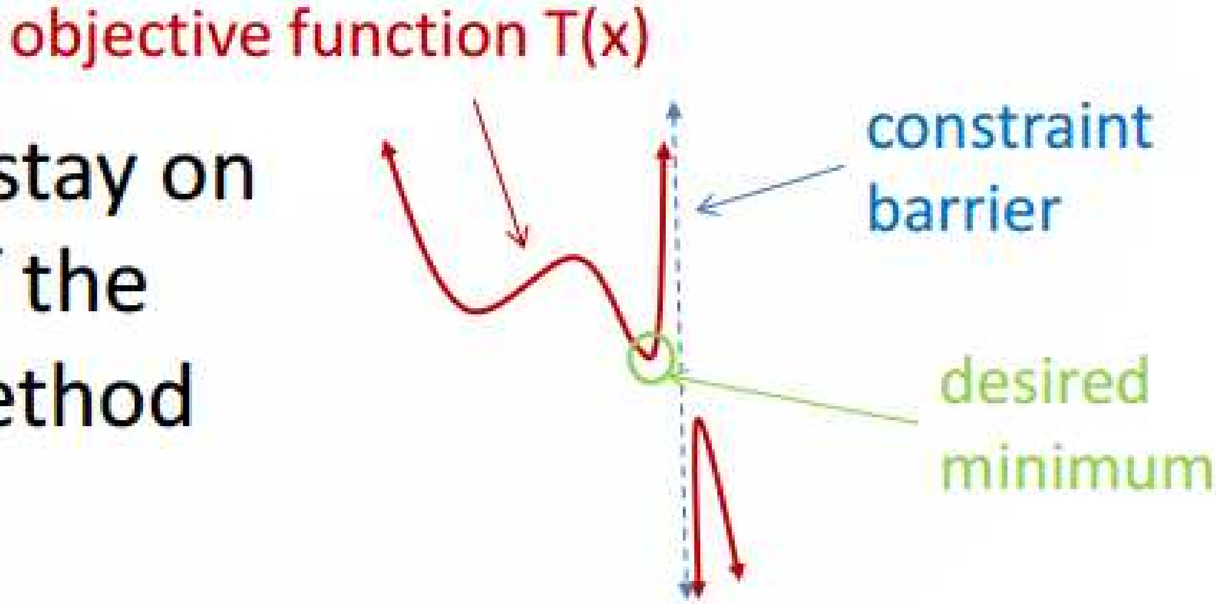
Although the algorithm is straightforward, there are a number of points to be considered in implementing the method:

1. The starting feasible point $\mathbf{X}_1$ may not be readily available in some cases.
2. A suitable value of the initial penalty parameter (r_1) has to be found.
3. A proper value has to be selected for the multiplication factor, c.
4. Suitable convergence criteria have to be chosen to identify the optimum point.
5. The constraints have to be normalized so that each one of them vary between -1 and 0 only.

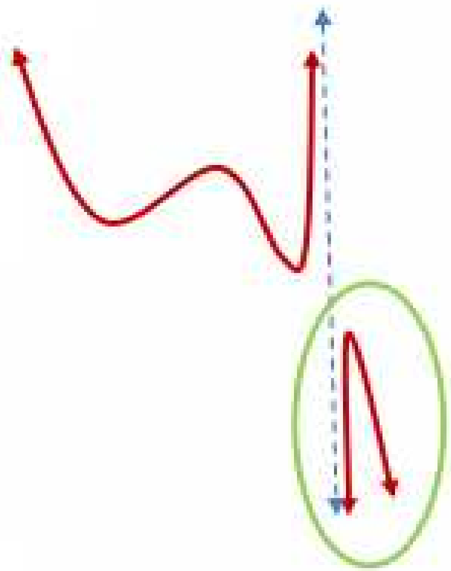
Συναρτήσεις ποινής εσωτερικού σημείου (Interior point penalty functions ή Barrier Functions)

As with exterior penalty functions, the inverse barrier method causes some graphical problems. Because it involves an inverse function 1/g(x), it creates an asymptotic graph:

As long as we stay on the left side of the barrier, the method will work.



Problems with Barriers



If the initial point ever moves outside the boundary, however, the method will fail, because outside the constraint (in this case, the right side of the graph) the other side of the function will come into play.

The values here are lower, so a minimization program will pick them up and run with them, even though they violate the constraint.

The Solution: r, again!

As with exterior penalty functions, part of the solution is to include a multiplier, in this case 1/r. We will begin with r = 1 and gradually increase r by factors of 10.

This has the effect of successively reducing the penalty. In the beginning the penalty is large, preventing the minimum point from crossing the barrier of the constraint. This barrier is so effective that the minimum point will not be very accurate.

As the penalty gets smaller, the minimum point will approach closer to the boundary.

Writing the Penalty Function

Using our previous example,

minimize f(x) = 100/x

subject to x ≤ 5

we will first convert the constraint to x – 5 ≤ 0,

then write the barrier function:

$$B(x) = -\frac{1}{x-5}$$

and then the modified objective function:

$$\text{minimize } T(x) = \frac{100}{x} + \frac{1}{r} \cdot \frac{-1}{x-5}$$

Final Comments

The difficulties on problem 3 bring up an interesting point: penalty methods are not just plug-and-play! There are many variables involved, including:

- the initial value of r
- how fast r grows
- the initial value of x
- the step size in the minimizer

In an actual problem, these variables are tried in different combinations, and typically both interior and exterior methods are used.

Συναρτήσεις ποινής (επεκτάσεις)

7.17 EXTENDED INTERIOR PENALTY FUNCTION METHODS

In the interior penalty function approach, the ϕ function is defined within the feasible domain. As such, if any of the one-dimensional minimization methods discussed in Chapter 5 is used, the resulting optimal step lengths might lead to infeasible designs. Thus the one-dimensional minimization methods have to be modified to avoid this problem. An alternative method, known as the *extended interior penalty function method*, has been proposed in which the ϕ function is defined outside the feasible region. The extended interior penalty function method combines the best features of the interior and exterior methods for inequality constraints. Several types of extended interior penalty function formulations are described in this section.

Linear extended penalty function method

function is constructed as follows:

$$\phi_k = \phi(\mathbf{X}, r_k) = f(\mathbf{X}) + r_k \sum_{j=1}^{m} \tilde{g}_j(\mathbf{X}) \tag{7.222}$$

where

$$\tilde{g}_j(\mathbf{X}) = \begin{cases} -\dfrac{1}{g_j(\mathbf{X})} & \text{if } g_j(\mathbf{X}) \le \varepsilon \\ -\dfrac{2\varepsilon - g_j(\mathbf{X})}{\varepsilon^2} & \text{if } g_j(\mathbf{X}) > \varepsilon \end{cases} \tag{7.223}$$

and ε is a small negative number that marks the transition from the interior penalty $[g_j(\mathbf{X}) \le \varepsilon]$ to the extended penalty $[g_j(\mathbf{X}) > \varepsilon]$. To produce a sequence of improved feasible designs, the value of ε is to be selected such that the function ϕ_k will have a positive slope at the constraint boundary. Usually, ε is chosen as

$$\varepsilon = -c(r_k)^a \tag{7.224}$$

Quadratic extended penalty function method

The ϕ_k function defined by Eq. (7.222) can be seen to be continuous with continuous first derivatives at $g_j(\mathbf{X}) = \varepsilon$. However, the second derivatives can be seen to be discontinuous at $g_j(\mathbf{X}) = \varepsilon$. Hence it is not possible to use a second-order method for unconstrained minimization [7.20]. The quadratic extended penalty function is defined so as to have continuous second derivatives at $g_j(\mathbf{X}) = \varepsilon$ as follows:

$$\phi_k = \phi(\mathbf{X}, r_k) = f(\mathbf{X}) + r_k \sum_{j=1}^{m} \tilde{g}_j(\mathbf{X}) \tag{7.225}$$

where

$$\tilde{g}_j(\mathbf{X}) = \begin{cases} -\dfrac{1}{g_j(\mathbf{X})} & \text{if} \quad g_j(\mathbf{X}) \le \varepsilon \\ \left\{ -\dfrac{1}{\varepsilon}\left[\dfrac{g_j(\mathbf{X})}{\varepsilon}\right]^2 - 3\dfrac{g_j(\mathbf{X})}{\varepsilon} + 3 \right\} & \text{if} \quad g_j(\mathbf{X}) > \varepsilon \end{cases} \tag{7.226}$$

Quadratic extended penalty function method

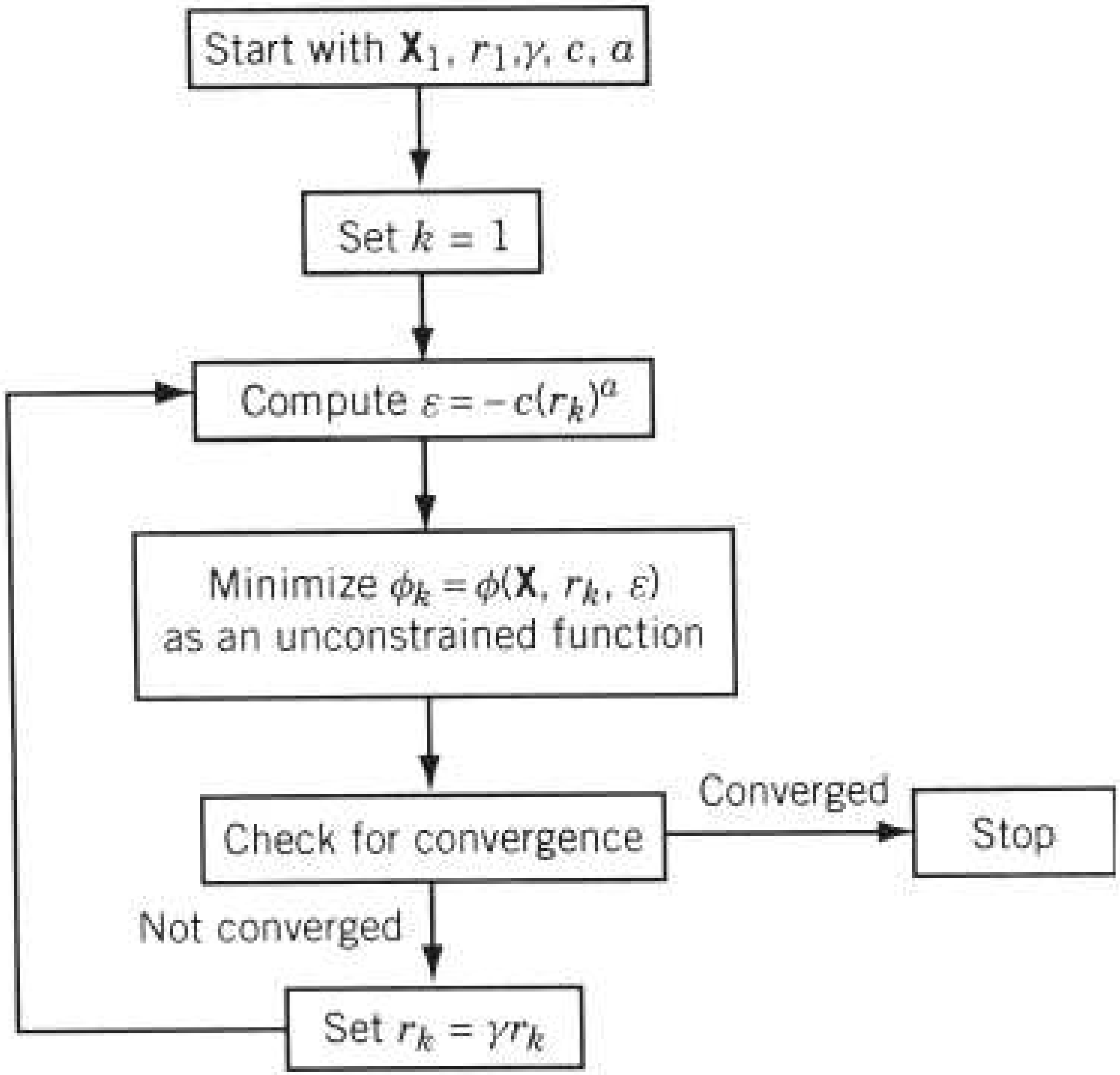


Figure 7.14 Linear extended penalty function method.