Nonlinear Programming

to accompany

Operations Research: Applications and Algorithms

4th edition, Chapter 11

by Wayne L. Winston

Review of Differential Calculus

- The equation: $\lim_{x \to a} f(x) = c$

  means that as x gets closer to a (but not equal to a), the value of $f(x)$ gets arbitrarily close to c.

- A function $f(x)$ is **continuous** at point a if $\lim_{x \to a} f(x) = f(a)$.

  If $f(x)$ is not continuous at $x=a$, we say that $f(x)$ is **discontinuous** (or has a discontinuity) at a.

- The **derivative** of a function $f(x)$ at $x = a$ (written $f'(a)$) is

  defined to be $\lim_{\Delta x \to 0} \frac{f(a+\Delta x)-f(a)}{\Delta x}$

  $f(a)$: the slope of $f(x)$ at $x=a$.
  If $f(a)>0$, then $f(x)$ is increasing at $x=a$.

- ***n*th-order derivatives, *n*th-order Taylor series expansion:** for $0 \leq h \leq b - a$ and some number p between a and $a+h$

$$f(a+h) = f(a) + \sum_{i=1}^{i=n} \frac{f^{(i)}(a)}{i!} h^i + \frac{f^{(n+1)}(p)}{(n+1)!} h^{n+1}$$

given that $f^{(n+1)}(x)$ exists for every point on interval $[a, b]$

- The partial derivative of $f(x_1, x_2, \ldots x_n)$ with respect to the variable x_i is written

$$\frac{\partial f}{\partial x_i} = \lim_{\Delta x_i \to 0} \frac{f(x_1, \ldots, x_i + \Delta x_i, \ldots, x_n) - f(x_1, \ldots, x_i, \ldots x_n)}{\Delta x_i}$$

- The second order partial derivatives: $\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i}$

1. Introduction to NLP

- A general **nonlinear programming problem** (NLP) can be expressed as follows:

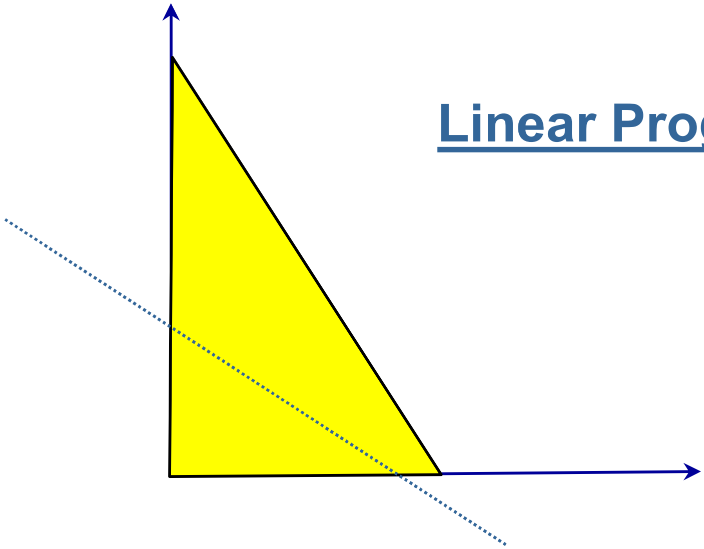
Find the values of decision variables $x_1, x_2, \ldots x_n$ that

$$
\begin{aligned}
\text{max} \quad & (\text{or min})\ z = f(x_1, x_2, \ldots, x_n) \\
\text{s.t.} \quad & g_1(x_1, x_2, \ldots, x_n)\ (\le, =, \text{or} \ge) b_1 \\
& g_2(x_1, x_2, \ldots, x_n)\ (\le, =, \text{or} \ge) b_2 \\
& \vdots \\
& g_m(x_1, x_2, \ldots, x_n)\ (\le, =, \text{or} \ge) b_m
\end{aligned}
$$

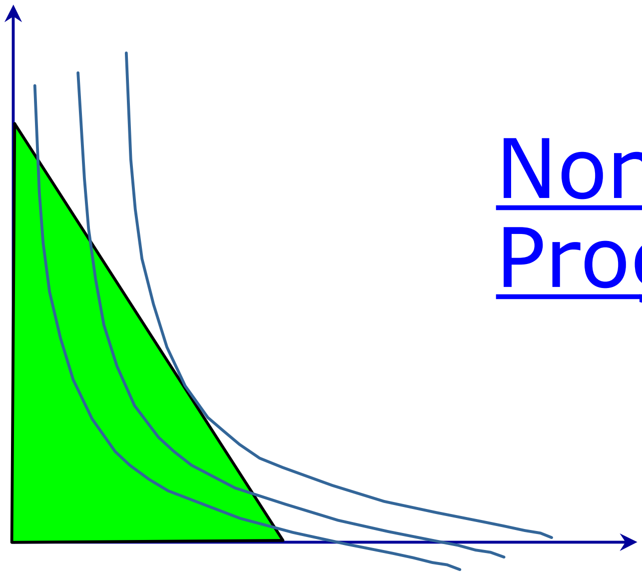
x_j: continuous variable

- As in linear programming, $f(x_1, x_2,\ldots,x_n)$ is the NLP's **objective function**, and $g_1(x_1, x_2,\ldots,x_n)$ ($\leq$, $=$, or $\geq$)b_1, ...$g_m(x_1, x_2,\ldots,x_n)$ ($\leq$, $=$, or $\geq$)b_m are the NLP's **constraints**.
- An NLP with no constraints is an **unconstrained NLP**.
- Unconstrained LP?
- The **feasible region** for NLP above is the set of points $(x_1, x_2,\ldots,x_n)$ that satisfy the m constraints in the NLP. A point in the feasible region is a *feasible point*, and a point that is not in the feasible region is an *infeasible point*.

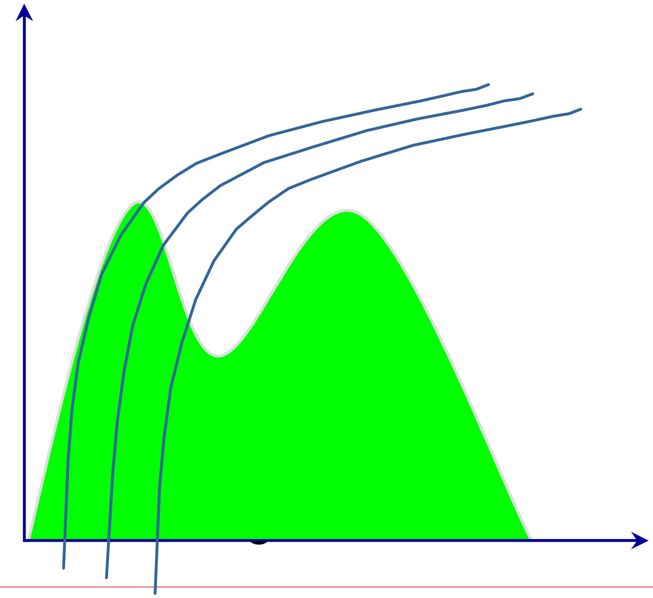
Difficulties of NLP Models



Linear Program:



Nonlinear Programs:



Example: Profit Maximization considering price-demand relation

- It costs \$$c$/unit to produce a product.
- Demand (demoted as D) is often modeled as a function of price (denoted as p). For example, $D(p) = 1 - bp$ (linear relation) for a parameter b. The function and its parameters can be determined by statistics regression.
- → nonlinear in profit: $(p - c) \cdot D(p)$
- To maximize profit, what is the price and how much should be produced (to satisfy the demand).
- NLP: $max \;\; z = (p - c)D(p)$
  s.t. $0 \le p \le 10$

- Excel file: some examples

Example: Tire Production

- Firerock produces rubber used for tires by combining three ingredients: rubber, oil, and carbon black.
- Costs (cents/pound): rubber (4), oil (1), carbon black (7).
- The rubber used in automobile tires must have
  - a hardness of between 25 and 35
  - an elasticity of at least 16
  - a tensile strength of at least 12
- To manufacture a set of four automobile tires, 100 pounds of product is needed.
- The rubber to make a set of tires must contain between 25 and 60 pounds of rubber and at least 50 pounds of carbon black.

Example (cont'd)

- Define decision variables:

  R = pounds of rubber in mixture used to produce four tires
  O = pounds of oil in mixture used to produce four tires
  C = pounds of carbon black used to produce four tires

- *Statistical analysis* has shown that the hardness, elasticity, and tensile strength of a 100-pound mixture of rubber, oil, and carbon black is

  Tensile Strength (TS) = $12.5 - .10(O) - .001(O)^2$
  Elasticity (E) = $17 + .35R - .04(O) - .002(O)^2$
  Hardness (H) = $34 + .10R + .06(O) - .3(C) + .001(R)(O) + .005(O)^2 + .001C^2$

- Formulate the NLP whose solution will tell Firerock how to minimize the cost of producing the rubber product needed to manufacture a set of automobile tires.

Example: continued

- Min $4R + O + 7C$

  s.t., $TS = 12.5 - .10(O) - .001(O)^2 >= 12$

  $E = 17 + .35R - .04(O) - .002(O)^2 >= 16$

  $H = 34 + .10R + .06(O) - .3(C) + .001(R)(O) + .005(O)^2 + .001C^2 >= 25$

  $H = 34 + .10R + .06(O) - .3(C) + .001(R)(O) + .005(O)^2 + .001C^2 <= 35$

  $R + O + C = 100$

  $R <= 60$

  $R >= 25$

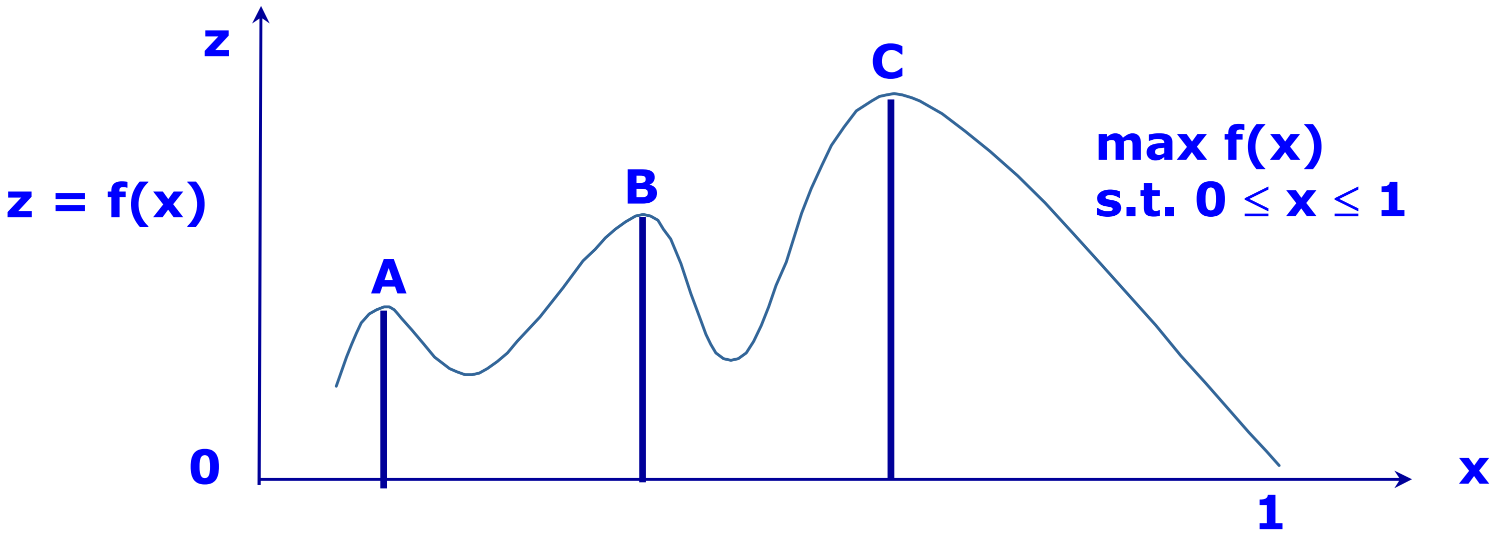
  $C >= 50$

  $O >= 0$

Local vs. Global Optima

Definition: Let x be a feasible solution, then

- x is a **global max** if $f(x) \geq f(y)$ for every feasible y.
- x is a **local max** if $f(x) \geq f(y)$ for every feasible y sufficiently close to x (i.e., $x_j - \varepsilon \geq y_j \geq x_j + \varepsilon$ for all j and some small ε).



There may be several locally optimal solutions.

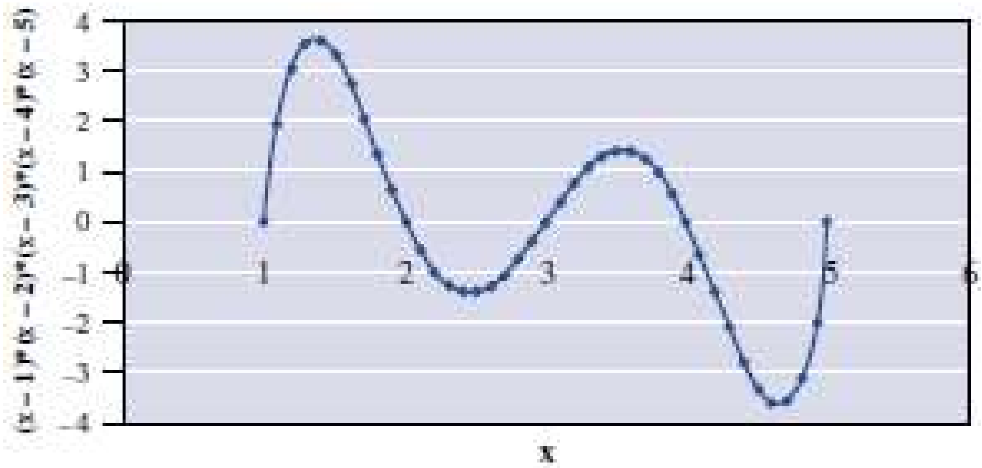
- If the NLP is a maximization problem, then any point $\bar{\mathbf{x}} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$ in the feasible region for which $f(\bar{\mathbf{x}}) \geq f(\mathbf{x})$ holds true for all points $\mathbf{x}$ in the feasible region is an **optimal solution** to the NLP.

- For any NLP (maximization), a feasible point $\mathbf{x}' = (x_1', x_2', \dots, x_n')$ is a **local maximum** if for sufficiently small ε, any feasible point $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ having $|x_i - x_1'| < \varepsilon$ for $i = 1,2,\ldots,n$ satisfies $f(\mathbf{x}') \geq f(\mathbf{x})$.

Relations of local and global optima

- For NLPs having multiple local optimal solutions, the Solver may fail to find the optimal solution because it may pick a local optima that is not a global optima.
- NLPs can be solved with LINGO or Excel Solver. However, in general, there is no guarantee that the solution found by them is optimal.

max $z=(x-1)(x-2)(x-3)(x-4)(x-5)$
s.t. $x>=1$
$x<=5$
Different initial values => ??
In LINGO: INIT:
x=2;
ENDINIT



- **When is a locally optimal solution also globally optimal**? Then, LINGO will find the optimal solution to an NLP. ---- Convexity

2 Convex and Concave Functions

- **Theorem**: Consider a general NLP. Suppose **the feasible region *S* for NLP is a convex set**. If $f(\mathbf{x})$ is **concave (convex)** on S, then any local maximum (minimum) for the NLP is an optimal solution (global optima) to the NLP.

Convex set and convex and concave functions

- Convex set: see (7-convexity-local and global optima.ppt) on convex set
- A function $f(x_1, x_2, \ldots, x_n)$ is a **convex function** on a convex set S if for any $\mathbf{x}' \in S$ and $\mathbf{x}'' \in S$
  $$f\,[c\mathbf{x}' + (1-c)\,\mathbf{x}''] \le cf(\mathbf{x}') + (1-c)f(\mathbf{x}'')$$
  holds for $0 \le c \le 1$.
- A function $f(x_1, x_2, \ldots, x_n)$ is a **concave function** on a convex set S if for any $\mathbf{x}' \in S$ and $\mathbf{x}'' \in S$
  $$f\,[c\mathbf{x}' + (1-c)\mathbf{x}''] \ge c\,f(\mathbf{x}') + (1-c)f(\mathbf{x}'')$$
  holds for $0 \le c \le 1$.

- A function $f(x_1, x_2, \ldots, x_n)$ is a convex function iff $-f(x_1, x_2, \ldots, x_n)$ is a concave function, and conversely.
- The sum of two convex functions is convex and the sum of two concave functions is concave.
- A linear function is both convex and concave.
- more …

- Suppose that $f(x)$ is a function of a single variable and $f''(x)$ exists for all x in a convex set S. Then $f(x)$ is a convex (concave) function of S if and only if $f''(x) \geq 0$ ($f''(x) \leq 0$) for all x in S. (single variable)

Suppose $f(x_1, x_2,\ldots, x_n)$ has continuous second-order partial derivatives for each point $\mathbf{x}=(x_1, x_2,\ldots, x_n)$ in a convex set S.

- $f(x_1, x_2,\ldots, x_n)$ is a **convex function** on S if and only if for each $x \in S$, all principal minors of H are non-negative.
- $f(x_1, x_2,\ldots, x_n)$ is a **concave function** on S if and only if for each $\mathbf{x} \in S$ and k=1, 2,…n, all nonzero principal minors have the same sign as $(-1)^k$.

✓ The Hessian of $f(x_1, x_2,\ldots, x_n)$ is the n x n matrix whose ijth entry is $\frac{\partial^2 f}{\partial x_i \partial x_j}$, denoted as $H(x_1, x_2,\ldots, x_n)$

✓ An ***i*th principal minor** of an n x n matrix is the determinant of any i x i matrix obtained by deleting $n - i$ rows and the corresponding $n - i$ columns of the matrix.

✓ The ***k*th leading principal minor** of an n x n matrix is the determinant of the k x k matrix obtained by deleting the last n-k rows and columns of the matrix. $H_k(x_1, x_2,\ldots, x_n)$ is the kth leading principal minor of the hessian matrix evaluated at the point $(x_1, x_2,\ldots, x_n)$.

Example 1

- $f(x_1, x_2) = x_1^3 + 2x_1x_2 + x_2^2$, then
  - $H(x_1x_2) = \begin{bmatrix} 6x_1 & 2 \\ 2 & 2 \end{bmatrix}$
  - Principal minors (*i*=1, 2): the first principle minors are $6x_1$ and 2, the second principle minor is the determinant of $H(x_1x_2)$, which is $12x_1 - 4$.
  - Leading principal minors (*k*=1, 2): $H_1(x_1x_2) = 6x_1$ and $H_2(x_1x_2) = 12x_1 - 4$.

Example 2

- Show that $f(x_1, x_2) = -x_1^2 - x_1x_2 - 2x_2^2$ is a concave function on R^2.
- We have $H(x_1x_2) = \begin{bmatrix} -2 & -1 \\ -1 & -4 \end{bmatrix}$
- Principal minors (i=1, 2): the first principle minors are -2 and -4. These are both nonpositive. The second principle minor is $7 > 0$. Thus, $f(x_1, x_2)$ is a concave function on R^2.

3 Unconstrained NLPs with Several Variables

- Consider this unconstrained NLP

$$\max\ (\,or \min)\, f(x_1, x_2, \ldots x_n)$$
$$\text{s.t. } (x_1, x_2, \ldots, x_n) \in R^n$$

- Assume that the first and second partial derivatives of $f(\mathbf{x})$ exist and are continuous at all points.

- A point $\bar{\mathbf{x}}$ having $\frac{\partial f(\bar{\mathbf{x}})}{\partial x_i} = 0$ for $i = 1, 2, \ldots, n$ is called a

  stationary point of f.

Single variable: stationary points

THEOREM 4

If $f'(x_0) = 0$ and $f''(x_0) < 0$, then x_0 is a local maximum. If $f'(x_0) = 0$ and $f''(x_0) > 0$, then x_0 is a local minimum.

THEOREM 5

If $f'(x_0) = 0$, and

1 If the first nonvanishing (nonzero) derivative at x_0 is an odd-order derivative [$f^{(3)}(x_0)$, $f^{(5)}(x_0)$, and so on], then x_0 is not a local maximum or a local minimum.

2 If the first nonvanishing derivative at x_0 is positive and an even-order derivative, then x_0 is a local minimum.

3 If the first nonvanishing derivative at x_0 is negative and an even-order derivative, then x_0 is a local maximum.

Multiple variables

- These theorems provide the basics of unconstrained NLP.
  - Necessary condition: If $\bar{\mathbf{x}}$ is a local optima, then $\frac{\partial f(\bar{\mathbf{x}})}{\partial x_i} = 0$ for $i = 1,2,\ldots,n$.
  - If $H_k(\bar{\mathbf{x}}) > 0$, k=1,2,…,n, then a stationary point $\bar{\mathbf{x}}$ is a local minimum.
  - If, for k=1,2,…,n, $H_k(\bar{\mathbf{x}})$ is nonzero and has the same sign as $(-1)^k$, then a stationary point $\bar{\mathbf{x}}$ is a local maximum.
  - If $H_n(\bar{\mathbf{x}}) \neq 0$ and the conditions of the previous two theorems do not hold, then a stationary point $\bar{\mathbf{x}}$ is not a local optima.
- If a stationary point $\mathbf{x}$ is not a local extremum, then it is called a **saddle point.**
- If $H_n(\mathbf{x})$=0 for a stationary point $\mathbf{x}$, then $\mathbf{x}$ may be a local minimum, a local maximum, or a saddle point, and the preceding tests are inconclusive.

Example 28

- Find all local maxima, local minima, and saddle points for

$$f(x_1, x_2) = x_1^2 x_2 + x_1 x_2^3 - x_1 x_2$$

4 The Method of Steepest Ascent

- The method of steepest ascent can be used to approximate a function's **stationary point** having $\nabla f(\mathbf{x}) = 0$ (candidates for optimal solutions).
- Given a vector $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in R^n$, the length of $\mathbf{x}$ (written $||x||$) is $||x|| = (x_1^2 + x_2^2 + \cdots + x_n^2)^{\frac{1}{2}}$
- For any vector $\mathbf{x}$, the unit vector $\mathbf{x}/||\mathbf{x}||$ is called the normalized version of $\mathbf{x}$.
- A **direction** can be represented by only one normalized vector.

- Consider a function $f(x_1, x_2, \ldots x_n)$, all of whose **partial derivatives exist at every point**.
- A **gradient vector** for $f(x_1, x_2, \ldots x_n)$, written $\nabla f(\mathbf{x})$, is

$$\nabla f(x) = \left[\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \cdots, \frac{\partial f(x)}{\partial x_n}\right]$$

- Suppose we are at a point **v** and we move from **v** a small distance δ in a direction **d**. Then for a given δ, the maximal increase in the value of $f(\mathbf{x})$ will occur if we choose

$$\mathbf{d} = \frac{\nabla f(\mathbf{v})}{||\nabla f(\mathbf{v})||}$$

- In other words, if we move a small distance away from **v** and we want $f(\mathbf{x})$ to increase as quickly as possible, then we should move in the direction of $\nabla f(\mathbf{v})$.

Procedure of steepest ascent method

- Begin at any point $\mathbf{v}_0$, and then move in the direction of $\nabla f(\mathbf{v}_0)$, a maximum rate of increase for f at $\mathbf{v}_0$. For some nonnegative value of t_0, we move to a point $\mathbf{v}_1 = \mathbf{v}_0 + t_0 \nabla f(\mathbf{v}_0)$.
- t_0 solves the following one-dimensional optimization problem:

$$\begin{aligned} &\max \quad f(\mathbf{v}_0 + t_0 \nabla f(\mathbf{v}_0)) \\ &s.t., \qquad t_0 \geq 0 \end{aligned}$$

  This single-variable NLP may be solved by the methods using differentials or, if necessary, by a search procedure such as the Golden Section Search.

- If $\|\nabla f(\mathbf{v}_1)\|$ is sufficiently small (say, less than 0.01) (**termination condition**), we may terminate the algorithm with the knowledge that $\mathbf{v}_1$ is near a stationary point $\mathbf{v}'$ having $\nabla f(\mathbf{v}') = 0$.

Example 29:

- Use the method of steepest ascent to approximate the solution to

$$\max\ z = -(x_1 - 3)^2 - (x_2 - 2)^2 = f(x_1, x_2)$$
$$s.t.,\ \ (x_1, x_2) \in R^2$$

We arbitrarily choose to begin at the point $\mathbf{v}_0 = (1, 1)$. Because $\nabla f(\mathbf{x}_1, \mathbf{x}_2) = (-2(x_1 - 3), -2(x_2 - 2))$, we have $\nabla f(1, 1) = (4, 2)$. Thus, we must choose t_0 to maximize

$$f(t_0) = f[(1, 1) + t_0(4, 2)] = f(1 + 4t_0, 1 + 2t_0) = -(-2 + 4t_0)^2 - (-1 + 2t_0)^2$$

Setting $f'(t_0) = 0$, we obtain

$$\begin{aligned} -8(-2 + 4t_0) - 4(-1 + 2t_0) &= 0 \\ 20 - 40t_0 &= 0 \\ t_0 &= 0.5 \end{aligned}$$

Our new point is $\mathbf{v}_1 = (1, 1) + 0.5(4, 2) = (3, 2)$. Now $\nabla f(3, 2) = (0, 0)$, and we terminate the algorithm. Because $f(x_1, x_2)$ is a concave function, we have found the optimal solution to the NLP.

5 Constrained NLP – KKT Conditions

- A general NLP:

$$\begin{aligned}
&\text{max (or min)} \quad f(x_1, x_2, \dots, x_n) \\
&\text{s.t. } g_i(x_1, x_2, \dots, x_n) \le 0 \text{ for } i = 1,2, \dots, q \\
&\quad g_i(x_1, x_2, \dots, x_n) = 0 \text{ for } i = q+1, q+2, \dots, m \\
&\quad x_j \ge \mathbf{0}, \le \mathbf{0}, \textbf{ or unrestricted} \text{ for } j = 1,2, \dots, n
\end{aligned}$$

- Associate multipliers $\lambda_1, \lambda_2, \dots, \lambda_m$ with the constraints
- Construct Lagrangian function as

$$\begin{aligned}
&L(x_1, x_2, \dots, x_n, \lambda_1, \lambda_2, \dots, \lambda_m) \\
&= f(x_1, x_2, \dots, x_n) - \sum_{i=1}^{m} \lambda_i g_i(x_1, x_2, \dots, x_n)
\end{aligned}$$

- The **KKT conditions** are **necessary** for a feasible point $\bar{\mathbf{x}} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$ to solve the NLP. (Of course, meanwhile satisfying all the original constraints and sign restrictions).

5.1 The (Karush)-Kuhn-Tucker Conditions

- KKT conditions:

$$\frac{\partial L}{\partial x_j} = \frac{\partial f(\bar{\mathbf{x}})}{\partial x_j} - \sum_{i=1}^{m} \bar{\lambda}_i \frac{\partial g_i(\bar{\mathbf{x}})}{\partial x_j} (\leq,\ \geq,\ \text{or} =)\ 0 \quad (j = 1,2,\ldots,n)$$

$\bar{\lambda}_i \cdot g_i(\bar{\mathbf{x}}) = 0\ \ (i = 1,2,\ldots,q)$ <u>complementary conditions</u>

$$\bar{x}_j \cdot \frac{\partial L}{\partial x_j} = 0 \qquad (j = 1,2,\ldots,n)$$

$\bar{\lambda}_i$ (≥ 0 (max NLP), ≤ 0 (min NLP)) $\ (i = 1,2,\ldots,q)$

$\bar{\lambda}_i$ unrestricted $\ (i = q+1,\ldots,m)$

		Max NLP	Min NLP
Variable	$x_j \geq 0$	$\partial L/\partial x_j \leq 0$	$\partial L/\partial x_j \geq 0$
	$x_j \leq 0$	$\partial L/\partial x_j \geq 0$	$\partial L/\partial x_j \leq 0$
	x_j unrestricted	$\partial L/\partial x_j = 0$	$\partial L/\partial x_j = 0$

5.2 Sufficient conditions

- Consider a maximization (minimization) NLP as in the proceeding page 27. If $f(\mathbf{x})$ is a concave (convex) function and the feasible region formed by all the constraints is convex set, then any feasible point $\bar{\mathbf{x}}$ satisfying the necessary KKT conditions is an optimal solution.
- The feasible region defined by $g_i(\mathbf{x}) \leq 0$ is convex set if $g_i(\mathbf{x})$ is a convex function.
- If all the constraints are defined by convex functions in terms of $\leq$ direction, the feasible region is convex set.
- The feasible region defined by linear constraint is convex set.

5.3 Special NLP – 1

$$\begin{aligned} &\max\,(or\ \min)\, f(x) \\ \text{s.t.}\quad & g_1(x_1, x_2, \ldots, x_n) = b_1 \\ & g_2(x_1, x_2, \ldots, x_n) = b_2 \\ & \vdots \\ & g_m(x_1, x_2, \ldots, x_n) = b_m \end{aligned}$$

- Simplify the KKT conditions for the following NLPs in which all the constraints are equality constraints and all variables are unrestricted.

- The KKT conditions: $\frac{\partial L}{\partial x_j} = \frac{\partial f(\bar{\mathbf{x}})}{\partial x_j} + \sum_{i=1}^{i=m} \bar{\lambda}_i \frac{\partial g_i(\bar{\mathbf{x}})}{\partial x_j} = 0$

- The original constraints: $\frac{\partial L}{\partial \lambda_i} = g_i(\bar{\mathbf{x}}) - b_i = 0$

- A point $(x_1, x_2, \ldots, x_n, \lambda_1, \lambda_2, \ldots, \lambda_m)$ that maximizes (minimizes) $L(x_1, x_2, \ldots, x_n, \lambda_1, \lambda_2, \ldots, \lambda_m)$ must satisfy

$$\frac{\partial L}{\partial x_1} = \frac{\partial L}{\partial x_2} = \cdots = \frac{\partial L}{\partial x_n} = \frac{\partial L}{\partial \lambda_1} = \frac{\partial L}{\partial \lambda_2} = \cdots = \frac{\partial L}{\partial \lambda_m} = 0$$

Example 30

- A company is planning to spend \$10,000 on advertising. It costs \$3,000 per minute to advertise on television and \$1,000 per minute to advertise on radio. If the firm buys x minutes of television advertising and y minutes of radio advertising, then its revenue in thousands of dollars is given by $f(x,y) = -2x^2 - y^2 + xy + 8x + 3y$.
- How can the firm maximize its revenue?

Special NLP – 2

- The (Karush-)Kuhn-Tucker conditions are used to solve NLPs **(26)**:

$$\begin{aligned}
&\max(\,or\ \min)\, f(x_1, x_2, \ldots, x_n) \\
&\text{s.t.}\ \ g_1(x_1, x_2, \ldots, x_n) \le b_1 \\
&\qquad g_2(x_1, x_2, \ldots, x_n) \le b_2 \\
&\qquad\qquad \vdots \\
&\qquad g_m(x_1, x_2, \ldots, x_n) \le b_m
\end{aligned}$$

- Associate multipliers $\lambda_1, \lambda_2, \ldots, \lambda_m$ with the constraints
- The **Kuhn-Tucker conditions** are necessary for a point $\bar{\mathbf{x}} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$ to solve the NLP.

KKT necessary conditions for NLP (26)

- If $\bar{\mathbf{x}} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$ is an optimal solution to NLP, then $\bar{\mathbf{x}}$ must **satisfy the *m* constraints** in the NLP, and there must exist **multipliers** $\lambda_1, \lambda_2, \ldots, \lambda_m$ **satisfying**
  - for the maximization NLP:

$$\frac{\partial f(\bar{x})}{\partial x_j} - \sum_{i=1}^{i=m} \bar{\lambda}_i \frac{\partial g_i(\bar{x})}{\partial x_j} = 0 \qquad (j = 1, 2, \ldots, n)$$

$$\bar{\lambda}_i [b_i - g_i(\bar{x})] = 0 \qquad (i = 1, 2, \ldots, m)$$

$$\bar{\lambda}_i \geq 0 \qquad (i = 1, 2, \ldots, m)$$

  - for the minimization NLP:

$$\frac{\partial f(\bar{x})}{\partial x_j} + \sum_{i=1}^{i=m} \bar{\lambda}_i \frac{\partial g_i(\bar{x})}{\partial x_j} = 0 \qquad (j = 1, 2, \ldots, n)$$

$$\bar{\lambda}_i [b_i - g_i(\bar{x})] = 0 \qquad (i = 1, 2, \ldots, m)$$

$$\bar{\lambda}_i \geq 0 \qquad (i = 1, 2, \ldots, m)$$

Special NLP – 3

- NLPs **(30)** in which the variables are nonnegative:

$$
\begin{aligned}
\max \text{ (or } \min) \ z &= f(x_1, x_2, \ldots, x_n) \\
\text{s.t.} \quad g_1(x_1, x_2, \ldots, x_n) &\le b_1 \\
g_2(x_1, x_2, \ldots, x_n) &\le b_2 \\
&\vdots \\
g_m(x_1, x_2, \ldots, x_n) &\le b_m \\
-x_1 &\le 0 \\
-x_2 &\le 0 \\
&\vdots \\
-x_n &\le 0
\end{aligned}
$$

- If $\bar{\mathbf{x}} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$ is an optimal solution to NLP, then $\bar{\mathbf{x}}$ must **satisfy the *m* constraints and sign restrictions** in the NLP, and there must exist **multipliers** $\lambda_1, \lambda_2, \ldots, \lambda_m$ **satisfying the KKT conditions as below.**

KKT Necessary conditions for NLP (30)

- for the maximization NLP:

$$\frac{\partial f(\bar{x})}{\partial x_j} - \sum_{i=1}^{i=m} \bar{\lambda}_i \frac{\partial g_i(\bar{x})}{\partial x_j} \leq 0 \ (j = 1,2,\dots,n)$$

$$\bar{\lambda}_i[b_i - g_i(\bar{x})] = 0 \ (i = 1,2,\dots,m)$$

$$\left[\frac{\partial f(\bar{x})}{\partial x_j} - \sum_{i=1}^{i=m} \bar{\lambda}_i \frac{\partial g_i(\bar{x})}{\partial x_j}\right] \bar{x}_j = 0 \ \ (j = 1,2,\dots,n)$$

$$\bar{\lambda}_i \geq 0 \quad (i = 1,2,\dots,m)$$

- for the minimization NLP:

$$\frac{\partial f(\bar{x})}{\partial x_j} + \sum_{i=1}^{i=m} \bar{\lambda}_i \frac{\partial g_i(\bar{x})}{\partial x_j} \geq 0 \ \ (j = 1,2,\dots,n)$$

$$\bar{\lambda}_i[b_i - g_i(\bar{x})] = 0 \ \ (i = 1,2,\dots,m)$$

$$\left[\frac{\partial f(\bar{x})}{\partial x_j} + \sum_{i=1}^{i=m} \bar{\lambda}_i \frac{\partial g_i(\bar{x})}{\partial x_j}\right] \bar{x}_j = 0 \ \ (j = 1,2,\dots,n)$$

$$\bar{\lambda}_i \geq 0 \quad (i = 1,2,\dots,m)$$

KKT Necessary conditions for NLP (30)

- for the maximization NLP:

$$\frac{\partial f(\bar{x})}{\partial x_j} - \sum_{i=1}^{i=m} \bar{\lambda}_i \frac{\partial g_i(\bar{x})}{\partial x_j} + \bar{\mu}_j = 0 \quad (j = 1, 2, \ldots, n)$$

$$\bar{\lambda}_i [b_i - g_i(\bar{x})] = 0 \quad (i = 1, 2, \ldots, m)$$

$$\left[\frac{\partial f(\bar{x})}{\partial x_j} - \sum_{i=1}^{i=m} \bar{\lambda}_i \frac{\partial g_i(\bar{x})}{\partial x_j}\right] \bar{x}_j = 0 \quad (j = 1, 2, \ldots, n)$$

$$\bar{\lambda}_i \geq 0 \quad (i = 1, 2, \ldots, m)$$

$$\bar{\mu}_j \geq 0 \quad (j = 1, 2, \ldots, n)$$

Because $\bar{\mu}_j \geq 0$, equivalently

$$\frac{\partial f(\bar{x})}{\partial x_j} - \sum_{i=1}^{i=m} \bar{\lambda}_i \frac{\partial g_i(\bar{x})}{\partial x_j} \leq 0 \quad (j = 1, 2, \ldots, n)$$

$$\bar{\lambda}_i [b_i - g_i(\bar{x})] = 0 \quad (i = 1, 2, \ldots, m)$$

$$\left[\frac{\partial f(\bar{x})}{\partial x_j} - \sum_{i=1}^{i=m} \bar{\lambda}_i \frac{\partial g_i(\bar{x})}{\partial x_j}\right] \bar{x}_j = 0 \quad (j = 1, 2, \ldots, n)$$

$$\bar{\lambda}_i \geq 0 \quad (i = 1, 2, \ldots, m)$$

KKT Necessary conditions for NLP (30)

- for the minimization NLP:

$$\frac{\partial f(\bar{x})}{\partial x_j} + \sum_{i=1}^{i=m} \bar{\lambda}_i \frac{\partial g_i(\bar{x})}{\partial x_j} - \bar{\mu}_j = 0 \quad (j = 1, 2, \dots, n)$$

$$\bar{\lambda}_i [b_i - g_i(\bar{x})] = 0 \quad (i = 1, 2, \dots, m)$$

$$\left[\frac{\partial f(\bar{x})}{\partial x_j} + \sum_{i=1}^{i=m} \bar{\lambda}_i \frac{\partial g_i(\bar{x})}{\partial x_j} \right] \bar{x}_j = 0 \quad (j = 1, 2, \dots, n)$$

$$\bar{\lambda}_i \geq 0 \quad (i = 1, 2, \dots, m)$$

$$\bar{\mu}_j \geq 0 \quad (j = 1, 2, \dots, n)$$

Because $\bar{\mu}_j \geq 0$, equivalently

$$\frac{\partial f(\bar{x})}{\partial x_j} + \sum_{i=1}^{i=m} \bar{\lambda}_i \frac{\partial g_i(\bar{x})}{\partial x_j} \geq 0 \quad (j = 1, 2, \dots, n)$$

$$\bar{\lambda}_i [b_i - g_i(\bar{x})] = 0 \quad (i = 1, 2, \dots, m)$$

$$\left[\frac{\partial f(\bar{x})}{\partial x_j} + \sum_{i=1}^{i=m} \bar{\lambda}_i \frac{\partial g_i(\bar{x})}{\partial x_j} \right] \bar{x}_j = 0 \quad (j = 1, 2, \dots, n)$$

$$\bar{\lambda}_i \geq 0 \quad (i = 1, 2, \dots, m)$$

5.4 Geometrical interpretation of KKT conditions

- Three KKT conditions for (26) hold at a point x if and only if ∇f is a linear combination of $\nabla g_1, \nabla g_2, \ldots, \nabla g_m$, and the weight multiplying ∇g_i in this linear combination equals 0 if the *i*th constraint in (26) is nonbinding.

FIGURE 43 Example of Kuhn-Tucker Conditions: Both Constraints Binding

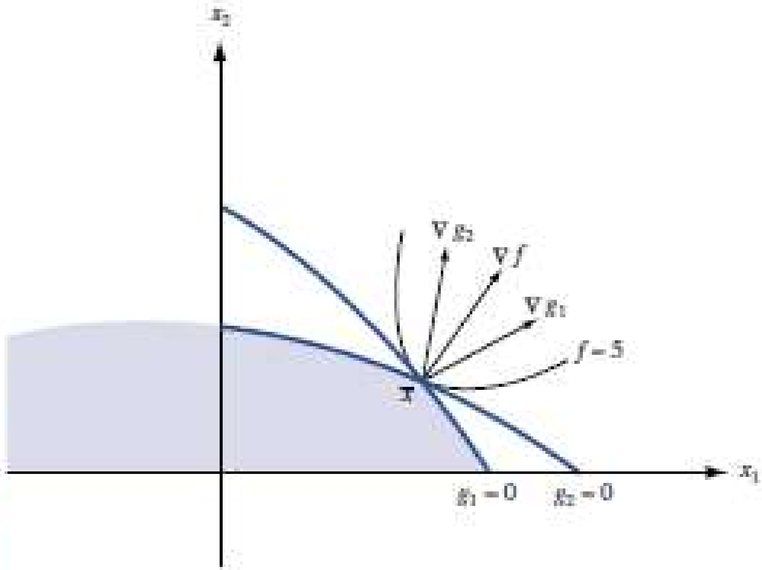
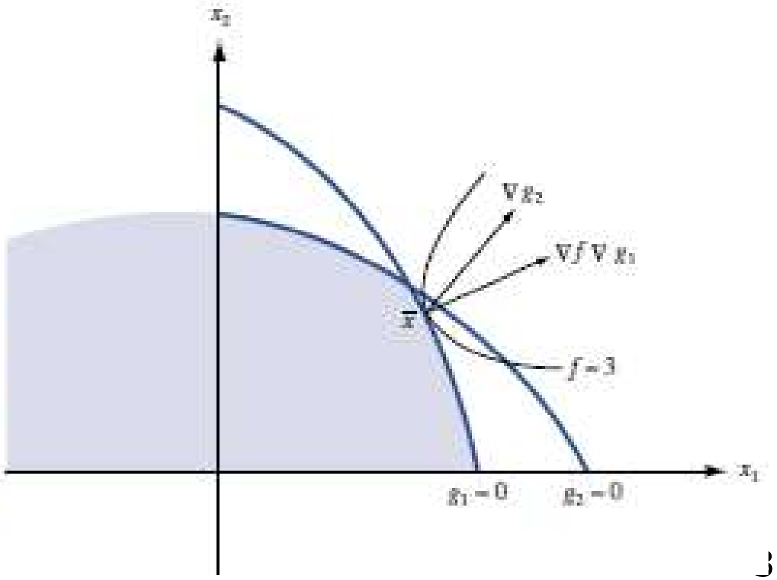


FIGURE 44 Example of Kuhn-Tucker Conditions: One Constraint Binding and One Constraint Nonbinding



Example 33

- A monopolist can purchase up to 17.25 oz of a chemical for \$10/oz. At a cost of \$3/oz, the chemical can be processed into an ounce of product 1; or, at a cost of \$5/oz, the chemical can be processed into an ounce of product 2. If x_1 oz of product 1 are produced, it sells for a price of \$$(30 - x_1)$ per ounce. If x_2 oz of product 2 are produced, it sells for a price of \$$(50 - 2x_2)$ per ounce. Determine how the monopolist can maximize profits.
- Solution. Let
  - $x1$=ounces of product 1 produced
  - $x2$=ounces of product 2 produced
  - $x3$=ounces of chemical processed

5.5 Constraint Qualifications

- For the theorems in this section to hold, the functions g_1, g_2, …, g_m must satisfy certain regularity conditions (constraint qualifications).
- Unless a constraint qualification or regularity condition is satisfied at an optimal point $\bar{\mathbf{x}}$, the Kuhn-Tucker conditions may fail to hold at $\bar{\mathbf{x}}$.
- When the constraints are linear, these regularity assumptions are always satisfied.
- One constraint qualification - **linear independence CQ:** If all g_i are continuous, and the gradients of all binding constraints (including any binding nonnegativity constraints) at optimal solution $\mathbf{x}$ form a set of linearly independent vectors, then the KKT conditions must hold at $\mathbf{x}$.

- A set of vectors is said to be **linearly dependent** if one of the vectors in the set can be defined as a linear combination of the others; if no vector in the set can be written in this way, then the vectors are said to be **linearly independent**.

- Linear Dependent: $a_1V_1+a_2V_2+\ldots+a_kV_k=0$ for a_i not all zero.

Example 34: Necessity of Constraint Qualification

Show that the Kuhn–Tucker conditions fail to hold at the optimal solution to the following NLP:

$$
\begin{aligned}
\max z &= x_1 \\
\text{s.t.} \quad & x_2 - (1 - x_1)^3 \leq 0 \qquad (56)\\
& x_1 \geq 0, x_2 \geq 0
\end{aligned}
$$

Solution If $x_1 > 1$, then the first constraint in (56) implies that $x_2 < 0$. Thus, the optimal z-value for (56) cannot exceed 1. Because $x_1 = 1$ and $x_2 = 0$ is feasible and yields $z = 1$, (1, 0) must be the optimal solution to NLP (56).

From Theorem 10, the following are two of the Kuhn–Tucker conditions for (56).

$$1 + 3\lambda_1(1 - x_1)^2 = -\mu_1 \qquad (57)$$

$$\mu_1 \geq 0 \qquad (58)$$

At the optimal solution (1, 0), (57) implies $\mu_1 = -1$, which contradicts (58). Thus, the Kuhn–Tucker conditions are not satisfied at (1, 0). We now show that at the point (1, 0) the Linear Independence Constraint Qualification is violated. At (1, 0) the constraints $x_2 - (1 - x_1)^3 \leq 0$ and $x_2 \geq 0$ are binding. Then

$$\nabla(x_2 - (1 - x_1)^3) = [0, 1]$$

$$\nabla(-x_2) = [0, -1]$$

Because $[0, 1] + [0, -1] = [0, 0]$, these gradients are linearly dependent. Thus, at (1,0) the gradients of the binding constraints are linearly dependent, and the constraint qualification is not satisfied.

LINGO

- If LINGO displays the message DUAL CONDITIONS:SATISFIED then you know it has found the point satisfying the Kuhn-Tucker conditions. Unless satisfying the sufficient conditions, LINGO might return a solution that is not optimal. Use different initial solutions to test the optimality.

6 The Method of Feasible Directions

- This method, a modification of the steepest ascent method, can be used to solve the NLP with linear constraints.

$$\begin{aligned} \max\ & z = f(\mathbf{x}) \\ \text{s.t.}\ & A\mathbf{x} \le \mathbf{b} \\ & \mathbf{x} \ge \mathbf{0} \end{aligned}$$

- To solve, begin with a feasible solution $\mathbf{x}^0$ (perhaps by using the two-phase simplex algorithm).
- Next, find a direction to move away from $\mathbf{x}^0$, which makes the new solution **remain feasible** and increase the value of z.
- Let $\mathbf{d}^0$ be a solution to the following LP:

$$\begin{aligned} \max\ & z = \nabla f(\mathbf{x}^0) \cdot \mathbf{d} \\ \text{s.t.}\ & A\mathbf{d} \le \mathbf{b} \\ & \mathbf{d} \ge \mathbf{0} \end{aligned}$$

- Choose our new point $\mathbf{x}^1$ to be $\mathbf{x}^1 = \mathbf{x}^0 + t^0(\mathbf{d}^0 - \mathbf{x}^0)$, where t^0 solves

$$\max \ f(\mathbf{x}^0 + t^0(\mathbf{d}^0 - \mathbf{x}^0))$$
$$s.t., \ 0 \leq t^0 \leq 1$$

  It is an NLP with a single variable.

- Continue in this fashion and generate directions of movement $\mathbf{d}^1$, $\mathbf{d}^2$, …, $\mathbf{d}^{n-1}$ and new points $\mathbf{x}^2$, $\mathbf{x}^3$, …, $\mathbf{x}^n$.
- We terminate the algorithm if $\mathbf{x}^k=\mathbf{x}^{k-1}$ or successive points are sufficiently close together. Return $\mathbf{x}^{k-1}$ as the solution to NLP.

Example 37

Perform two iterations of the feasible directions method on the following NLP:

$$\max z = f(x, y) = 2xy + 4x + 6y - 2x^2 - 2y^2$$
$$\text{s.t.} \quad x + y \leq 2$$
$$x, y \geq 0$$

Begin at the point (0,0).

7 Quadratic Programming

- A **quadratic programming** (QP) is an NLP in which each term in the objective function is of degree 2,1, or 0 (quadratic function) and all constraints are linear.
- LINGO, Excel and **Wolfe's method (**a modified version of Phase I of the two-phase simplex to find a point satisfying the KKT conditions**)** may be used to solve QP problems.
- Wolfe's method is guaranteed to obtain the optimal solution to a QP if all leading principal minors of the objective function's Hessian are positive (positive definite).
- In practice, the method of complementary pivoting is most often used to solve QPs (Shapiro, 1979).

Example 35: Portfolio Optimization

- I have $1,000 to invest in three stocks. Let Si be the random variable representing the annual return on $1 invested in stock i. Thus, if Si = 0.12, $1 invested in stock i at the beginning of a year was worth $1.12 at the end of the year. We are given the following information:
- E(S1) =0.14, E(S2)= 0.11, E(S3)= 0.10, var(S1)=0.20, var(S2)=0.08, var(S3)=0.18, cov (S1,S2) = 0.05, cov(S1,S3)=0.02, cov(S2,S3)=0.03.
- Formulate a QP that can be used to find the portfolio that attains an expected annual return of at least 12% and minimizes the variance of the annual dollar return on the portfolio.

8 Unconstrainted NLPs with One Variable

$$\max\,(or\ \min)\, f(x)$$
$$\text{s.t.} \qquad x \in [a,b]$$

- There are three types of points for which the NLP can have a local maximum or minimum (these points are often called *extremum candidates*).
  - Points where $a<x<b$, $f'(x) = 0$ (called a **stationary point** of $f(x)$)
  - Points where $f'(x)$ does not exist
  - Boundary points (endpoints) a and b of the interval $[a,b]$
- To find the optimal solution for the NLP, find all the local optima. The optimal solution is the local maximum (or minimum) having the largest (or smallest) value of $f(x)$.

Case 1: stationary points

THEOREM 4

If $f'(x_0) = 0$ and $f''(x_0) < 0$, then x_0 is a local maximum. If $f'(x_0) = 0$ and $f''(x_0) > 0$, then x_0 is a local minimum.

THEOREM 5

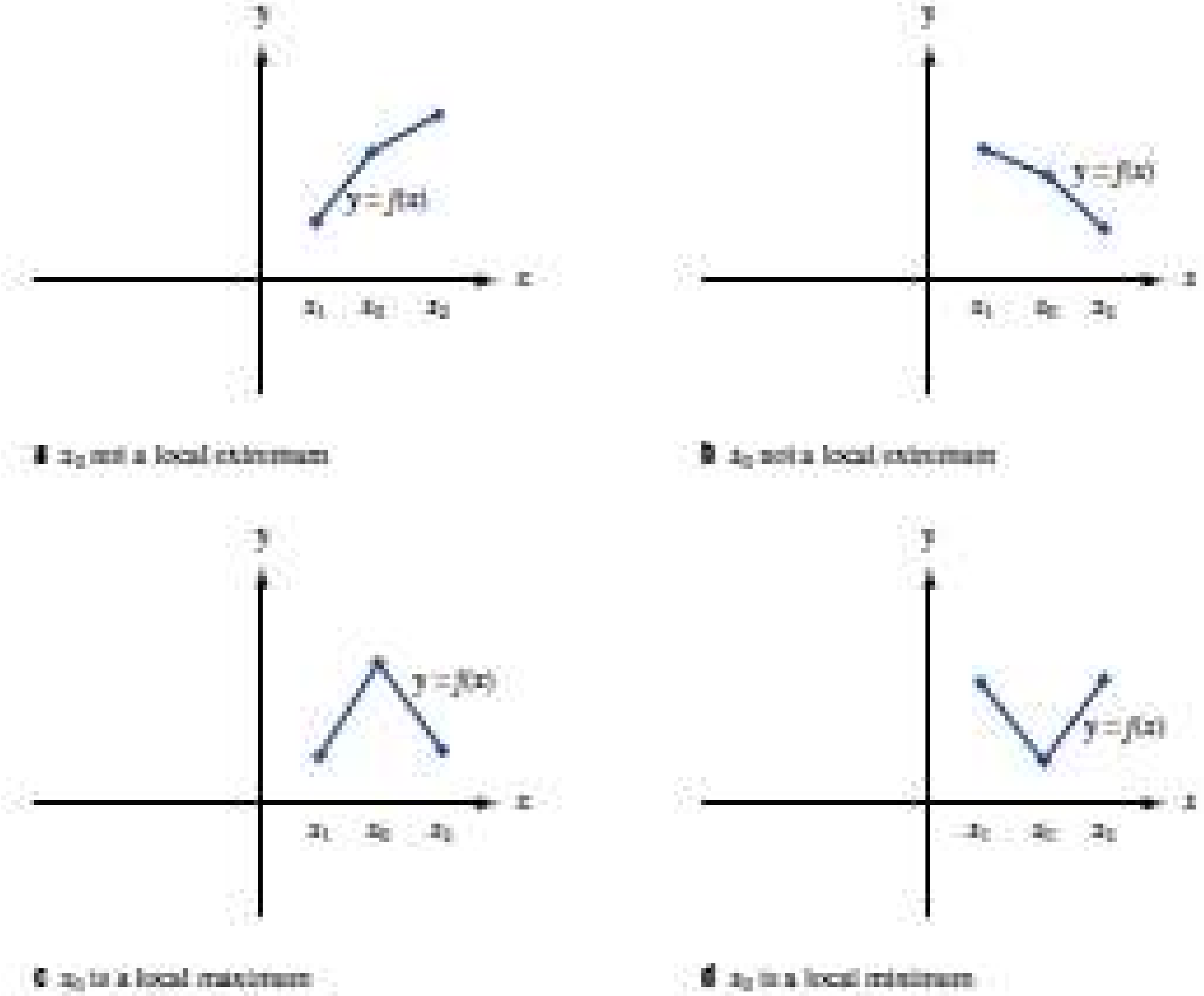
If $f'(x_0) = 0$, and

1 If the first nonvanishing (nonzero) derivative at x_0 is an odd-order derivative [$f^{(3)}(x_0)$, $f^{(5)}(x_0)$, and so on], then x_0 is not a local maximum or a local minimum.

2 If the first nonvanishing derivative at x_0 is positive and an even-order derivative, then x_0 is a local minimum.

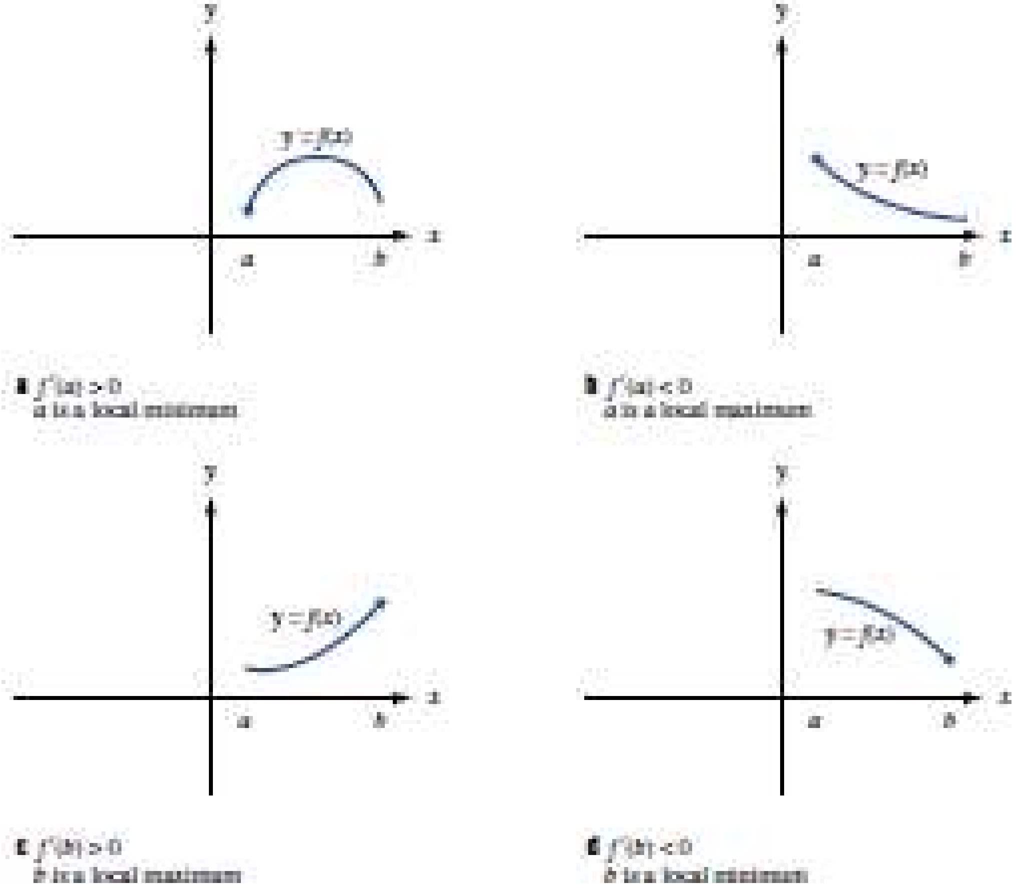
3 If the first nonvanishing derivative at x_0 is negative and an even-order derivative, then x_0 is a local maximum.

Case 2: non-differentiable points



Case 3: endpoints

- If $f'(a) > 0$, then a is a local minimum.
- If $f'(a) < 0$, then a is a local maximum.
- If $f'(b) > 0$, then b is a local maximum.
- If $f'(b) < 0$, then b is a local minimum.
- If $f'(a) = 0$ or $f'(b) = 0$, evaluate $f(x)$ at some point $a < x < b$ sufficient close to a or b.



Example: Profit Maximization by Monopolist

- It costs a monopolist \$5/unit to produce a product.
- If he produces x units of the product, then each can be sold for 10-x dollars.
- To maximize profit, how much should the monopolist produce.

- Note that: Demand (demoted as d) is often modeled as a function of price (denoted as p). For example, $d = 1 - bp$ (linear relation) for a parameter b. → nonlinear in revenue and profit.

Example: Solution

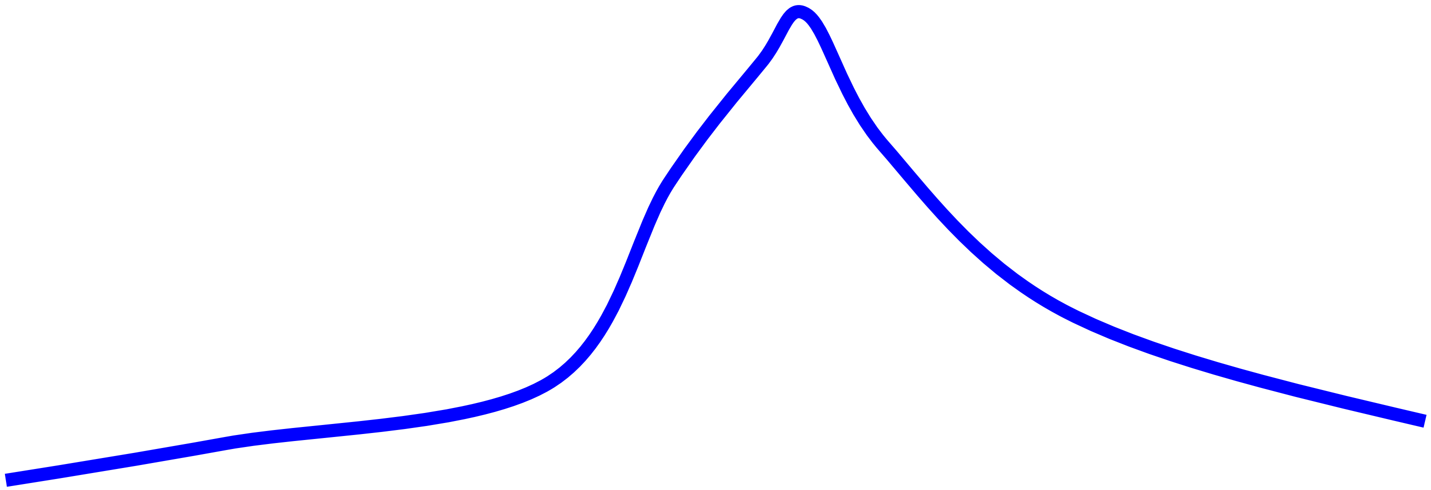
- The monopolist wants to solve the NLP

$$\max P(x) = x(10 - x) - 5x = 5x - x^2$$
$$\text{s.t.} \quad 0 \le x \le 10$$

- The extremum candidates can be classified as
  - Case 1 check tells us x=2.5 is a local maximum yielding a profit P(2.5)=6.25.
  - $P'(x)$ exists for all points in [0,10], so there are no Case 2 candidates.
  - $a = 0$ has $P'(0) = 5 > 0$ so $a = 0$ is a local minimum; b=10 has $P'(10)$=-15<0, so $b = 10$ is a local minimum

8.1 Local Search

- The some numerical methods can be used if the function is a unimodal function.
- A function *f(x)* is **unimodel** on [*a*,*b*] if for some point $\bar{x}$ on [*a*,*b*], *f(x)* is strictly increasing on $[a,\bar{x}]$ and strictly decreasing on $[\bar{x},b]$.
- Not necessary concave or even f'(x) may not exist
- A single variable function is **unimodal** if there is at most one local maximum (or at most one local minimum) .



- The optimal solution of the NLP is some point on the interval [a,b]. By evaluating $f(x)$ at two points x_1 and x_2 on $[a,b]$, we may reduce the size of the interval in which the solution to the NLP must lie.
- After evaluating $f(x_1)$ and $f(x_2)$, one of these cases must occur. It can be shown in each case that the optimal solution will lie in a subset of $[a,b]$.
  - Case 1: $f(x_1) < f(x_2)$ and $\bar{x} \in (x_1, b]$
  - Case 2: $f(x_1) = f(x_2)$ and $\bar{x} \in (a, x_2]$
  - Case 3: $f(x_1) > f(x_2)$ and $\bar{x} \in (a, x_2]$
- The interval in which *x-bar* must lie – either $[a,x_2)$ or $(x_1, b]$ – is called the **interval of uncertainty**.

The three cases:

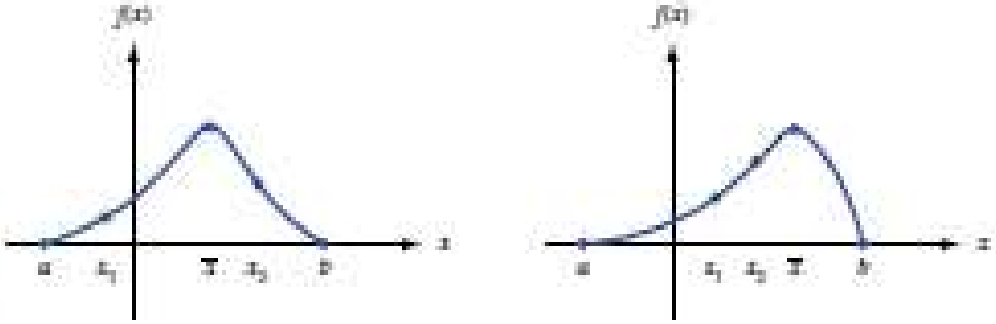


FIGURE 33
If $f(x_1) < f(x_2)$, $\bar{x} \in (x_1, b]$

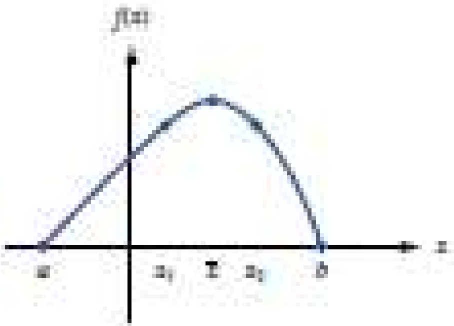


FIGURE 34
If $f(x_1) = f(x_2)$, $\bar{x} \in [a, x_2)$

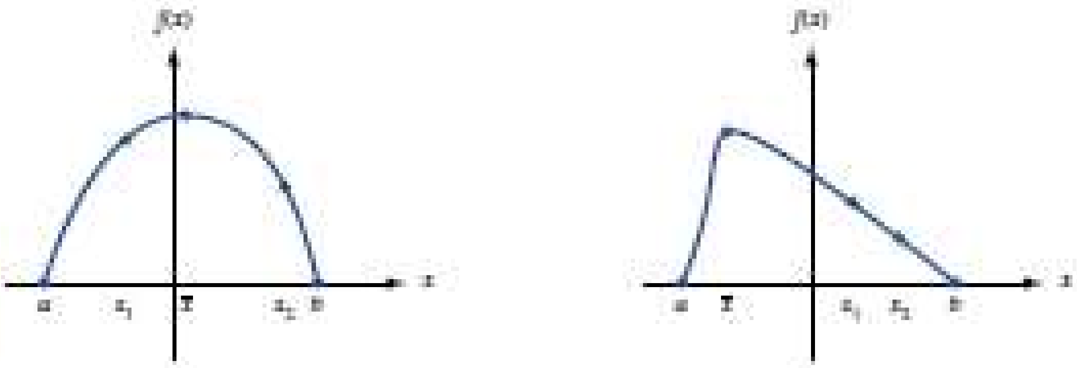


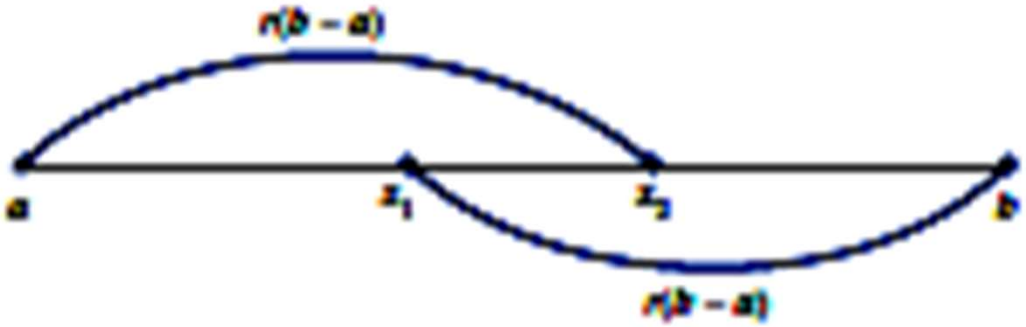
FIGURE 35
If $f(x_1) > f(x_2)$, $\bar{x} \in [a, x_2)$

- Many search algorithms use these ideas to reduce the interval of uncertainty. Most of these algorithms proceed as follows:
  - Begin with the interval of uncertainty for x being $[a,b]$. Evaluate $f(x)$ at two judiciously chosen points x_1 and x_2.
  - Determine which of cases 1-3 holds, and find a reduced interval of uncertainty.
  - Evaluate $f(x)$ at two new points (the algorithm specifies how the two new points are chosen). Return to step 2 unless the length of the interval of uncertainty is sufficiently small.
- Such search algorithms: golden section search, bisection search, Fibonacci search

Golden section search

- How to choose the points to evaluate:
  - x1=b-r(b-a) and x2=a+r(b-a)
- After each iteration, the interval of uncertainty is reduced by r times (check the 3 cases).
  - b-x1=r(b-a) and x2-a=r(b-a)
- After *k* iterations, the interval of uncertainty = $r^k(b-a)$
- Determine *r*:
  - L/rL=rL/(1-r)L
  - x1=b-r(b-a)=a+r[r(b-a)]

  ➔ $r=(5^{1/2}-1)/2=0.618$



Example

Use Golden Section Search to find

$$\begin{aligned} &\max \; -x^2 - 1 \\ &\text{s.t.} \qquad -1 \leq x \leq 0.75 \end{aligned}$$

with the final interval of uncertainty having a length less than $\frac{1}{4}$.

Bisection Search

If f(x) is concave (or simply unimodal) and differentiable

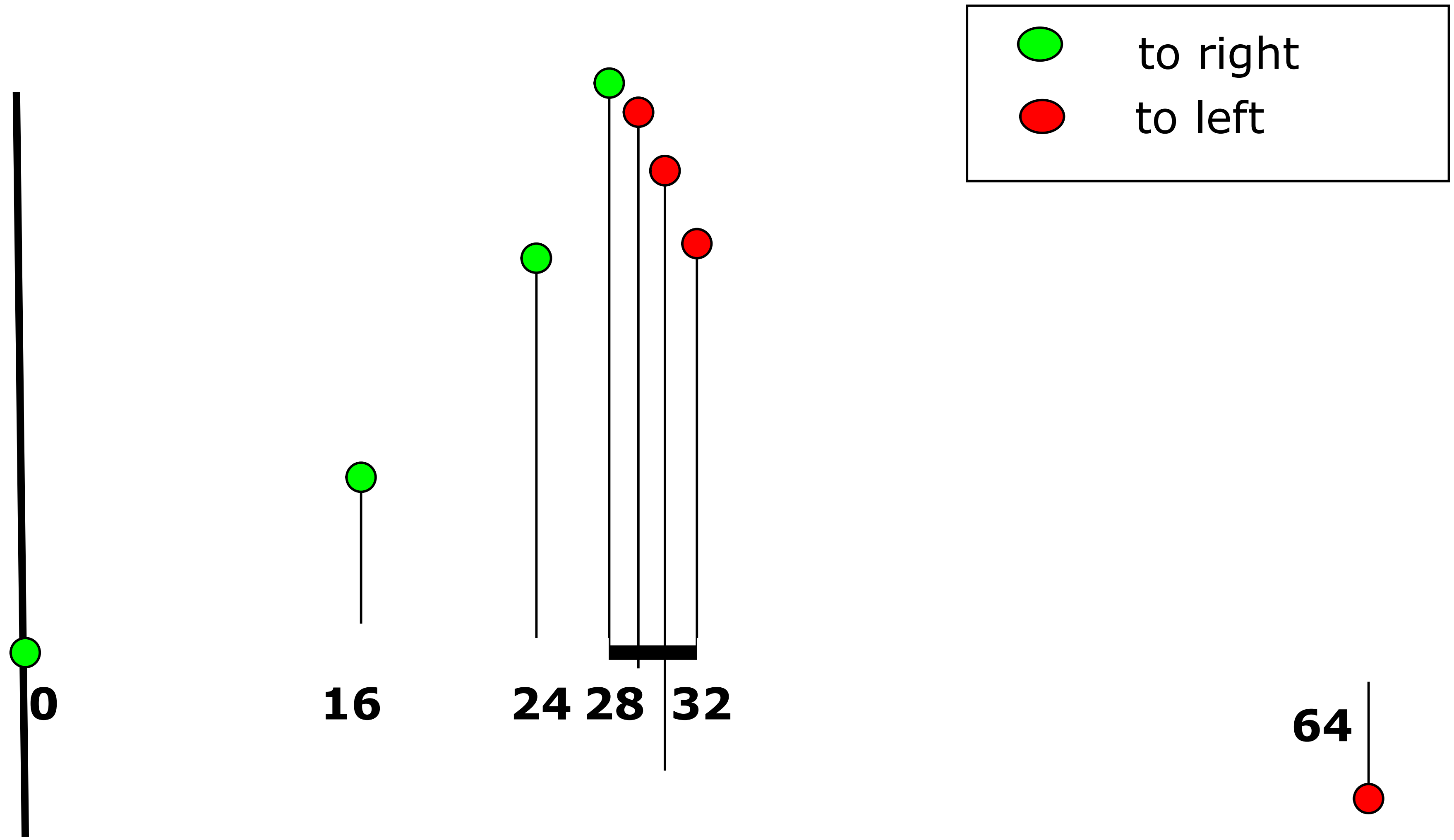
$$\max \quad f(x)$$

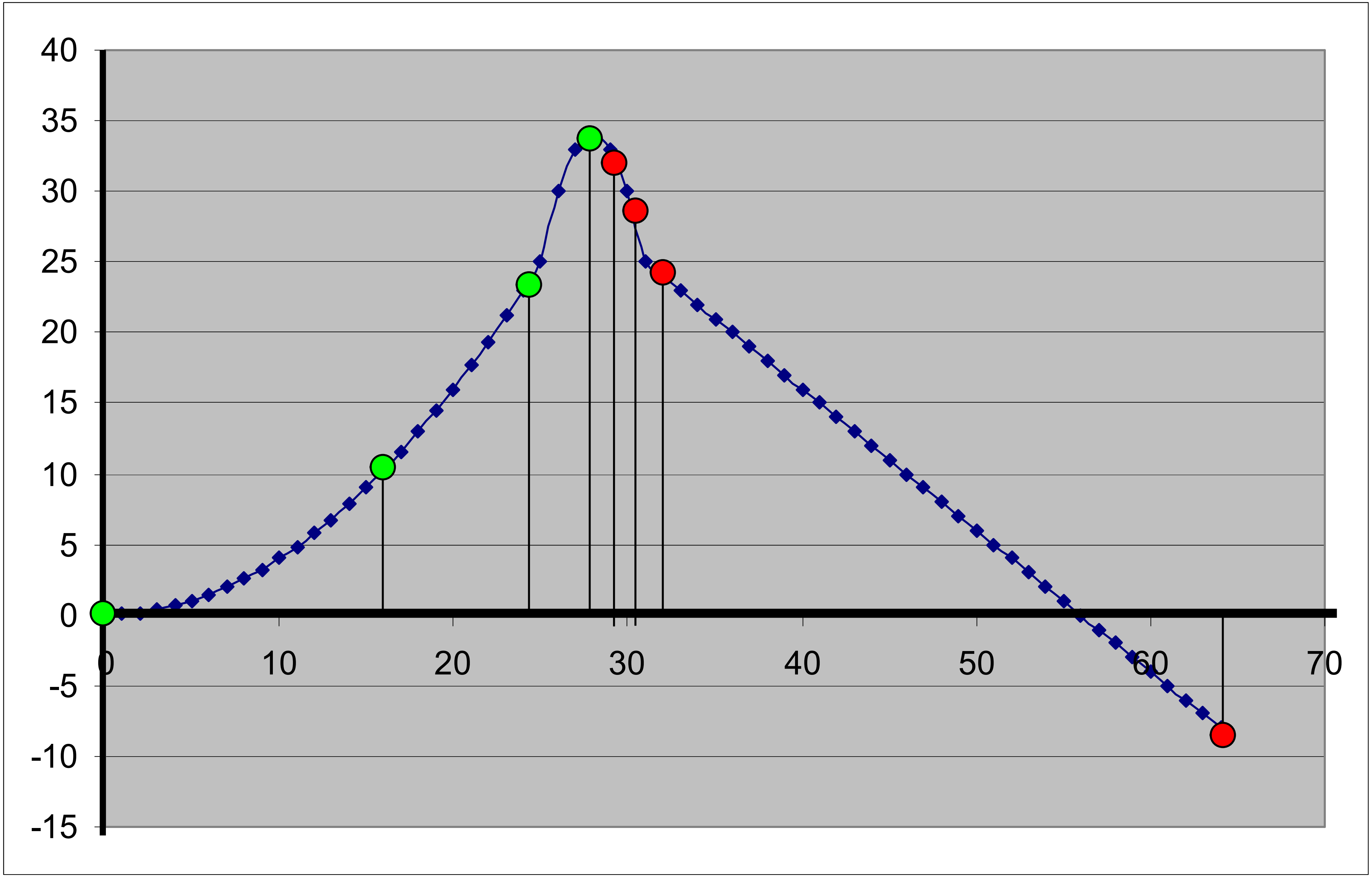
$$\text{s.t.} \quad a <= x <= b$$

Bisection (or Bolzano) Search:

- Step 1. Begin with the region of uncertainty for x as [a, b]. Evaluate $f'(x)$ at the midpoint x_M =(a+b)/2.
- Step 2. If $f'(x_M) > 0$, then eliminate the interval up to x_M. If $f'(x_M) < 0$, then eliminate the interval beyond x_M.
- Step 3. Evaluate $f'(x)$ at the midpoint of the new interval. Return to Step 2 until the interval of uncertainty is sufficiently small.

Determine by taking a derivative if a local maximum is to the right or left.





40
35
30
25
20
15
10
5
0
-5
-10
-15
0
10
20
30
40
50
60
70

Fibonacci Search

- Instead of taking derivatives (which may be computationally intensive), use two function evaluations to determine updated interval.
- Fibonacci Search
- Step 1. Begin with the region of uncertainty for q as [a, b]. Evaluate $f(q_1)$ and $f(q_2)$ for 2 **symmetric** points $q_1<q_2$.
- Step 2. If $f(q_1) <= f(q_2)$, then eliminate the interval up to q_1. If $f(q_1) > f(q_2)$, then eliminate the interval beyond q_2.
- Step 3. Select a second point symmetric to the point already in the new interval, rename these points q_1 and q_2 such that $q_1<q_2$ and evaluate $f(q_1)$ and $f(q_2)$. Return to Step 2 until the interval is sufficiently small.

On Fibonacci search

Fibonacci numbers: 1, 1, 2, 3, 5, 8, 13, 21, 34

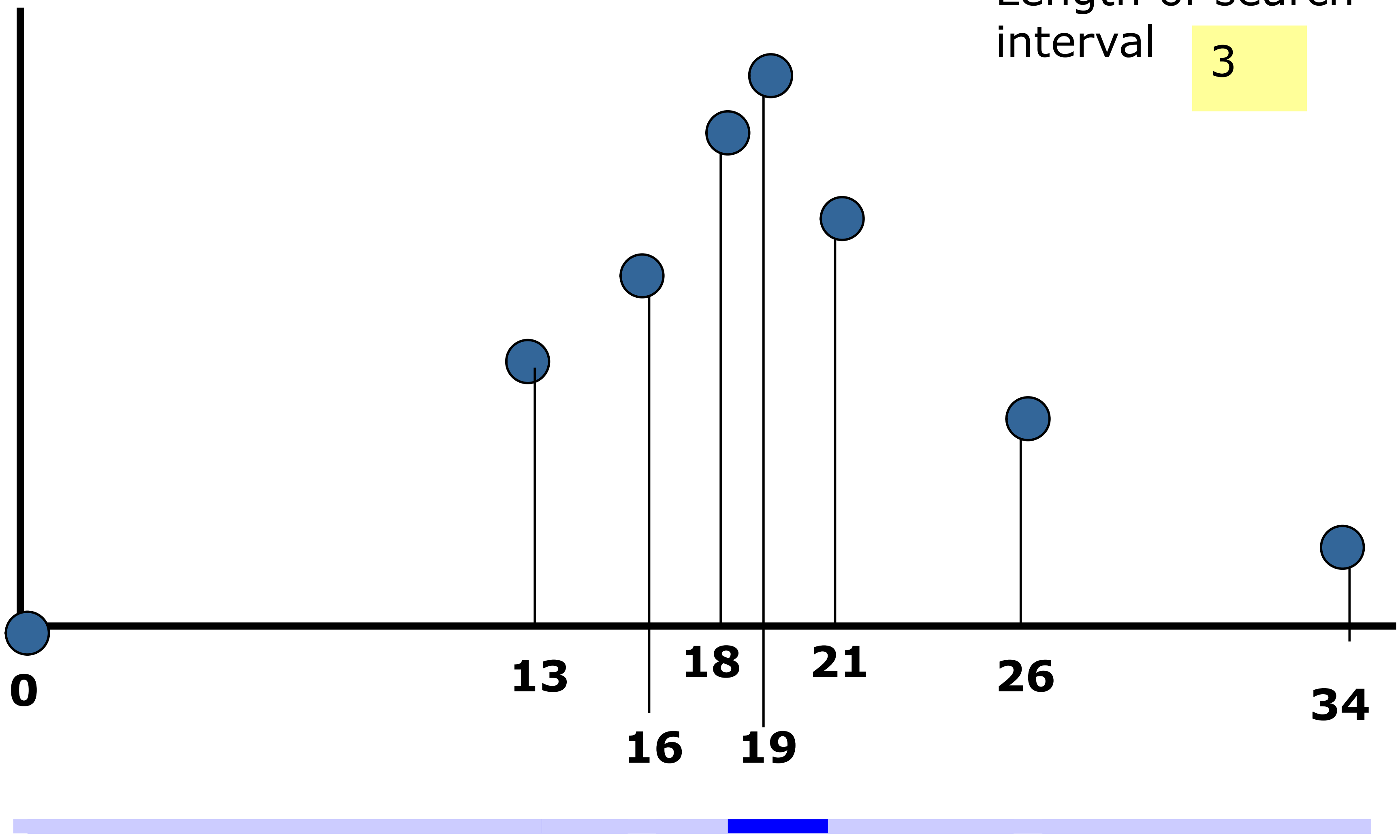
At iteration 1, the length of the search interval is the *k*th Fibonacci number for some *k*

At iteration *j*, the length of the search interval is the k-j+1 Fibonacci number.

The technique converges to the optimal when the function is unimodal.

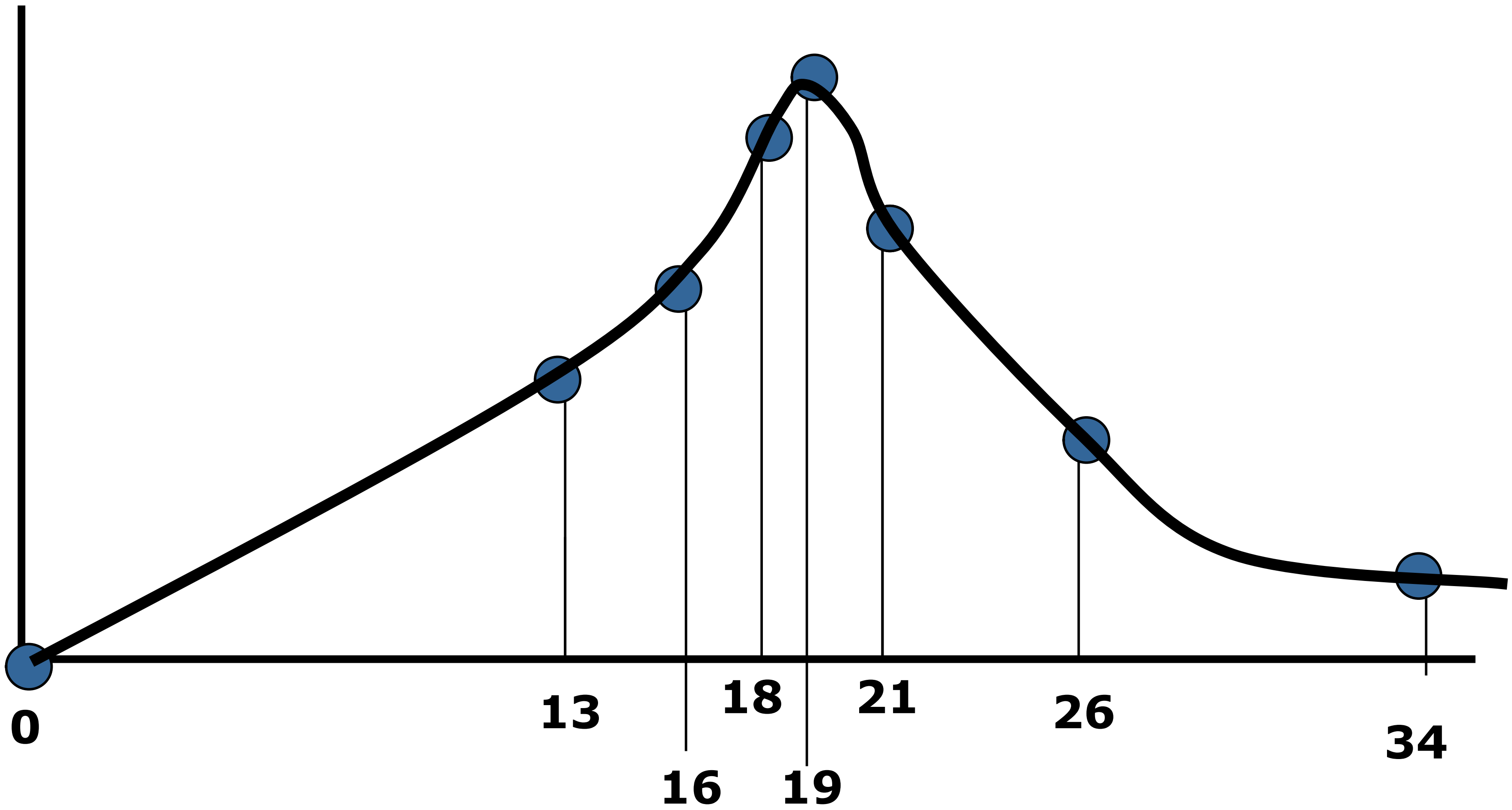
Finding a local maximum using Fibonacci Search

Length of search interval 3

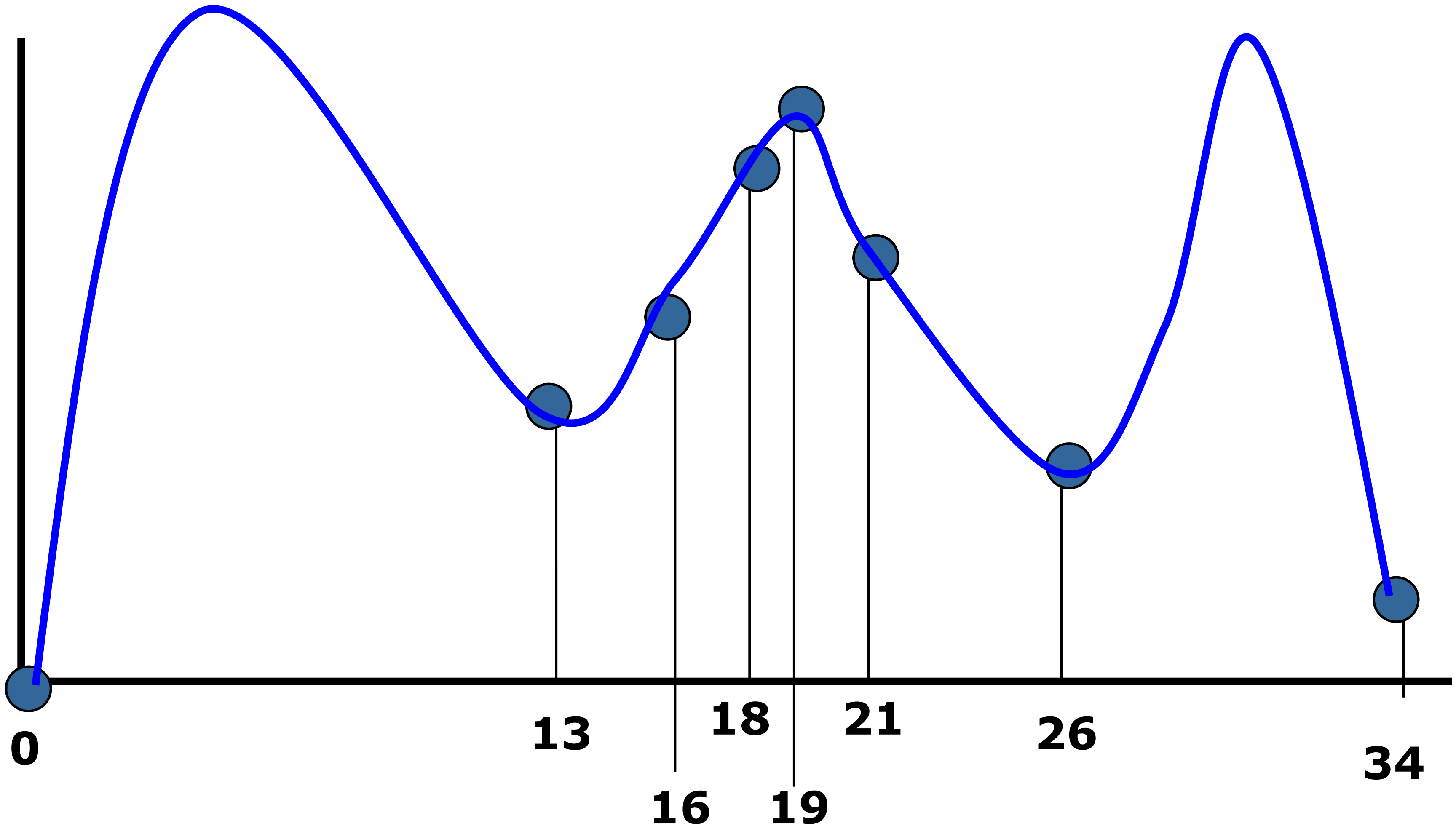


Where the maximum may be

The search finds a local maximum, but not necessarily a global maximum.



The search finds a local maximum, but not necessarily a global maximum.



Number of function evaluations in Fibonacci Search

- As new point is chosen symmetrically, the length l_k of successive search intervals is given by: $l_k = l_{k+1} + l_{k+2}$.
- Solving for these lengths given a final interval length of 1, $l_n =$ 1, gives the Fibonacci numbers: 1, 2, 3, 5, 8, 13, 21, 34,…
- Thus, if the initial interval has length 34, it takes 8 function calculations to reduce the interval length to 1.

- Remark: If the function is convex or unimodal, then Fibonacci search converges to the global maximum.