

## 26. OPTIMIZATION

Topics:

- 

Objectives:

- 

### 26.1 INTRODUCTION

- For simple examples use derivatives and find the maxima/minima
- Consider the example below to find the

$$L =$$

- local and global optimums

- search functions, gradient descent, random walk, simplex

## **26.2 OBJECTIVES AND CONSTRAINTS**

- Objectives are those things we want to minimize, or maximize.
  - money
  - time
  - mass
  - volume
  - power consumption
  - some combination of factors
- Expressed as a function of variables that provides a value
- Consider the example of building a fenced pasture. In this case when the area becomes too large, there is a reduced value. We want to maximize the value of  $V$ .

$$C_{fence} = 20\left(\frac{\$}{m}\right)(2w + 2d)$$

where,

$C_{fence}$  = cost to construct the fence (\$)

$w$  = width of the pasture (m)

$d$  = depth of the pasture (m)

$$C_{land} = 0.35\left(\frac{\$}{m^2}\right)$$

where,

$C_{land}$  = cost for the land

$$R = 0.05\left(\frac{\$}{m^2 yr}\right)wd \quad wd < 300m^2$$

$$R = 0.01\left(\frac{\$}{m^2 yr}\right)(wd - 300m^2) + 60\$ \quad wd \geq 300m^2$$

where,

$R$  = revenue generated by pasture land

$$V = R - C_{fence} - C_{land}$$

where,

$V$  = total value

Figure 26.1 Example cost function for building a fence around a pasture

- The cost function can be written as..

```

double cost(double w, double d){
    double value;
    double cfence, cland, R;

    Cfence = 40*(w + d);
    Cland = 0.05*w*d;
    if (w*d < 300){
        R = 0.05 * w * d;
    } else {
        R = 0.01 * (w * d - 300) + 60;
    }
    value = R - Cfence - Cland

    return value;
}

```

*Figure 26.2* A subroutine for cost function calculation

- Constraints are boundaries that cannot be crossed.
- Example of constraints, the pasture cannot be larger than one 1600m by 1600m because of the constraints of an existing road system.

$$w \leq 1600m$$

$$d \leq 1600m$$

*Figure 26.3* Example constraint functions for a pasture

- The cost function can be written as..

```

double cost(double w, double d){
    double value;
    double cfence, cland, R;

    Cfence = 40*(w + d);
    Cland = 0.05*w*d;
    if (w*d < 300){
        R = 0.05 * w * d;
    } else {
        R = 0.01 * (w * d - 300) + 60;
    }
    value = R - Cfence - Cland

    if(w > 1600) value = 1000000;
    if(d > 1600) value = 1000000;

    return value;
}

```

*Figure 26.4* A subroutine for cost function calculation

- Slack variables allow constraints to be considered as part of the cost function. Helps with a system with many local minimum.

$$C_{penalty} = \left(\frac{w}{1600m}\right)^4 + \left(\frac{d}{1600m}\right)^4$$

$$V = R - C_{fence} - C_{land} - C_{penalty}$$

*Figure 26.5* Example of slack variables for including constraints

- The cost function can be written as..

```

double cost(double w, double d){
    double value;
    double cfence, cland, R;
    double slack;

    Cfence = 40*(w + d);
    Cland = 0.05*w*d;
    if (w*d < 300){
        R = 0.05 * w * d;
    } else {
        R = 0.01 * (w * d - 300) + 60;
    }
    // slack variable
    slack = pow((w/1600), 4) + pow((d/1600), 4);
    value = R - Cfence - Cland - slack;

    return value;
}

```

*Figure 26.6* A subroutine for cost function calculation

## 26.3 SEARCHING FOR THE OPTIMUM

- Local Search Space
- A topographical map shows the relationship between search parameters and cost values.

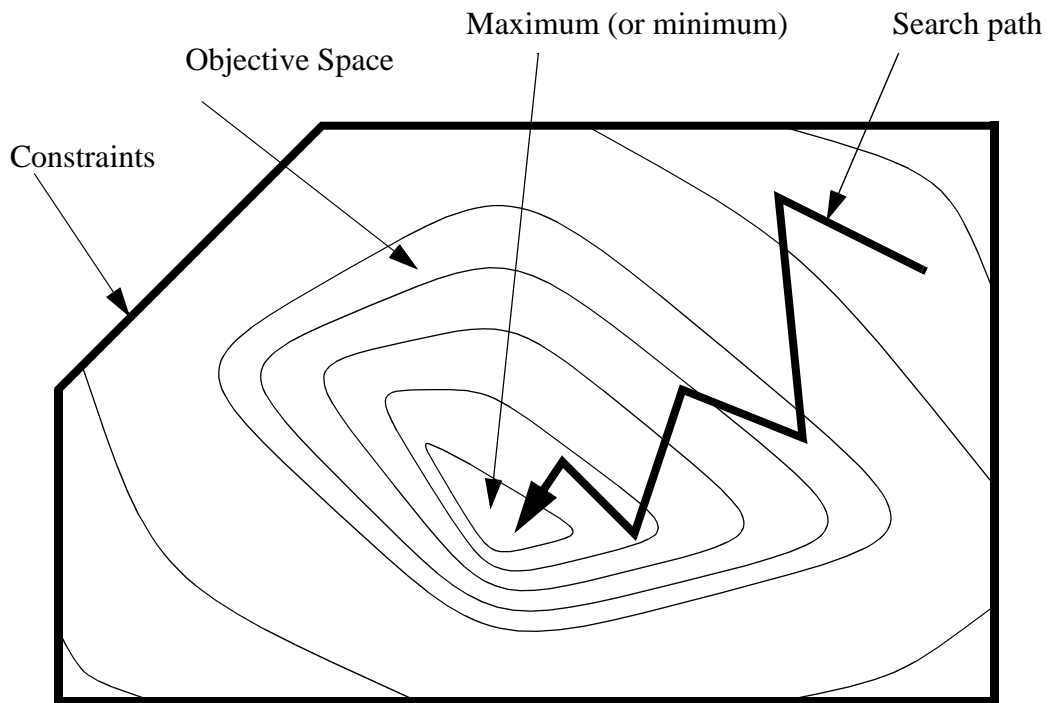


Figure 26.7 Local searches

- Global Search space. In this case the system becomes 'stuck' in a local mimina.

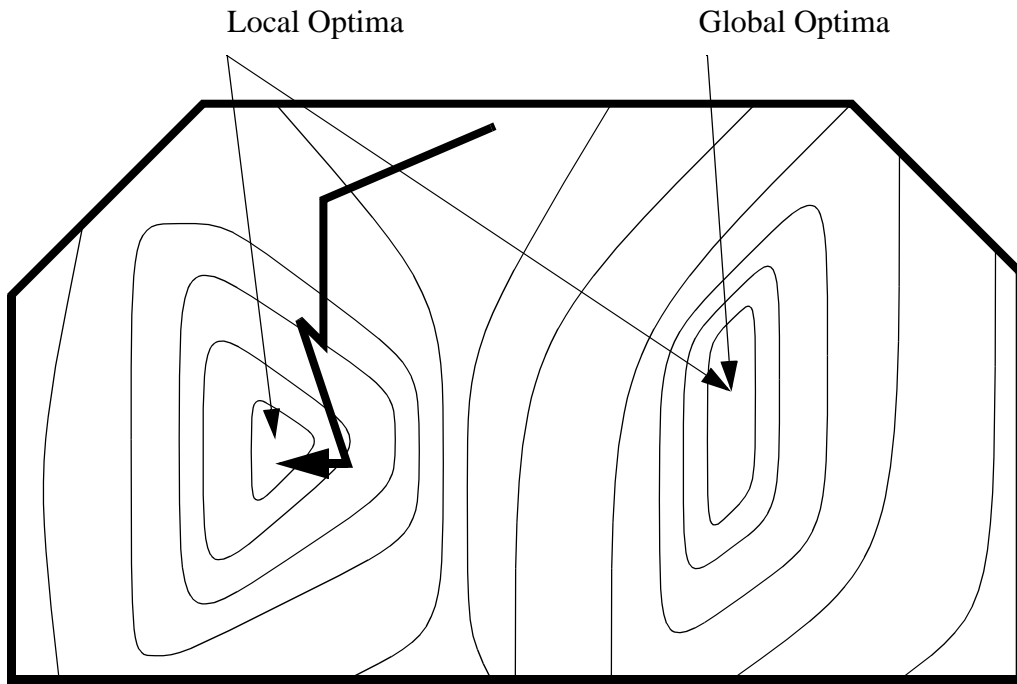
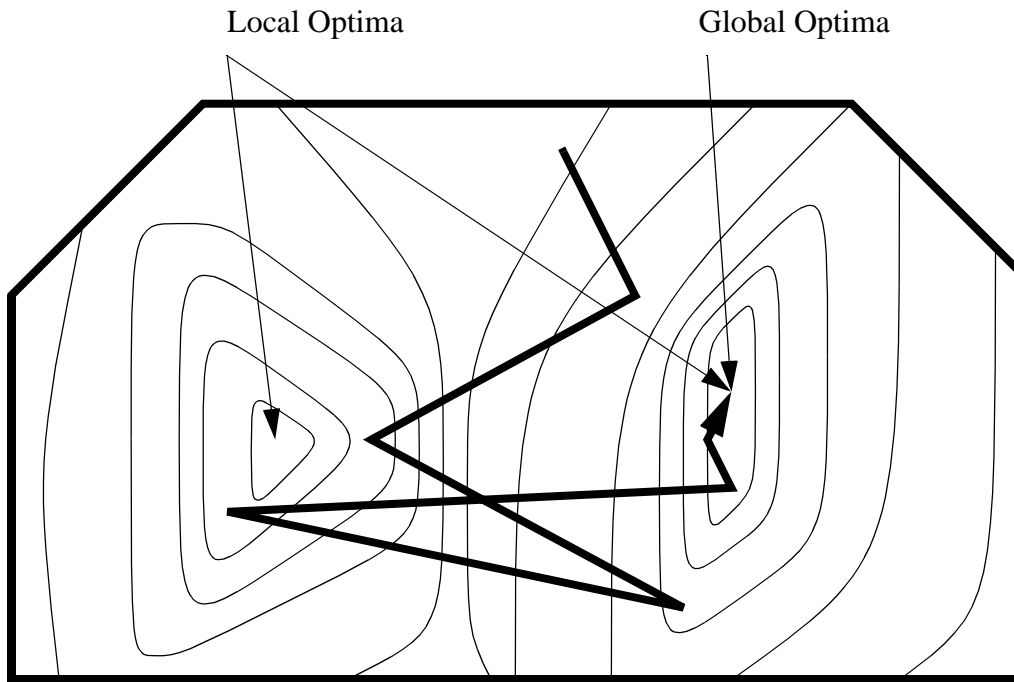


Figure 26.8 Global searches

- Global Search space. In this case the system searches all maxim.



*Figure 26.9* A Global Search

## 26.4 OPTIMIZATION ALGORITHMS

- The search algorithms change system parameters and try to lower system parameters.
- The main question is how to change the system parameters to minimize the system value.

### 26.4.1 Random Walk

## **26.4.2 Gradient Decent**

## **26.4.3 Simplex**

## **26.5 SUMMARY**

•

## **26.6 PRACTICE PROBLEMS**

1.

## **26.7 PRACTICE PROBLEM SOLUTIONS**

1.

## **26.8 ASSIGNMENT PROBLEMS**

1.