**RESOURCE AND MANAGEMENT TECHNIQUES**

Submitted by-:

Name :- Deepak Kanaujia

Roll No. :- 205119029

**UNIT- 3**

1. **What is Non-Linear Programming? Importance of Non-Linear Programming.**

**Answer:**

Nonlinear Programming (NLP) is the process of solving an optimization problem where some of the constraints or the objective function are nonlinear. Let *n*, *m*, and *p* be positive integers. Let *X* be a subset of *Rn*, let *f*, *gi*, and *hj* be real valued function on *X* for each *i* in {*1*, …, *m*} and each *j* in {*1*, …, *p*}, with at least one of *f*, *gi*, and *hj* being nonlinear.

A nonlinear minimization problem is an optimization problem of the form

Minimize f(x)

Subject to gi(x) <=0 for each i = 1,2,…m

hj(x) =0 for each j = 1,2,….p

{\displaystyle {\begin{aligned}{\text{minimize }}&f(x)\\{\text{subject to }}&g\_{i}(x)\leq 0{\text{ for each }}i\in \{1,\dotsc ,m\}\\&h\_{j}(x)=0{\text{ for each }}j\in \{1,\dotsc ,p\}\\&x\in X.\end{aligned}}}

A nonlinear maximization problem is defined in a similar way.

Non-linear programming methods assure obtaining, with certain precision, local minimum in the search space while the qualify use of metaheuristics could assure to obtain close to the global optimal solution, or populations of them close to the optimal.

**2. Types of Non-Linear Programming.**

ollows: i). NLP technique is successfully applied to the overall cost minimization of transformer active and

mechanical part, ii). Transformer design variables such as the conductors‟ cross-section and windings are

added to the optimization algorithm for an enlarged and transverse optimum transformer designs. The

proposed methods find acceptable optimum transformer design by minimizing either the overall transformer

material cost (i.e. the transformer active part cost plus mechanical part cost) or the overall transformer

materials and operating cost taking into consideration proper loss evaluation factors, while simultaneously

satisfying all the constraints imposed by international standards and transformer user needs, instead of

focusing on the optimization of only one parameter of transformer performance (e. g no-load losses or short

circuit impedance). Using the proposed technique, a graphic user interface (GUI) software package is

developed that combine‟s transformer design with analysis, optimization and visualization tools, useful for

both design optimization and educational use. The technique is applied to the design of power transformers

of several ratings and loss. Categories and the results are compared with transformer design optimization

method (which is already used by transformer industry), resulting to significant cost savings.

**Answer:**

ollows: i). NLP technique is successfully applied to the overall cost minimization of transformer active and

mechanical part, ii). Transformer design variables such as the conductors‟ cross-section and windings are

added to the optimization algorithm for an enlarged and transverse optimum transformer designs. The

proposed methods find acceptable optimum transformer design by minimizing either the overall transformer

material cost (i.e. the transformer active part cost plus mechanical part cost) or the overall transformer

materials and operating cost taking into consideration proper loss evaluation factors, while simultaneously

satisfying all the constraints imposed by international standards and transformer user needs, instead of

focusing on the optimization of only one parameter of transformer performance (e. g no-load losses or short

circuit impedance). Using the proposed technique, a graphic user interface (GUI) software package is

developed that combine‟s transformer design with analysis, optimization and visualization tools, useful for

both design optimization and educational use. The technique is applied to the design of power transformers

of several ratings and loss. Categories and the results are compared with transformer design optimization

method (which is already used by transformer industry), resulting to significant cost savings.

1. **UNCONSTRAINED OPTIMIZATION IN ONE DIMENSION:**

Here we begin by considering a significantly simplified (but nonetheless important) nonlinear programming problem, i.e., the domain and range of the function to be minimized are one-dimensional and there are no constraints. A necessary condition for a minimum of a function was developed in calculus and is simply f’(x) = 0.

Note that higher derivative tests can determine whether the function is a max or a min, or the value f(x + δ) may be compared to f(x).

Note that if we let g(x) = f’(x) then we may convert the problem of finding a minimum or maximum of a function to the problem of finding a zero.

1. **Bisection Algorithm:**

Let x ∗ be a root, or zero, of g(x), i.e., g(x\*) = 0. If an initial bracket [a, b] is known such that x ∗ ∈ [a, b], then a simple and robust approach to determining the root is to bisect this interval into two intervals [a, c] and [c, b] where c is the midpoint, i.e., c = (a + b)/2

If g(a)g(c) < 0

then we conclude x ∗ ∈ [a, c]

while if g(b)g(c) < 0

then we conclude x ∗ ∈ [b, c]

This process may now be iterated such that the size of the bracket (as well as the actual error of the estimate) is being divided by 2 every iteration.

1. **Newton’s Method:**

Note that in the bisection method the actual value of the function g(x) was only being used to determine the correct bracket for the root. Root finding is accelerated considerably by using this function information more effectively.

1. **UNCONSTRAINED OPTIMIZATION IN HIGHER DIMENSION:**

Now we consider the problem of minimizing (or maximizing) a scalar function of many variables, i.e., defined on a vector field. We consider the extension of Newton’s method presented in the previous section as well as a classical approach known as steepest descent.

**a. Taylor Series in Higher Dimensions**

**b. Newton’s Method**

**c. Steepest Descent Method**

3. **CONSTRAINED OPTIMIZATION AND LAGRANGE MULTIPLIERS:**

Some constrained optimizations are

**a. One Equality Constrained**

**b. Several Equality Constrained**

**c. Inequality Constrained**

**3. Applications of Non-Linear Programming.**

1**.Application of Nonlinear Programming for Optimization of Nutrient Requirements for Maximum Weight Gain in Buffaloes:**

Nonlinear effects of nutrient ingredients are introduced as an approach closer to the true effects of nutrient ingredients. A nonlinear model is developed to take consideration of nutrient ingredients more effectively. Proposed model with nonlinear programming measures its performance and gives a comparative result with linear programming models.. Leading to the same guideline a ration can be formulated using all its nutrient ingredients simultaneously at the optimum level. In this paper, it is envisaged to develop a mathematical model using non-linear programming to take simultaneous effects of all nutrient ingredients and the diet is optimized by using Kuhn- Tucker conditions. This result is also compared to than that of linear programming formulation of the model.

#### 1. **Weightage of Variables**

First of all, linear relationship for dependent and independent variables is formulated to decide the weightage of the variables. Assuming a linear relationship between weight gain of buffaloes and intake of DM, CP and TDN, the weightage of these variables was decided.

Using least square method, the relationship is depicted in the following equation which describes the weightage of the variables x1, x2 and x3.

|  |  |
| --- | --- |
|  | (1) |

#### 2. **Relationship between Variables**

By using least square method, the relations between y and x1, y and x2, y and x3 of different degrees were established and then by using F-test the relation of best fit was decided. Applying the F-test, the following most appropriate relationship between the variables were derived:

|  |  |
| --- | --- |
|  |  |

#### 3. **Formulation of Objective Function**

The objective function was established by using the appropriate relations of the variables x1, x2, x3 according to their weightage on weight gain of the buffalo calves. The weightage with respect to total effect of this weightage was considered:

|  |  |
| --- | --- |
|  |  |

#### 4. **Constraints**

The constraints according to feeding standards on the above-mentioned variables according to feeding standards of NRC (1981) were applied

|  |  |
| --- | --- |
|  |  |

#### 5. **Problem Defined**

The main problem is formulated to maximise weight gain of the animal:



subject to:

|  |  |
| --- | --- |
|  |  |

#### 6**. Solution of the Problem**

Introducing Kuhn-Tucker conditions, the weight gain of the buffalo calves could be maximized as:



Using Kuhn-Tucker conditions, the following set of equations were obtained for optimal solutions:













x1 ≤ 396.311

x2 ≤ 37.708

x3 ≤ 368.1687036



Solving these equations the optimum values of the three nutrients is found out to maximize the body weight gain.

Accordingly we have:

x1= 381.3028, x2= 7.708, x3= 368.1687036 g/kg W0.75

It also gives, λ1= 0.393301399, λ2= 0.027548012

which satisfied all the conditions.

The problem is also formulized and solved by simplex method and it gives,

x1= 396.311, x2= 37.708, x3= 368.1687036 g/kg W0.75

Comparison shows that by linear programming result is obtained at corner points of feasible area and optimization is at comparatively at higher values of nutrient ingredients. This comparison represents that non-linear programming is better way to take simultaneous effect of all nutrient ingredients together and maximize the weight gain in animal with optimized value of nutrient ingredients.

**UNIT-4**

**1.What is Integer Programming?**

**Answer:**

An integer programming problem (ILP) is a mathematical optimization or feasibility program in which some or all of the variables are restricted to be integers.

An integer linear program in canonical form is expressed as:

Maximize cT x

Subject to Ax<=b,

x>=0,

and x € Z

{\displaystyle {\begin{aligned}&{\text{maximize}}&&\mathbf {c} ^{\mathrm {T} }\mathbf {x} \\&{\text{subject to}}&&A\mathbf {x} \leq \mathbf {b} ,\\&&&\mathbf {x} \geq \mathbf {0} ,\\&{\text{and}}&&\mathbf {x} \in \mathbb {Z} ^{n},\end{aligned}}}

and an ILP in standard form is expressed as

maximize cT x

subject to Ax +s =b,

x>=0, s>=0,

and x € Z

{\displaystyle {\begin{aligned}&{\text{maximize}}&&\mathbf {c} ^{\mathrm {T} }\mathbf {x} \\&{\text{subject to}}&&A\mathbf {x} +\mathbf {s} =\mathbf {b} ,\\&&&\mathbf {s} \geq \mathbf {0} ,\\&&&\mathbf {x} \geq \mathbf {0} ,\\&{\text{and}}&&\mathbf {x} \in \mathbb {Z} ^{n},\end{aligned}}}

where {\displaystyle \mathbf {c} ,\mathbf {b} }c,bc, b are vectors and A{\displaystyle A} is a matrix, where all entries are integers. As with linear programs, ILPs not in standard form can be converted to standard form  by eliminating inequalities, introducing slack variables (s{\displaystyle \mathbf {s} }ss) and replacing variables that are not sign-constrained with the difference of two sign-constrained variables.

**2. Types of Integer Programming.**

**Answer:**

There are three types of IP models:

1. In mixed integer programming, only some of the variables are restricted to integer values.
2. In pure integer programming, all the variables are integers.
3. In binary integer programming or 0-1 integer programming, all the variables are binary (restricted to the values 0 or 1).

**Mixed Integer Programming:-**

The problems most commonly solved by the Gurobi Parallel Mixed Integer Programming solver are of the form:

|  |  |
| --- | --- |
| Objective: | minimize cT x |
| Constraints: | A x = b (linear constraints) |
|  | l ≤ x ≤ u (bound constraints) |
|  | some or all xj must take integer values (integrality constraints) |

**Binary Integer Programming**:-

Binary Integer Programming (BIP) is an approach to solve a system of linear inequalities in binary unknowns (0 or 1 in what follows). Integer programming has been studied in mathematics, computer science, and operations research for more than 40 years. It has been successfully applied to solve a huge number of large-scale combinatorial problems.

The general form of an integer linear programming problem is

max { c T x | Ax ≤ b, x ∈ Zn }

with a real matrix A of a dimension m by n, and vectors c ∈ Rn , b ∈ Rm, c T x being the scalar product of the vectors c and x. If the system Ax ≤ b includes the constraints 0 ≤ x ≤ 1, we get a binary integer linear programming problem (BIP).

A vector x\* in Zn with Ax\* ≤ b is called a feasible solution. If moreover,

c T x\* = max { c T x | Ax ≤ b, x ∈ Zn },

then x\* is called an optimal solution and c T x\* the optimal value.

We use specialized branch and bound method for solving BIP known as Balas Additive Algorithm.

The keys how Balas Additive Algorithm works lies in its special structure:

1. The objective function is minimization and all of the coefficients are nonnegative, so we would prefer to set all the variables to zero to give the smallest value of Z.
2. If we cannot set all the variables to zero without violating one or more constraints, then we prefer to set all the variables that has smallest index 1. This is because variables are ordered so that those earlier in the list increase by Z by smallest amount.

**3. Applications of Integer Programming.**

**Answer:**

# **1.Integer programming models for mid-term production planning for high-tech low-volume supply chains:**

Mid-term production planning (6 to 24 months) allocates the capacity of production resources to different products over time and coordinates the associated inventories and material inputs so that known or predicted demand is met in the best possible manner. High-tech low-volume industries can be characterized by the limited production quantities and the complexity of the supply chain. To model this, we introduce a mixed integer linear programming model that can handle general supply chains and production processes that require multiple resources. Furthermore, it supports semi-flexible capacity constraints and multiple production modes.

Since the number of alternative capacity constraints is exponential, we first solve the second formulation without capacity constraints. Each time an incumbent is found during the branch and bound process a maximum flow problem is used to find missing constraints. If a missing constraint is found it is added and the branch and bound process is restarted. Results from a realistic test case show that utilizing this algorithm to solve the second formulation is significantly faster than solving the first formulation.

**2.Integer Programming in Telecommunication:**

The cellular telecommunication network design aims to define and dimension the cellular telecommunication system topology in order to serve the voice and/or data traffic demand of a particular geographic region. In this article, we introduce a novel model that addresses the cellular system design problem in a complete fashion.

We propose a linear mixed-integer programming model that gathers together into the same model the base station location problem, the frequency channel assignment problem and the base station connection to the fixed network. We still present some computational analyses in order to evaluate the model tradeoffs and its time complexity. In conclusion, we mention our current work towards an effective technique to solve the proposed model.

**UNIT-5**

**1.What is Queuing Theory? Importance of Queuing Theory.**

**ANS :**

**Queuing Theory :-**

Queuing theory (or queueing theory) refers to the mathematical study of the formation, function, and congestion of waiting lines, or queues.

At its core, a queuing situation involves two parts-

1. Someone or something that requests a service—usually referred to as the customer, job, or request.
2. Someone or something that completes or delivers the services—usually referred to as the server.

To illustrate, let’s take two examples. When looking at the queuing situation at a bank, the customers are people seeking to deposit or withdraw money, and the servers are the bank tellers. When looking at the queuing situation of a printer, the customers are the requests that have been sent to the printer, and the server is the printer.

**Importance of Queuing Theory :-**

SERVER

QUEUE

POPULATION OF CUSTOMERS

Waiting in line is a part of everyday life because as a process it has several important functions. Queues are a fair and essential way of dealing with the flow of customers when there are limited resources. Negative outcomes arise if a queue process isn’t established to deal with overcapacity.

**Example** **:** when too many visitors navigate to a website, the website will slow and crash if it doesn’t have a way to change the speed at which it processes requests or [a way to queue visitors](https://queue-it.com/online-queueing-system/).

Or, imagine planes waiting for a runway to land. When there is an excess of planes, the absence of a queue would have real safety implications as planes all tried to land at the same time.

Queuing theory is important because it helps describe features of the queue, like average wait time, and provides the tools for optimizing queues. From a business sense, queuing theory informs the construction of efficient and cost-effective workflow systems.

1. **Models of QT. Briefly explained.**

**QUEUING MODELS :**

**INTRODUCTION :-**

A queuing system consists of one or more servers that provide service of some sort to arriving customers. Customers who arrive to find all servers busy generally join one or more queues (lines) in front of the servers, hence the name **Queuing Systems**. There are several everyday examples that can be described as queuing systems, such as bank-teller service, computer systems, manufacturing systems, maintenance systems, communications systems and so on.

Components of a Queuing System: A queuing system is characterised by three components:

**- Arrival process**

**– Service mechanism**

**– Queue discipline**

**Arrival Process :-**

Arrivals may originate from one or several sources referred to as the calling population. The calling population can be limited or 'unlimited'. An example of a limited calling population may be that of a fixed number of machines that fail randomly. The arrival process consists of describing how customers arrive to the system. If Ai is the interarrival time between the arrivals of the (i-1)th and ith customers, we shall denote the mean (or expected) inter-arrival time by E(A) and call it (λ ); = 1/(E(A) the arrival frequency.

**Service Mechanism:-**

The service mechanism of a queuing system is specified by the number of servers (denoted by s), each server having its own queue or a common queue and the probability distribution of customer's service time. let Si be the service time of the ith customer, we shall denote the mean service time of a customer by E(S) and µ = 1/(E(S) the service rate of a server.

**Queue Discipline:-**

Discipline of a queuing system means the rule that a server uses to choose the next customer from the queue (if any) when the server completes the service of the current customer.

Commonly used queue disciplines are:

**FIFO -** Customers are served on a first-in first-out basis.

**LIFO -** Customers are served in a last-in first-out manner.

**Priority -** Customers are served in order of their importance on the basis of their service requirements.

1. **Applications of Queuing Theory :**

Queuing theory is powerful because the ubiquity of queue situations means there are countless and diverse applications of queuing theory.

* telecommunications
* transportation
* logistics
* finance
* emergency services
* computing
* industrial engineering
* project management

# **Analysis of an M/M/*C* Queueing System with Impatient Customers and Synchronous Vacations:**

An queueing M/M/C system with impatient customers and a synchronous vacation policy. Customers arrive according to a Poisson process at rate λ. The service is provided by servers, who serve the customers on a first-come first-served (FCFS) basis. The service time of each customer is exponentially distributed with mean 1/µ.

The multiple synchronous vacation policy is described as follows. When the server finishes serving a customer and finds the system empty, all servers immediately leave for a vacation. If servers return from a vacation to find an empty queue, they immediately leave for another vacation; otherwise, they return to serve the queue. The duration of a vacation is exponentially distributed with mean 1/Υ.

During the vacation, customers are impatient. That is, an arriving customer who finds that all servers are on vacation activates an “impatience timer" T, which is exponentially distributed with mean 1/ƺ. If the customer’s service has not been completed before the customer’s timer expires, the customer abandons the queue and never returns.

**END**