Quadratic Programming
(Wolfe's and Beale's Method)

Quadratic programming deals with the NLPP of maximizing (or minimizing) the quadratic objective function subject to a set of linear inequality constraints.

The general QPP can be defined as follows

Def^n:-

Let X^T and $C \in R^n$ and Q be a symmetric $n \times n$ real matrix.

Then the problem of maximizing (i.e. to determining x) so as to maximize

$$f(X) = CX + \frac{1}{2} \underline{X^T Q X}$$

defines a quadratic form

subject to the constraints

$$AX \leq b$$

and $X \geq 0$.

where $b^T \in R^m$, and A be $m \times n$ real matrix, is called General Quadratic Programming problem (GQPP).

1) if $x^T Q x$ is +ve semi definite (or negative semi-definite) then it is convex (or concave) in x over all of R^n.

2) if $x^T Q x$ is +ve definite (or neg. definite) then it is strictly convex (or strictly concave) in x over all of R^n.

positive definite → $x^T Q x > 0$ for $x \neq 0$

positive semi definite → $x^T Q x \geq 0 \ \forall \ x$.

//ly.

-ve definite → if $-x^T Q x$ is +ve definite

-ve semi definite → if $-x^T Q x$ is +ve semi definite.

$x^T Q x$ is -ve-definite in max. case

" +ve definite " min "

The constraints are assumed to be linear which ensures a convex solution

Scanned with CamScanner

WOLFE'S METHOD TO Solve QPP

Discuss wolfe's method modified simplex method to solve NLPP or QPP

Let us consider the QPP

$$\max.\ Z = \sum_{j=1}^{n} c_j x_j + \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n} c_{jk} x_j x_k$$

S.t.

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \ ; \ i = 1, 2, \dots m$$

$$x_j \geq 0$$

Where, $c_{jk} = c_{kj} \ \forall \ k, j = 1, 2, \dots n$

and $b_i \geq 0$

Also assume that quadratic form $\sum_{j=1}^{n}\sum_{k=1}^{n} c_{jk} x_j x_k$ be negative semi definite.

We will discuss wolfe's method step by step as follows

Step I:

Let us convert the inequality constraints into equations by introducing slack variables q_i^2 in the ith constraint where $i=1,2,\dots m$

and slack variables r_j^2 in the jth non-negative constraint where $j=1,2,3\dots n$.

Step II:

Construct the Lagrangian function

$$L(X,q,r,\lambda,\mu) = f(X) - \sum_{i=1}^{m}\lambda_i\left\{\sum_{j=1}^{n} a_{ij}x_j + q_i^2 - b_i\right\} - \sum_{j=1}^{n}\mu_j\left(-x_j + r_j^2\right)$$

$$= \sum_{j=1}^{n} c_{ij}x_j + \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n} c_{jk}x_jx_k - \sum_{i=1}^{m}\lambda_i\left\{\sum_{j=1}^{n} a_{ij}x_j + q_i^2 - b_i\right\} - \sum_{j=1}^{n}\mu_j\left\{-x_j + r_j^2\right\} \quad \text{——— (1)}$$

where,

$$x = (x_1, x_2, \dots, x_n)$$
$$q = (q_1^2, q_2^2 \dots q_m^2)$$
$$r = (r_1^2, r_2^2 \dots r_n^2)$$
$$\lambda = (\lambda_1, \lambda_2 \dots \lambda_m)$$
$$\mu = (\mu_1, \mu_2 \dots \mu_n)$$

differentiating eq^n ① partially w.r.t. x, q, r, λ, μ and equating the first order derivatives to zero, we get the Kuhn-Tucker conditions from the resulting eq^n ①.

Step III:

Introduce wolfe's non-negative artificial variable v_j; $j = 1, 2, \dots n$ in the Kuhn-Tucker condition

$$c_j + \sum c_{jk} x_k - \sum_{i=1}^{m} \lambda_i a_{ij} + \mu_j = 0 \qquad j = 1, 2 \dots n$$

and construct an objective function

$$Z_v = v_1 + v_2 + \dots + v_n$$

Step-IV

obtain the initial B.F.S. of the L.P.P

$$\min Z_v = v_1 + v_2 + \cdots + v_n$$

s.t.

$$\sum_{k=1}^{n} c_{jk} x_k - \sum_{i=1}^{m} \lambda_i a_{ij} + \mu_j + v_j = -c_j$$

and

$$\sum_{j=1}^{n} a_{ij} x_j + q_i^2 = b_i$$

$$\lambda_i, v_j, \mu_j, x_j \geq 0$$

$$i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, n$$

Step V:

Apply usual Two phase simplex method to find an optimal solution of the above L.P.P. formed in Step IV

This solution must satisfy the complementary slackness conditions

$$\sum_{j=1}^{n} \mu_j x_j + \sum_{i=1}^{m} \lambda_i s_i = 0 \quad (s_i = q_i^2)$$

or, $\lambda_i S_i = 0$ or $\mu_j x_j = 0$

$i = 1, 2, \ldots m$

$j = 1, 2, \ldots n$

Step. VI

The optimal solution obtained in step V is the optimal solution of the given NLPP also.

If not apply usual simplex Algorithm to find the optimal solution.