# Lagrange Multiplier Method

Solve the multi-objective minimization problem:

$$min_x E1(x), E2(x), \ldots, Ek(x) \tag{1}$$

where

$$E_i = 0.5x^T H_i x + x^T f_i \tag{2}$$

and $E_i$ is deemed "more important" than $E_{i+1}$ (lexicographical ordering).

## Algorithm

$$C_0 = H_0 \tag{3}$$

$$d_0 = f_0 \tag{4}$$

$$C_{i+1} = \begin{bmatrix} H_{i+1} & C_i^T \\ C_i & 0 \end{bmatrix} \tag{5}$$

$$d_i = \begin{bmatrix} f_{i+1} \\ d_i \end{bmatrix} \tag{6}$$

Where $H_{i+1}$ is padded by zeros to match the size of $C_i$, and $f_{i+1}$ is padded by zeros to match the size of $d_i$.

When there are no more Energies or $C_i$ is full-rank the equation $C\vec{z} = \vec{d}$ is solved. The final solution is the first n elements of $\vec{z}$.

$$\min_{x,\lambda_1,\lambda_2,\ldots,\lambda_{i-1}} \frac{1}{2}x^T H_i x + x^T f_i + 0 \cdot \lambda_1 + 0 \cdot \lambda_2 + \ldots + 0 \cdot \lambda_{i-1} \tag{7}$$

such that $C_{i-1} \cdot [x^T \lambda_1 \lambda_2 \ldots \lambda_{i-1}]^T = D_{i-1}$ or

$$y = [x^T \lambda_1 \lambda_2 \ldots \lambda_{i-1}]^T \tag{8}$$

$$\min_y \frac{1}{2}y^T A y - y^T B \tag{9}$$

such that $C_{i-1} \cdot [x^T \lambda_1 \lambda_2 \ldots \lambda_{i-1}]^T = D_{i-1}$.

## Example

For example, on the third iteration we have the following:

$$C_2 = \begin{bmatrix} H_2 & 0 & H_1 & H_0^T \\ 0 & 0 & H_0 & 0 \\ H_1^T & H_0^T & 0 & 0 \\ H_0 & 0 & 0 & 0 \end{bmatrix} \tag{10}$$

where each $H_i$ and 0 are $n \times n$ matrices

$$d_2 = \begin{bmatrix} f_2 \\ 0 \\ f_1 \\ f_0 \end{bmatrix} \tag{11}$$

where each $f_i$ and 0 are $n \times 1$ vectors.

On the fourth iteration we have the following

$$C_3 = \begin{bmatrix} H_3 & 0 & 0 & 0 & H_2^T & 0 & H_1 & H_0 \\ 0 & 0 & 0 & 0 & 0 & 0 & H_0 & 0 \\ 0 & 0 & 0 & 0 & H_1^T & H_0^T & 0 & 0 \\ 0 & 0 & 0 & 0 & H_0 & 0 & 0 & 0 \\ H_2 & 0 & H_1 & H_0^T & 0 & 0 & 0 & 0 \\ 0 & 0 & H_0 & 0 & 0 & 0 & 0 & 0 \\ H_1^T & H_0^T & 0 & 0 & 0 & 0 & 0 & 0 \\ H_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{12}$$

$$d_3 = \begin{bmatrix} f_3 \\ 0 \\ 0 \\ 0 \\ f_2 \\ 0 \\ f_1 \\ f_0 \end{bmatrix} \tag{13}$$

As can been seen the $C$ matrix doubles in size each iteration. The size of $C_i$ is $(n2^i) \times (n2^i)$.

## Limitations

QR factorization on the ith set of constraints will be very expensive because the number of non-zeros will be $O(n2^i)$.

The eventual solve will not necessarily behave well because the constraints are not full rank.