Gradient Projection Algorithm

This Hypercard stack was prepared by:
Dennis L. Bricker,
Dept. of Industrial Engineering,
University of Iowa,
Iowa City, Iowa 52242
e-mail: dbricker@icaen.uiowa.edu

author

Consider the problem of optimizing a nonlinear function subject to linear constraints:

$$\begin{aligned} &\text{Minimize } f(x) \\ &\text{subject to} \\ &\qquad Ax \le b' \\ &\qquad Ex = b'' \end{aligned}$$

Suppose that

$$Mx^o = b$$

and that we wish to choose a step direction d so that

$$M(x^o + d) = b$$

Then

$$Mx^o + Md = b$$
$$b + Md = b$$
$$Md = 0$$

i.e., d lies in the null space of the matrix M:

$$\{ x \mid Mx = 0 \}$$

Assume that M is $m \times n$ with full row rank, i.e., rank M = m, so that there are no linearly dependent rows.

Then MM^T is $m \times m$ with rank m, so that MM^T is nonsingular, i.e., $(MM^T)^{-1}$ exists.

Consider Py, where y is an arbitrary vector and

$$\boxed{P = I - M^T\left(MM^T\right)^{-1} M}$$

Consider the matrix P defined by

$$\boxed{P = I - M^T\left(MM^T\right)^{-1}M}$$

for arbitrary matrix M with full row rank.

This is a *projection* matrix if

- $P = P^T$ *i.e., P is symmetric*
- $PP = P$ *i.e., the projection of the projection of a vector is the same as the projection!*

To show: • $P = P^T$

$$P^T = \left[I - M^T\left(MM^T\right)^{-1} M\right]^T$$

$$= I - \left[M^T\left(MM^T\right)^{-1} M\right]^T$$

$$= I - M^T\left(\left(MM^T\right)^{-1}\right)^T\left(M^T\right)^T$$

$$= I - M^T\left[\left(M^T\right)^T M^T\right]^{-1} M$$

$$= I - M^T\left(MM^T\right)^{-1} M = P$$

To show: • $PP = P$

$$PP = \left(I - M^T(MM^T)^{-1}M\right)\left(I - M^T(MM^T)^{-1}M\right)$$

$$= I - 2M^T(MM^T)^{-1}M + M^T(MM^T)^{-1}\underbrace{MM^T(MM^T)^{-1}}_{I}M$$

$$= I - 2M^T(MM^T)^{-1}M + M^T(MM^T)^{-1}\,I\,M$$

$$= I - M^T(MM^T)^{-1}\,M = P$$

What is the result of multiplying the projection Py by M?

$$M\,(Py) = M\left(I - M^T(MM^T)^{-1}M\right)y$$

$$= My - MM^T(MM^T)^{-1}My$$

$$= My - I\,My = 0$$

That is, P projects y onto the null space of the matrix M.

Given an initial feasible solution X^o, Rosen's Gradient Projection Algorithm projects the steepest descent direction onto the null space of the tight constraints, so that the resulting direction will be feasible.

Rosen's Gradient Projection Algorithm

Step 0 Choose an initial feasible point x^o, and let $Mx^o = b$ be the tight constraints. Let $k=0$.

Step 1 If M is vacuous (# rows =0), let P=I; otherwise, let

$$P = I - M^T\left(MM^T\right)^{-1}M$$

Step 2 Let $d = -P\,\nabla f(x^k)$ be the search direction.

If $d^k = 0$, then let $\lambda = -\left(MM^T\right)^{-1} M\,\nabla f(x^k)$

LAGRANGE MULTIPLIERS

If $\lambda_i \geq 0$ for each inequality $M_i x \leq b_i$

STOP; x^k is optimal.

Otherwise, if $\lambda_i < 0$ for some inequality $M_i x \leq b_i$ remove row i from the matrix M of binding constraints, and return to step 1.

Step 3 Perform a one-dimensional search to

$$\begin{array}{l} \text{Minimize } f(x^k + t\,d^k) \\ \text{subject to } 0 \leq t \leq t_{max} \end{array}$$

where

$$t_{max} = \min \left\{ \hat{b}_j / \hat{d}_j \mid \hat{b}_j > 0,\ \hat{d}_j > 0 \right\}$$

$$\hat{b} = b - Ax^k,\ \hat{d} = Ad^k$$

Let $x^{k+1} = x^k + t^* d^k$, increment k, let $Mx^k = b$ be the new binding constraints, and go to step 1.

Suppose that $d = P\left(-\nabla f(x)\right) = 0$

i.e.,

$$-\left(I - M^T\left(MM^T\right)^{-1}M\right)\nabla f(x) = 0$$

$$\Longrightarrow \quad -\nabla f(x) + M^T\left(MM^T\right)^{-1}M\nabla f(x) = 0$$

$$\Longrightarrow \quad -\nabla f(x) - M^T\lambda = 0$$

$$\Longrightarrow \quad -\nabla f(x) = \sum_i M_i\lambda_i$$

where $\lambda = -\left(MM^T\right)^{-1}M\ \nabla f(x^k)$

and M_i = row #i of matrix M

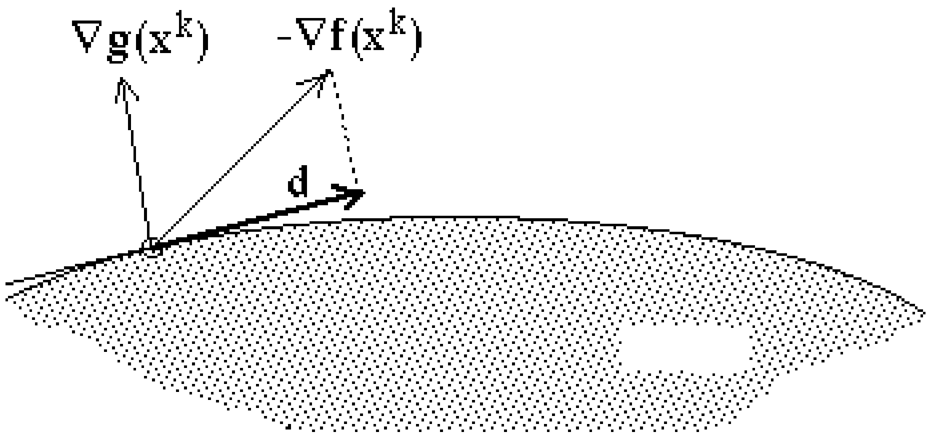
= gradient of tight constraint # i

Then x^k & λ satisfy the K-K-T conditions if $\lambda_i \geq 0$ for each inequality $M_i x \leq b_i$

If $\lambda_i < 0$ for inequality $M_i x \leq b_i$, then the steepest descent direction $-\nabla f(x)$ points *into* the feasible region of this constraint, so that the constraint should be made loose, i.e., non-binding.

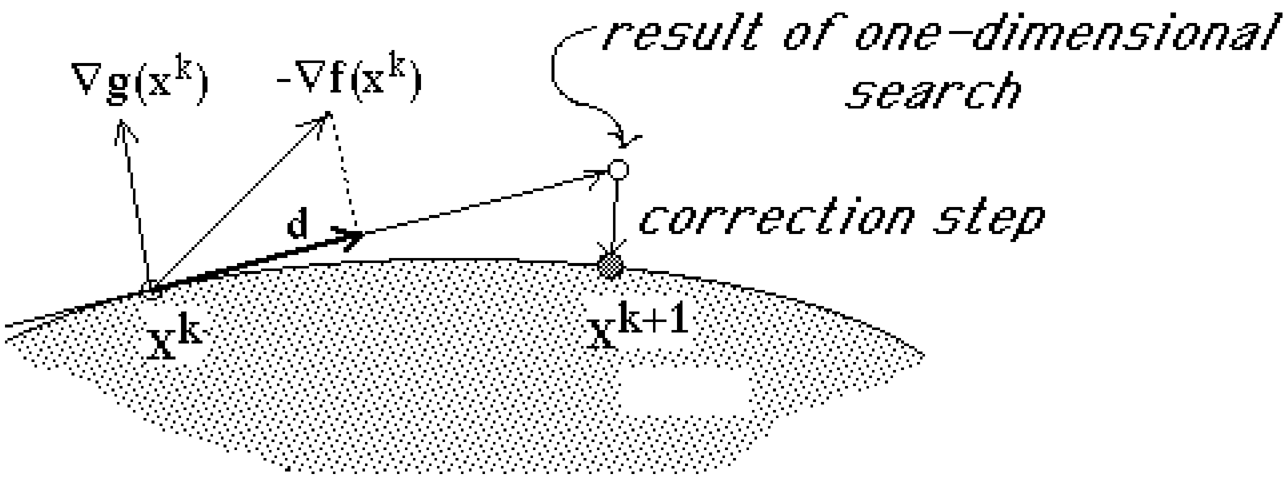
Nonlinear Constraints

If the constraints are nonlinear, then the steepest descent direction should be projected onto the null space of the gradients of the tight constraints to obtain the search direction.



$$\boxed{d = -P\nabla f(x^k)}$$

After the one-dimensional search in this direction, a "correction" step will generally be required to regain feasibility *(e.g., using Newton-Raphson method.)*



EXAMPLE

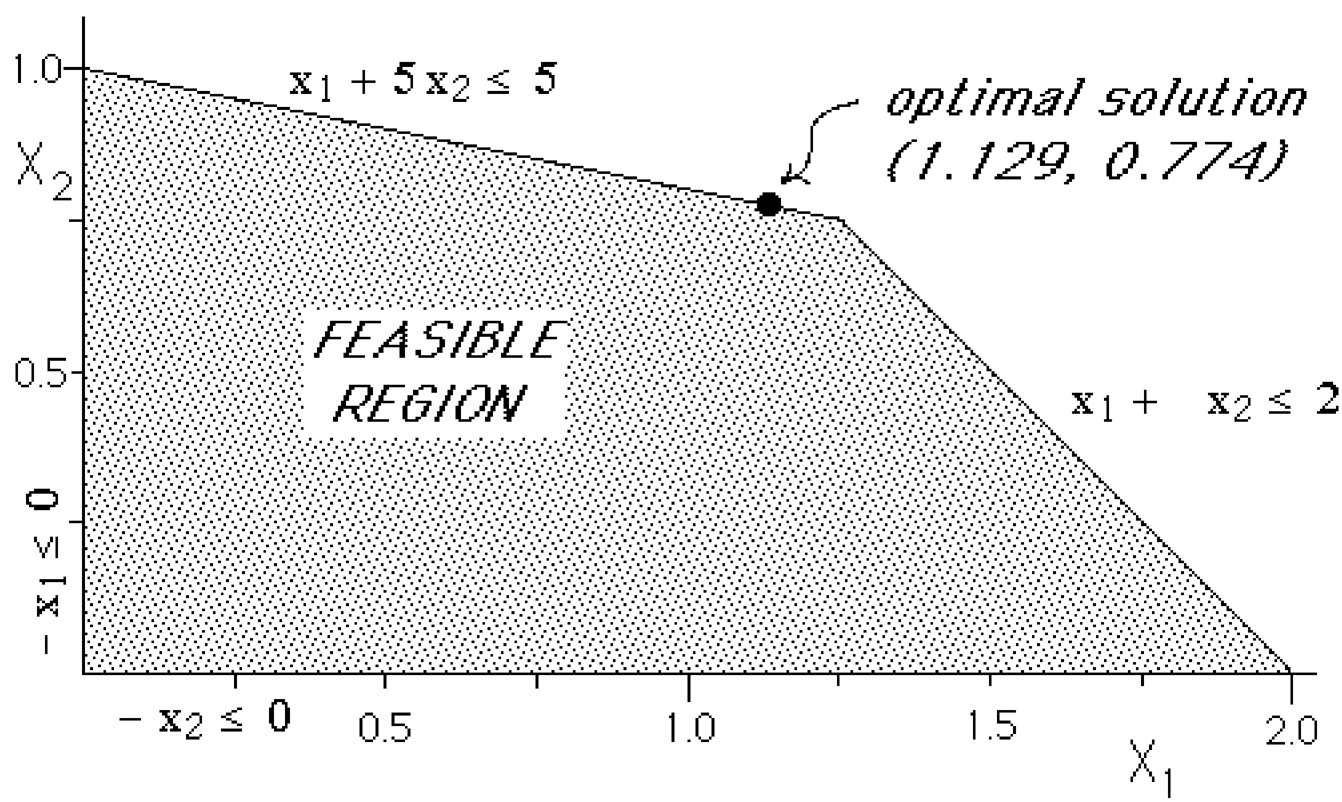
$$\text{Minimize } 2x_1^2 + 2x_2^2 - 2x_1x_2 - 4x_1 - 6x_2$$

subject to

$$\begin{aligned} x_1 + \quad x_2 &\le 2 \\ x_1 + 5x_2 &\le 5 \\ -x_1 \qquad &\le 0 \\ -x_2 &\le 0 \end{aligned}$$

NONNEGATIVITY CONSTRAINTS

Actually, this is a QP problem, and would be better solved by a QP algorithm!



Objective

```
Z←F X;Q
⍝
⍝        Sample NLP problem for
⍝     Gradient Projection Algorithm
⍝     (Quadratic objective function)
⍝
Q←2 2⍴2 ¯1 ¯1 2
Z←(X+.×Q+.×X)+(¯4 ¯6+.×X)
```

Gradient of objective

```
G←GRADIENT X;Q
⍝
⍝     Gradient for objective function
⍝          of Sample NLP problem
⍝
Q←2 2⍴2 ¯1 ¯1 2
G←(2×Q+.×X)+(¯4 ¯6)
```


Inequality Constraints

1	1	≤	2
1	5	≤	5
¯1	0	≤	0
0	¯1	≤	0

Equality Constraints

——None——

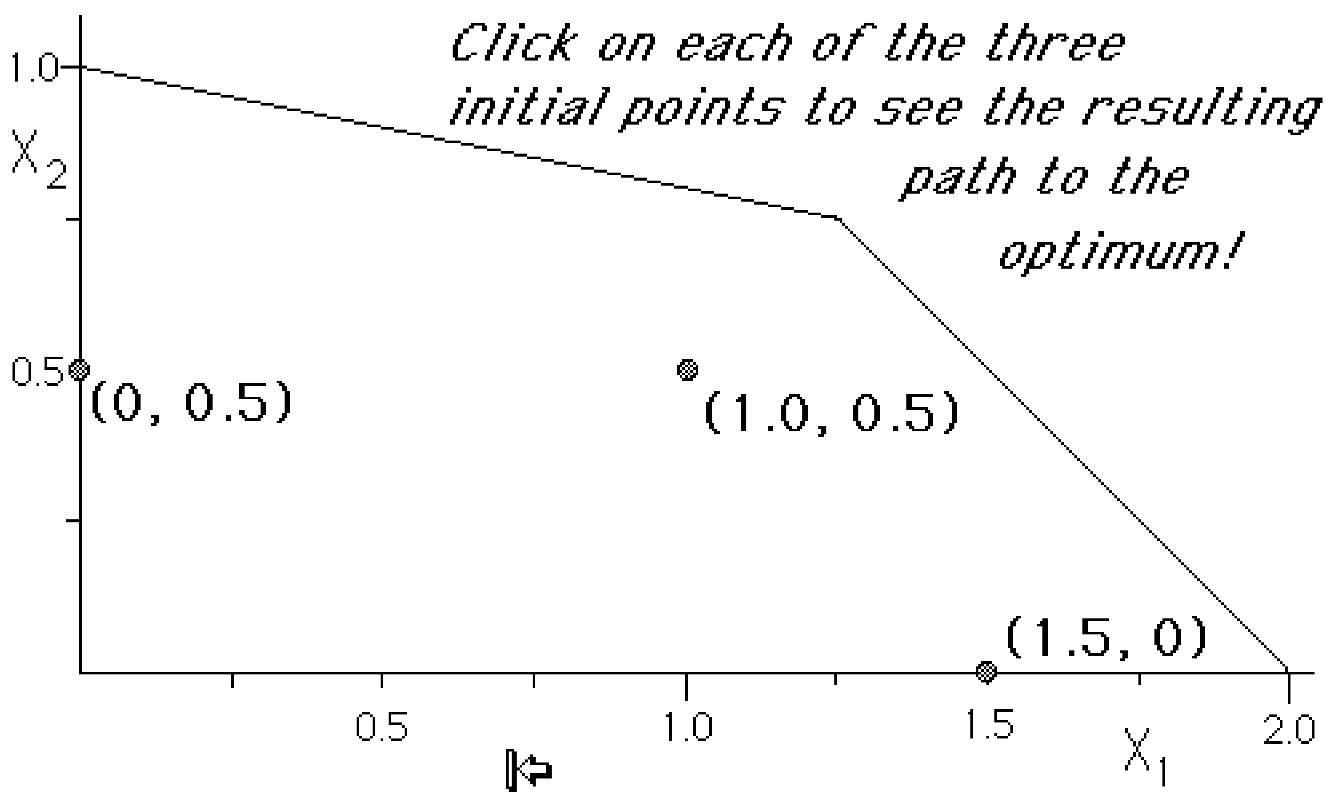
Gradient Projection Algorithm

Tolerance for search direction = 0.001
(TOL ≥ +/|Projection of $\nabla f(x)$ onto tight constraints)

Maximum # of Gradient Projection iterations = 25

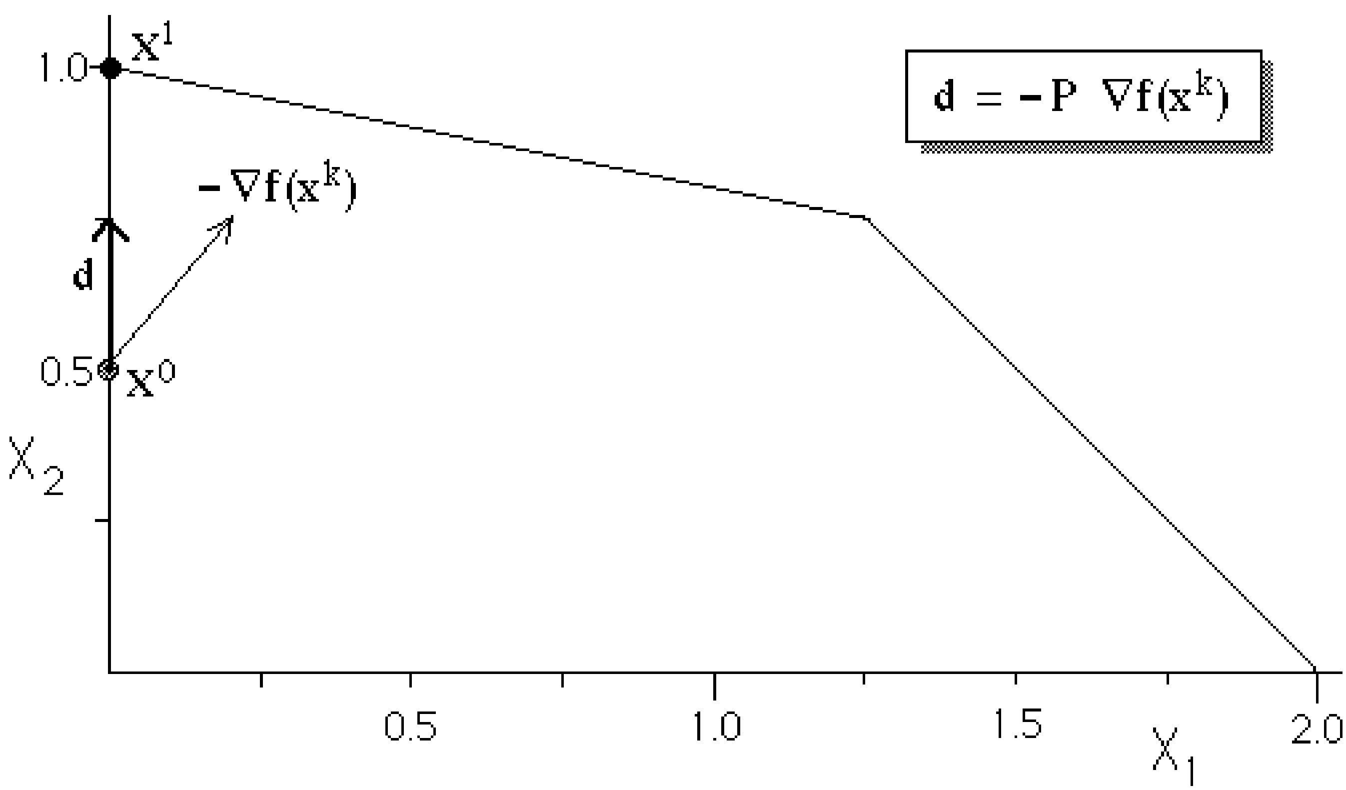
Tolerance for one-dimensional searches = 0.005

Maximum # of 1-dimensional search iterations = 25



Iteration 1

```
X= 0 0.5
F(X) = ¯2.5
Slack variables: 1.5 2.5 0 0.5

Constraint Partition: Tight: 3          Slack: 1 2 4
Matrix M =
                              ¯1   0

Projection Matrix P =
                              0 0
                              0 1

Gradient ∇f(x) = ¯5 ¯4

Search Direction = 0 4
Maximum step size = 0.125
Optimal step size = 0.125
```

Iteration 2

X= 0 1
F(X) = ⁻4
Slack variables: 1 0 0 1

Constraint Partition: Tight: 2 3 Slack: 1 4

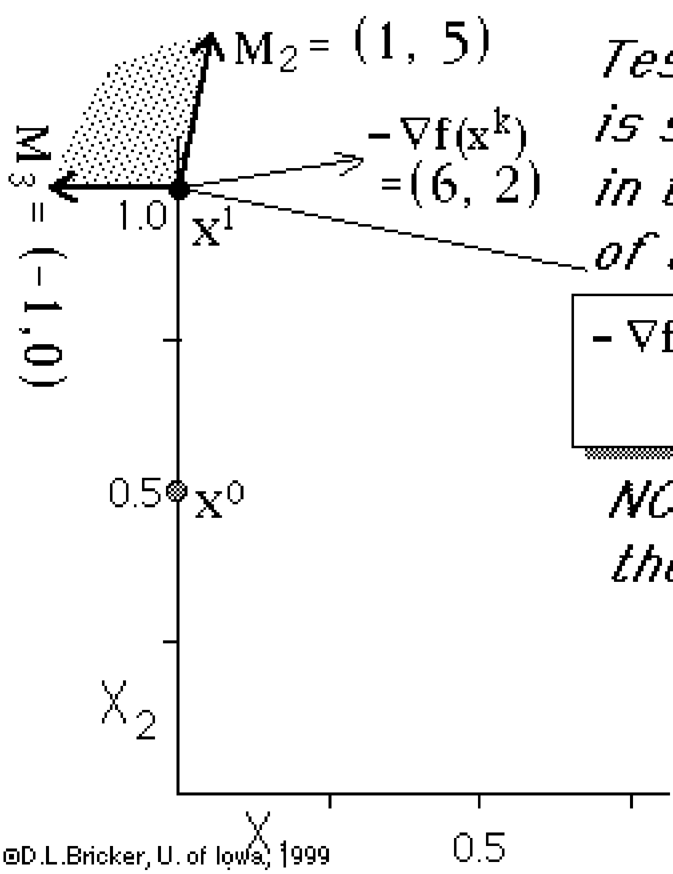
Matrix M =

$$\begin{bmatrix} 1 & 5 \\ -1 & 0 \end{bmatrix}$$

Projection Matrix P =

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Gradient ∇f(x) = ⁻6 ⁻2
Search Direction = 0 0 = d



Test K-K-T conditions....
is steepest descent direction
in the cone of the gradients
of the tight constraints?

$$\boxed{\begin{array}{c} -\nabla f(x^1) = \lambda_2 M_2 + \lambda_3 M_3 \\ \lambda_2 \geq 0,\ \lambda_3 \geq 0 \end{array}}$$

NO! Therefore, "release"
the offending constraint!

$$\begin{bmatrix} 6 \\ 2 \end{bmatrix} = 0.4 \begin{bmatrix} 1 \\ 5 \end{bmatrix} - 5.6 \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

©D.L.Bricker, U. of Iowa, 1999

Test K-K-T conditions....

$$\boxed{\begin{array}{c} -\nabla f(x^1) = \lambda_2 M_2 + \lambda_3 M_3 \\ \lambda_2 \geq 0, \lambda_3 \geq 0 \end{array}}$$

$$\begin{bmatrix} 6 \\ 2 \end{bmatrix} = 0.4 \begin{bmatrix} 1 \\ 5 \end{bmatrix} - 5.6 \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

negative!

```
Lagrange Multipliers = 0.4 ¯5.6
***Release Tight Constraint 2
Constraint Partition: Tight: 2          Slack: 1 4 3
```



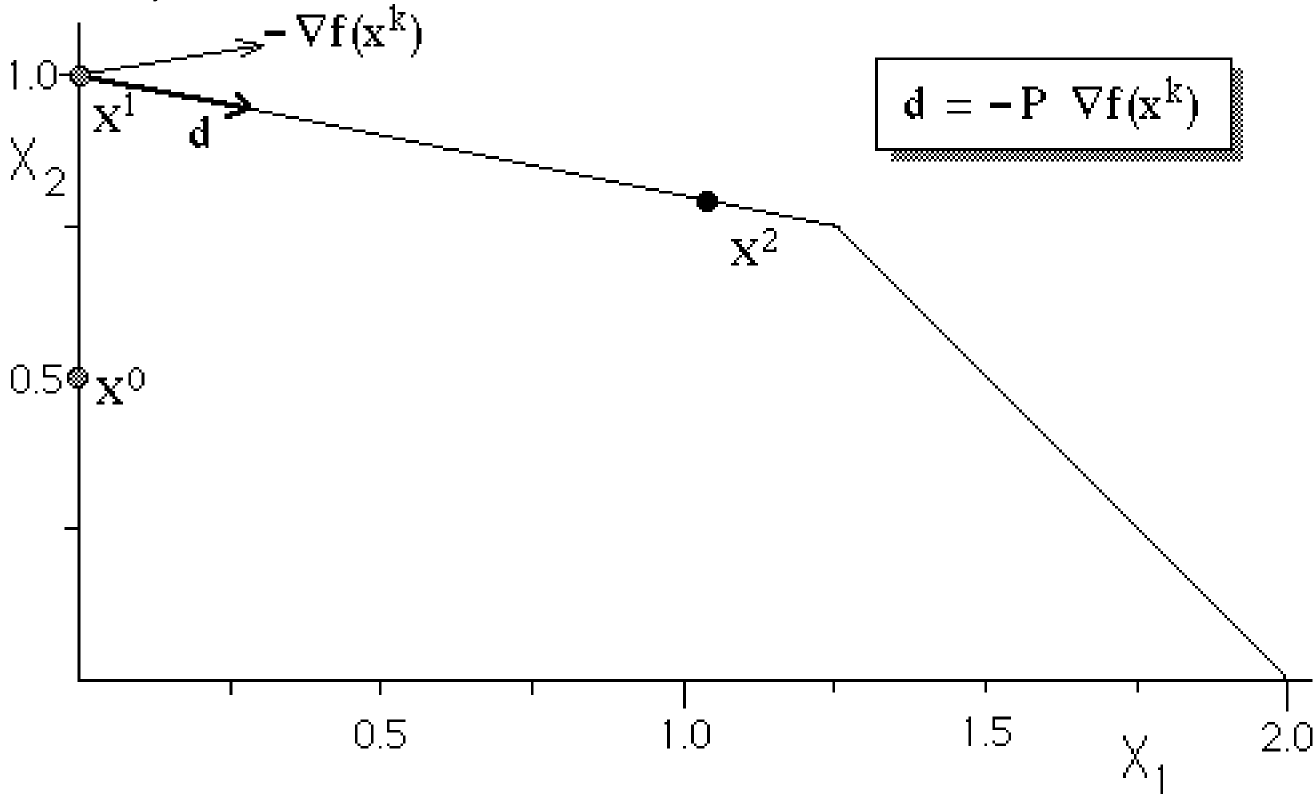
```
X= 0 1
F(X) = ¯4
Slack variables: 1 0 0 1

Constraint Partition: Tight: 2          Slack: 1 4 3
Matrix M =
```

$$\begin{bmatrix} 1 & 5 \end{bmatrix}$$

```
Projection Matrix P =
```

$$\begin{bmatrix} 0.961538 & ¯0.192308 \\ ¯0.192308 & 0.0384615 \end{bmatrix}$$

```
Gradient ∇f(x) = ¯6 ¯2
Search Direction = 5.38462 ¯1.07692

Maximum step size = 0.232143
Optimal step size = 0.209677
```



Iteration 3

```
X= 1.12903 0.774194
F(X) = ¯7.16129
Slack variables: 0.0967742 0 1.12903 0.774194

Constraint Partition: Tight: 2          Slack: 1 3 4
Matrix M =
```

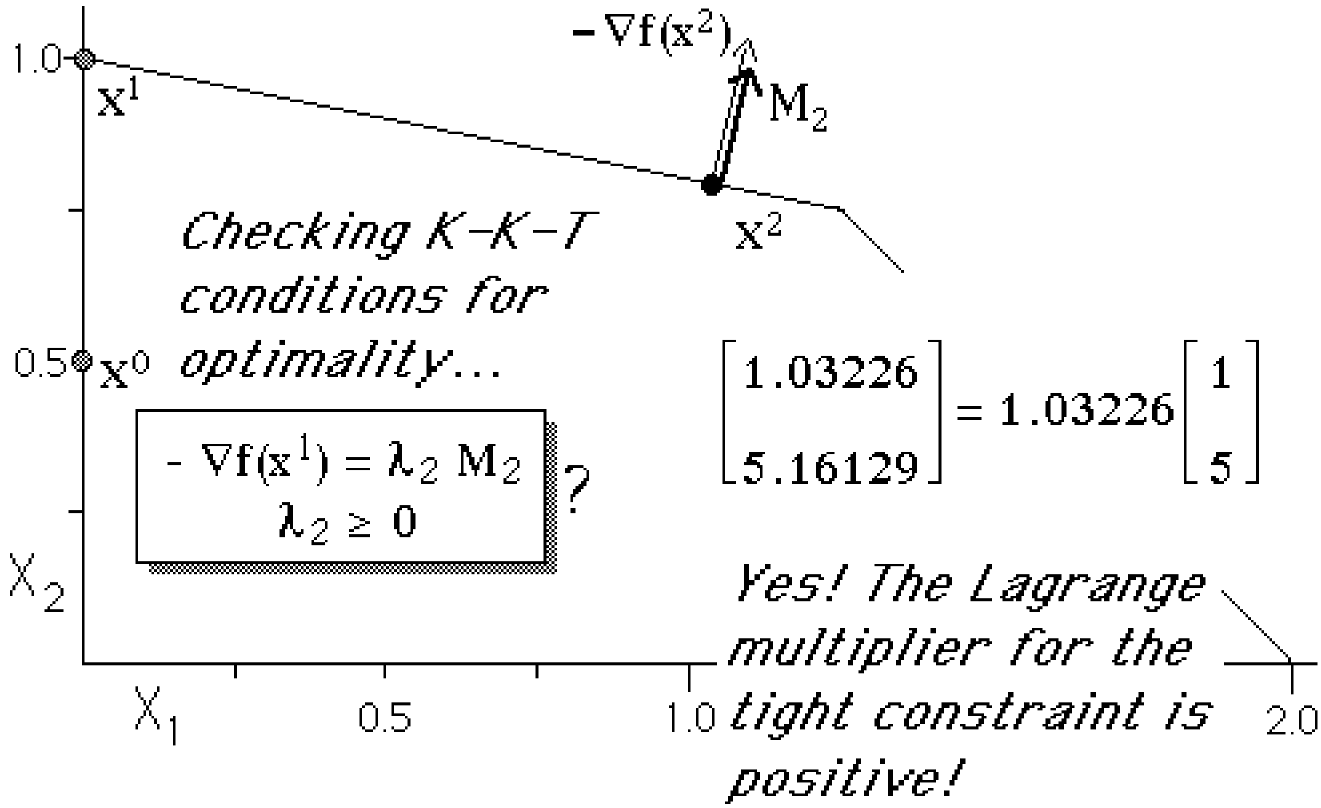
$$\begin{bmatrix} 1 & 5 \end{bmatrix}$$

Projection Matrix P =

$$\begin{bmatrix} 0.961538 & ^{-}0.192308 \\ ^{-}0.192308 & 0.0384615 \end{bmatrix}$$

Gradient $\nabla f(x)$ = ¯1.03226 ¯5.16129

Search Direction = 3.33067E¯16 ¯2.22045E¯16 = $d \approx 0$




```
Lagrange Multipliers = 1.03226

***Optimality Conditions Satisfied
     X= 1.12903 0.774194
     F(X) = ¯7.16129

Lagrange Multipliers= 1.03226
Slack in inequality constraints: 0.0967742 0 1.12903 0.774
```

Iteration 1

```
X= 1 0.5
F(X) = ¯5.5
Slack variables: 0.5 1.5 1 0.5
Constraint Partition: Tight: none          Slack: 1 2 3 4
Matrix M =

Gradient ∇f(x) = ¯1 ¯6
Search Direction = 1 6

Maximum step size = 0.0483871
Optimal step size = 0.0483871
```


Iteration 2

```
X= 1.04839 0.790323
F(X) = ¯7.14516
Slack variables: 0.16129 0 1.04839 0.790323

Constraint Partition: Tight: 2          Slack: 1 3 4
Matrix M =
```

$$\begin{bmatrix} 1 & 5 \end{bmatrix}$$

Projection Matrix P =

$$\begin{bmatrix} 0.961538 & ¯0.192308 \\ ¯0.192308 & 0.0384615 \end{bmatrix}$$

```
Gradient ∇f(x) = ¯1.3871 ¯4.93548
Search Direction = 0.384615 ¯0.0769231

Maximum step size = 0.524194
Optimal step size = 0.209677
```

©D.L.Bricker, U. of Iowa, 1999

Iteration 3

X= 1.12903 0.774194
F(X) = ¯7.16129
Slack variables: 0.0967742 0 1.12903 0.774194

Constraint Partition: Tight: 2 Slack: 1 3 4
Matrix M =

$$\begin{bmatrix} 1 & 5 \end{bmatrix}$$

Projection Matrix P =

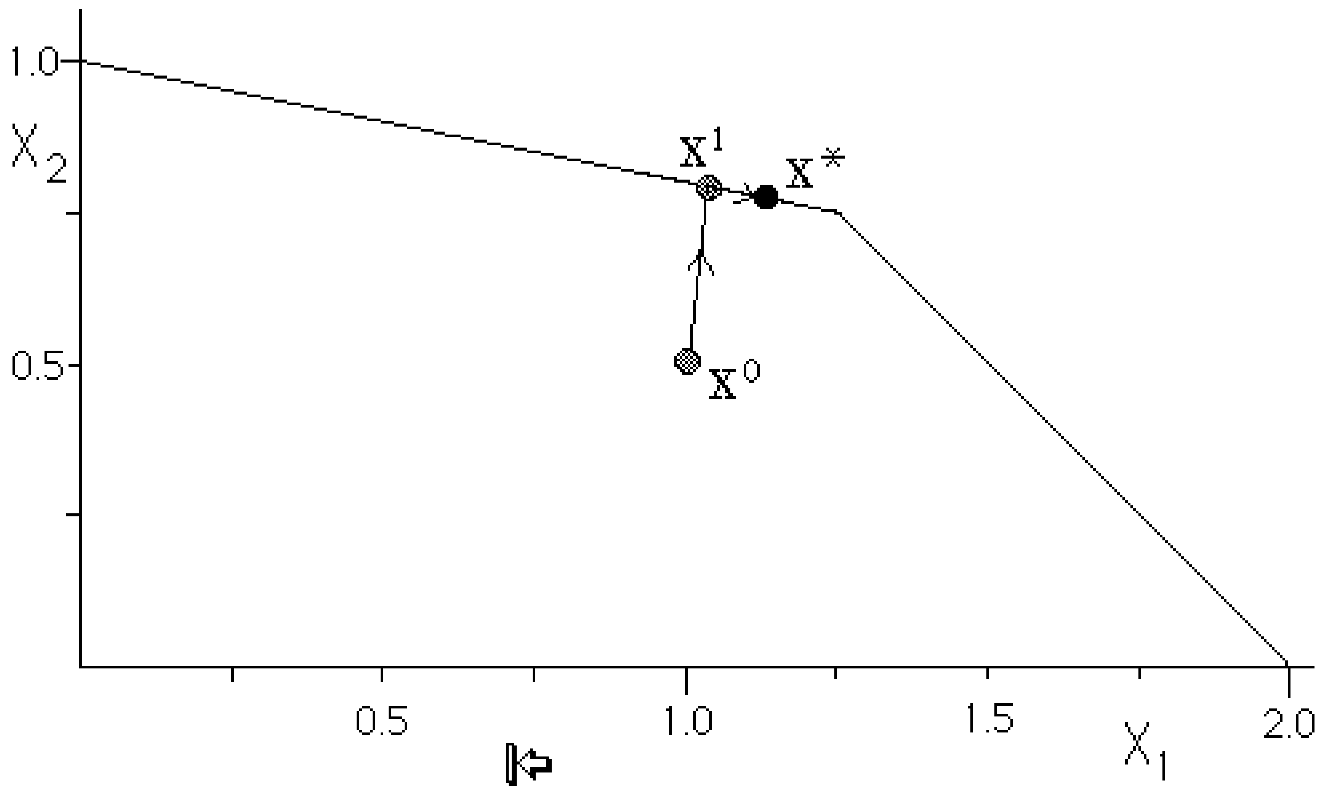
$$\begin{bmatrix} 0.961538 & ^{-}0.192308 \\ ^{-}0.192308 & 0.0384615 \end{bmatrix}$$

Gradient ∇f(x) = ¯1.03226 ¯5.16129
Search Direction = 3.33067E¯16 ¯2.22045E¯16 ≈ 0
Lagrange Multipliers = 1.03226

***Optimality Conditions Satisfied
X= 1.12903 0.774194
F(X) = ¯7.16129

Lagrange Multipliers= 1.03226

Slack in inequality constraints: 0.0967742 0 1.12903 0.7741



Iteration 1

X= 1.5 0
F(X) = ¯1.5
Slack variables: 0.5 3.5 1.5 0

Constraint Partition: Tight: 4 Slack: 1 2 3
Matrix M =

$$\begin{bmatrix} 0 & ^{-}1 \end{bmatrix}$$

Projection Matrix P =

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

Gradient ∇f(x) = 2 ¯9
Search Direction = ¯2 0

Maximum step size = 0.75
Optimal step size = 0.25

Iteration 2

X= 1 0
F(X) = ¯2
Slack variables: 1 4 1 0

Constraint Partition: Tight: 4 Slack: 1 2 3
Matrix M =

$$\begin{bmatrix} 0 & -1 \end{bmatrix}$$

Projection Matrix P =

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

Gradient ∇f(x) = 0 ¯8
Search Direction = 0 0

Lagrange Multipliers = ¯8

***Release Tight Constraint 1
Constraint Partition: Tight: none Slack: 1 2 3 4

X= 1 0
F(X) = ¯2
Slack variables: 1 4 1 0

Constraint Partition: Tight: none Slack: 1 2 3 4
Matrix M =

Gradient ∇f(x) = 0 ¯8
Search Direction = 0 8

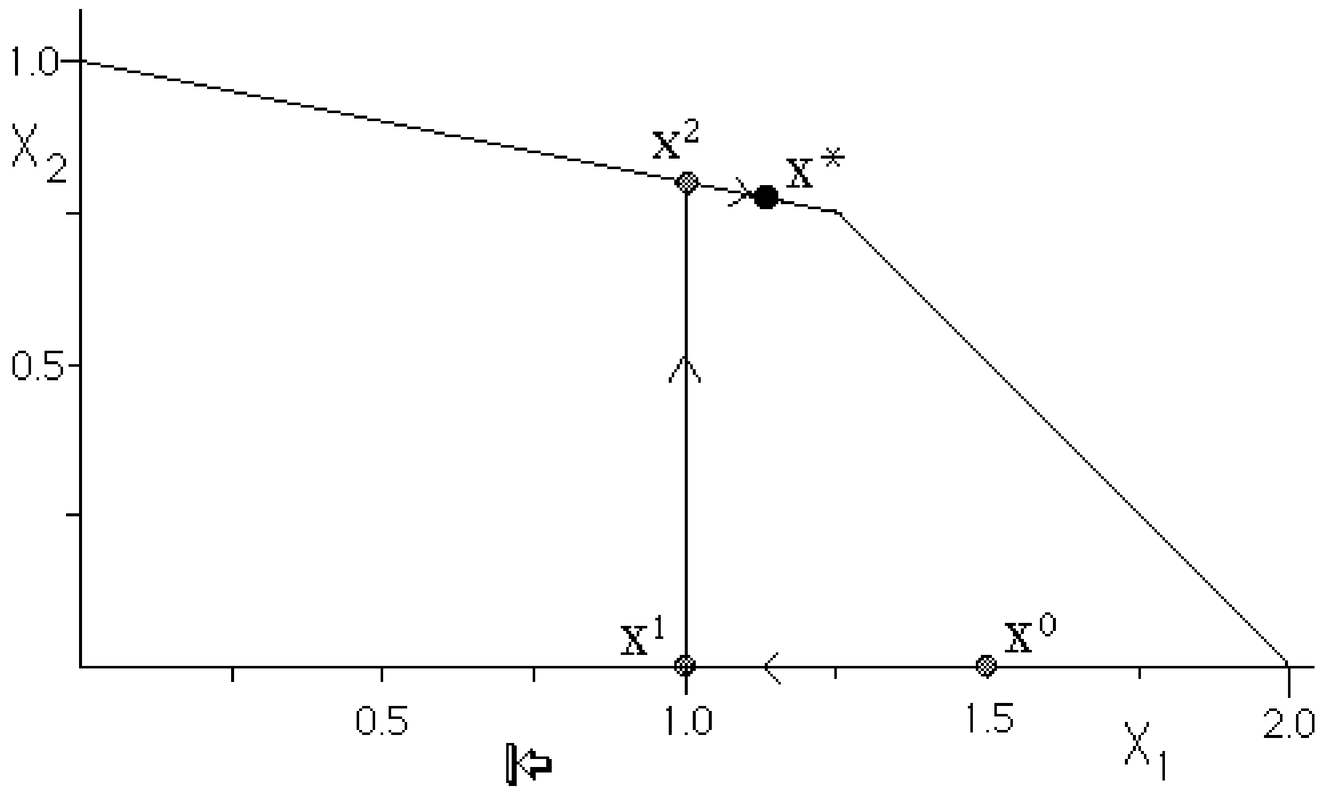
Maximum step size = 0.1
Optimal step size = 0.1

Iteration 3

```
X= 1 0.8
F(X) = ¯7.12
Slack variables: 0.2 0 1 0.8
Constraint Partition: Tight: 2          Slack: 1 3 4
Matrix M =
                             1  5

Projection Matrix P =
                      0.961538 ¯0.192308
                     ¯0.192308  0.0384615
Gradient ∇f(x) = ¯1.6 ¯4.8
Search Direction = 0.615385 ¯0.123077
Maximum step size = 0.40625
Optimal step size = 0.209677
```


Iteration 4

```
X= 1.12903 0.774194
F(X) = ¯7.16129
Slack variables: 0.0967742 0 1.12903 0.774194

Constraint Partition: Tight: 2          Slack: 1 3 4
Matrix M =
                             1  5

Projection Matrix P =
                      0.961538 ¯0.192308
                     ¯0.192308  0.0384615

Gradient ∇f(x) = ¯1.03226 ¯5.16129
Search Direction = 4.77396E¯15 ¯1.08247E¯15  ≈ 0
Lagrange Multipliers = 1.03226

***Optimality Conditions Satisfied
X= 1.12903 0.774194
F(X) = ¯7.16129

Lagrange Multipliers= 1.03226
Slack in inequality constraints: 0.0967742 0 1.12903 0.774
```


1.0
X_2
0.5
x^2
x^*
x^1
x^0
0.5
1.0
1.5
2.0
X_1