FEASIBLE DIRECTION ALGORITHM for Constrained NLP

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 constrained nonlinear programming problem

$$\begin{aligned}\text{Minimize} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \le 0,\ i=1,\cdots m \\ & x \in R^n\end{aligned}$$

Suppose that x^t is the current iterate in a search algorithm.

Expand each function in a Taylor Series at x^t, ignoring terms higher than first order:

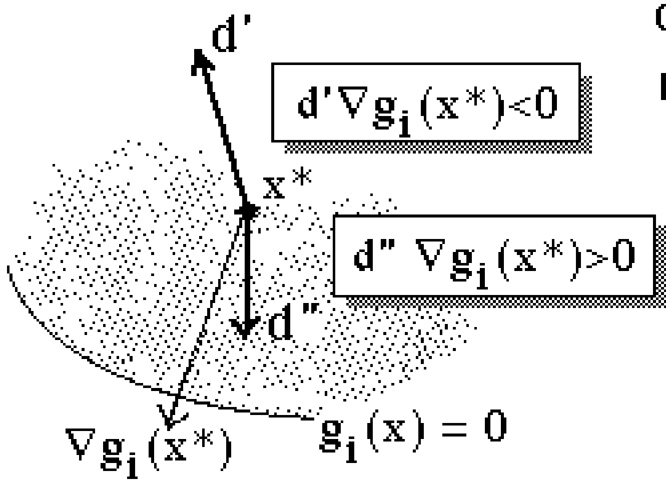
$$f(x^t + \lambda d) \approx f(x^t) + \lambda \nabla f(x^t) \bullet d$$
$$g_i(x^t + \lambda d) \approx g_i(x^t) + \lambda \nabla g_i(x^t) \bullet d$$

where
$\lambda \ge 0$ is a scalar
d is a vector.

A **FEASIBLE DIRECTION** d must satisfy:

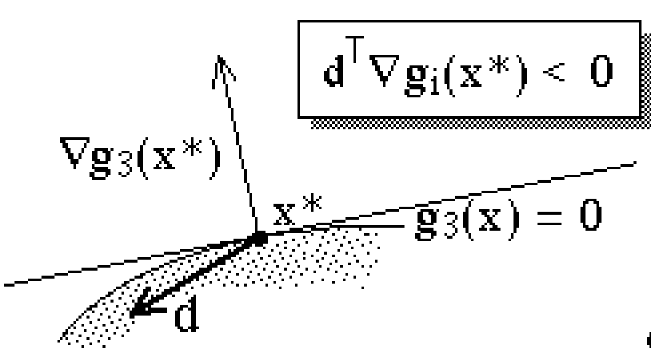
$$g_i(x^t + \lambda d) \le 0 \text{ for sufficiently "small" } \lambda > 0$$

If $g_i(x^t) < 0$, then *any* direction is feasible with respect to constraint i.

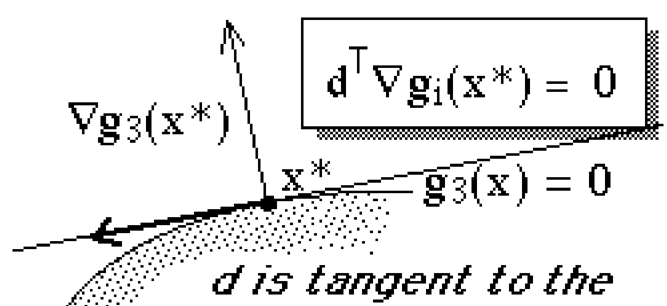


FEASIBLE DIRECTION

If $g_i(x^t) = 0$, then d should satisfy $\nabla g_i(x^t) \bullet d < 0$



d is a feasible direction



d is tangent to the constraint boundary

d is "feasible", but any positive step in this direction may be infeasible

FEASIBLE DIRECTION

Let $I = \{i \mid g_i(x^t)=0\}$

index set of tight constraints at x^t

and $D = \{d \mid \nabla g_i(x^t)\bullet d < 0 \ \forall\ i\in I\}$

set of feasible directions at x^t

DESCENT DIRECTION

$$f(x^t + \lambda d) \approx f(x^t) + \lambda \nabla f(x^t)\bullet d$$

$$f(x^t + \lambda d) < f(x^t) \Rightarrow \lambda \nabla f(x^t)\bullet d < 0$$

To be a descent direction, d must satisfy

$$\nabla f(x^t)\bullet d < 0$$

Let $F_0 = \{d \mid \nabla f(x^t)\bullet d < 0\}$

set of descent directions at x^t

If x^* is an optimal solution to

Minimize $f(x)$
subject to $g_i(x) \leq 0$, $i=1,2,\ldots m$

then

The directional derivative of $f(x)$ is nonnegative in every feasible direction at x^*

i.e., there should be no feasible direction which is also a descent direction!

i.e., $F_0 \cap D = \varnothing$

A Necessary Condition for Optimality of a point x^t is

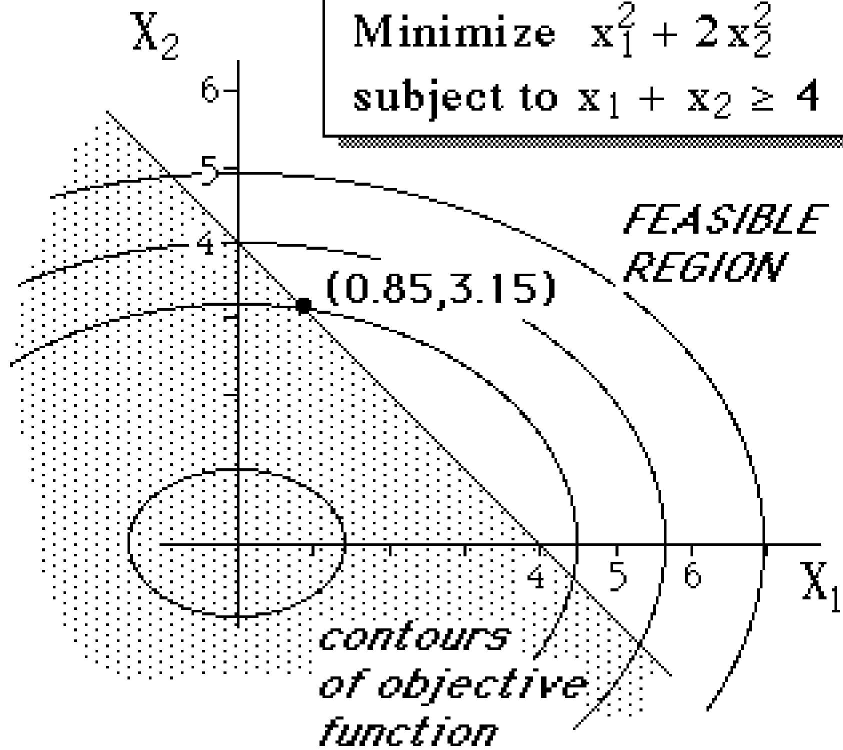
$$F_0 \cap D = \varnothing$$

How can we easily test this optimality condition at x^t?

EXAMPLE

Minimize $x_1^2 + 2x_2^2$
subject to $x_1 + x_2 \geq 4$

Suppose that we wish to test the point $X^0 = (0.85, 3.15)$ for optimality:



EXAMPLE

Minimize $x_1^2 + 2x_2^2$
subject to $x_1 + x_2 \geq 4$

$$\Rightarrow \begin{cases} f(x) = x_1^2 + 2x_2^2, \quad g(x) = 4 - x_1 - x_2 \leq 0 \\ \nabla f(x) = \begin{bmatrix} 2x_1 \\ 4x_2 \end{bmatrix}, \quad \nabla g(x) = \begin{bmatrix} -1 \\ -1 \end{bmatrix} \end{cases}$$

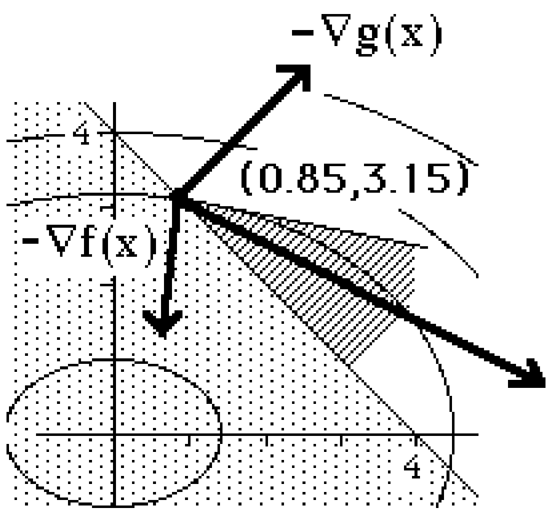
$$d \in F_0 \Leftrightarrow \nabla f(x^t) \bullet d = [1.7,\ 12.6]\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} < 0$$

$$d \in D \Leftrightarrow \nabla g(x^t) \bullet d = [-1,\ -1]\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} < 0$$

That is,

$$d \in F_0 \cap D \quad \Leftrightarrow \quad \begin{cases} 1.7d_1 + 12.6d_2 < 0 \\ -d_1 \quad - d_2 < 0 \end{cases}$$

Is there such a direction d?



$$\begin{cases} 1.7d_1 + 12.6d_2 < 0 \\ -d_1 \quad - d_2 < 0 \end{cases}$$

for directions in the shaded cone

We wish to search for a feasible solution to the system of (strict) inequalities:

$$\begin{cases} 1.7d_1 + 12.6d_2 < 0 \\ -d_1 \qquad -d_2 < 0 \end{cases}$$

This could be done by, for example, solving the linear programming problem:

$$\text{Maximize } z$$
$$\text{s.t.} \begin{cases} 1.7d_1 + 12.6d_2 + z \leq 0 \\ -d_1 \qquad -d_2 + z \leq 0 \end{cases}$$

(d_1, d_2, & z unconstrained in sign)

$$\text{Maximize } z$$
$$\text{s.t.} \begin{cases} 1.7d_1 + 12.6d_2 + z \leq 0 \\ -d_1 \qquad -d_2 + z \leq 0 \end{cases}$$

(d_1, d_2, & z unconstrained in sign)

If $z^*>0$ for some (d_1,d_2), then $(d_1,d_2) \in F_0 \cap D$

Furthermore, the LP will be unbounded above since K times (d_1,d_2) yields an objective value which is K times z^*.

Since we are concerned only with the *direction* and not the *magnitude* of (d_1,d_2), we add the "normalizing" constraints:

$$\text{Maximize } z$$
$$\text{s.t.} \begin{cases} 1.7d_1 + 12.6d_2 + z \le 0 \\ -d_1 \quad -d_2 + z \le 0 \\ -1 \le d_1 \le 1 \\ -1 \le d_2 \le 1 \end{cases}$$

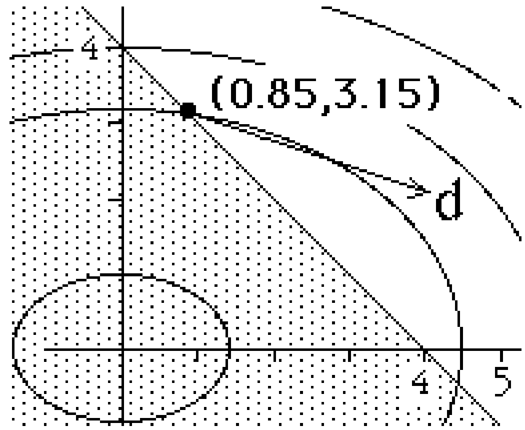
The optimal solution of the LP

$$\text{Maximize } z$$
$$\text{s.t.} \begin{cases} 1.7d_1 + 12.6d_2 + z \le 0 \\ -d_1 \quad -d_2 + z \le 0 \\ -1 \le d_1 \le 1 \\ -1 \le d_2 \le 1 \end{cases}$$

is $(d_1,d_2) = (1.0, -0.199)$, $z = +0.801 > 0$

Therefore, $x^0=(0.85, 3.15)$ is *not* optimal!

To find an improved solution, we next perform a one-dimensional search in the direction d^0



$$\text{Minimize } f(x^0 + \lambda\, d^0)$$
$$\lambda \geq 0$$

Iteration 1

```
X = 0.85 3.15
F(X) = 20.5675
∇F(X) = 1.7 12.6
G(X) = 0
Tight Constraints: 1
Jacobian of tight constraints =
¯1 ¯1

The Simplex Method with Upper Bounding is used to search for
a direction which is both feasible and improves the objective.

OBJECTIVE Z= 0.8014705882
Search Direction d = 1 ¯0.1985294118
Projections of Gradients onto Search Direction :
   Objective         : ¯0.8014705882
   Tight Constraints: ¯0.8014705882

No maximum stepsize
Optimal stepsize = 0.371454345
```

Iteration 2

```
X = 1.221454345 3.076255387
F(X) = 20.41864513
∇F(X) = 2.44290869 12.30502155
G(X) = ¯0.2977097324
Tight Constraints: None

The steepest descent direction, -∇f, is selected.
Search Direction d = ¯0.1985294118 ¯1

Projections of Gradients onto Search Direction :
   Objective          : ¯12.79001077
Computing Max α, starting at estimate α= 0.2483958503
   at which G(X+αd)= 0
Maximum stepsize = 0.2483958503

Optimal stepsize = 0.2483958503
```

Iteration 3

```
X = 1.172140463 2.827859537
F(X) = 17.36749239
∇F(X) = 2.344280926 11.31143815
G(X) = 0
Tight Constraints: 1
Jacobian of tight constraints =
¯1 ¯1

The Simplex Method with Upper Bounding is used to search for
a direction which is both feasible and improves the objective.

OBJECTIVE Z= 0.7283598483
Search Direction d = 1 ¯0.2716401517
Projections of Gradients onto Search Direction :
   Objective          : ¯0.7283598483
   Tight Constraints: ¯0.7283598483

No maximum stepsize
Optimal stepsize = 0.3173469017
```

Iteration 4

```
X = 1.489487365 2.741655377
F(X) = 17.25192102
∇F(X) = 2.978974729 10.96662151
G(X) = ¯0.2311427412
Tight Constraints: None

The steepest descent direction, -∇f, is selected.
Search Direction d = ¯0.2716401517 ¯1
Projections of Gradients onto Search Direction :
   Objective        : ¯11.77583065

Computing Max α, starting at estimate α= 0.1817674134
   at which G(X+αd)= 0
Maximum stepsize = 0.1817674134

Optimal stepsize = 0.1817674134
```

Iteration 5

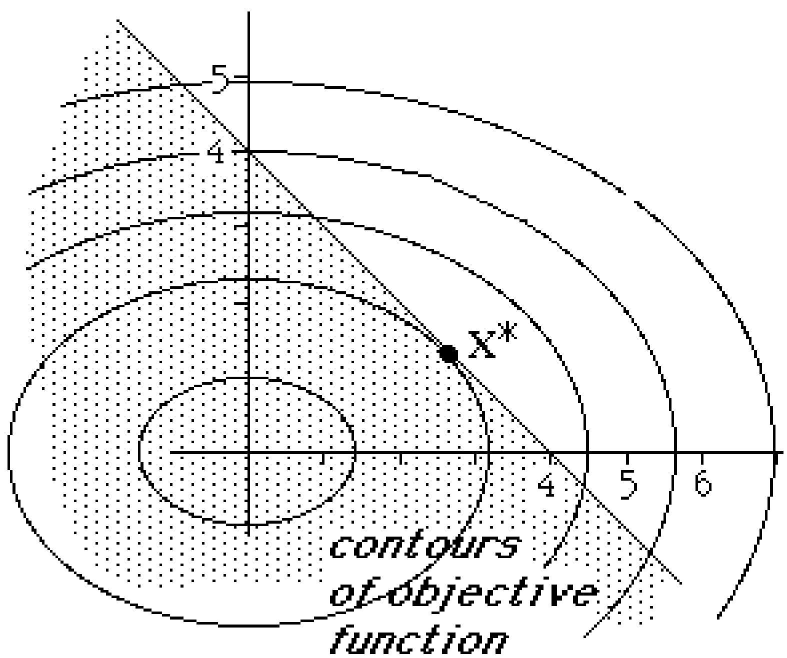
```
X = 1.440112037 2.559887963
F(X) = 15.17997545
∇F(X) = 2.880224074 10.23955185
G(X) = 0
Tight Constraints: 1
Jacobian of tight constraints =
¯1 ¯1

The Simplex Method with Upper Bounding is used to search for
a direction which is both feasible and improves the objective.

OBJECTIVE Z= 0.6547705705
Search Direction d = 1 ¯0.3452294295
Projections of Gradients onto Search Direction
   Objective        : ¯0.6547705705                :
   Tight Constraints: ¯0.6547705705

No maximum stepsize
Optimal stepsize = 0.2643686079
```

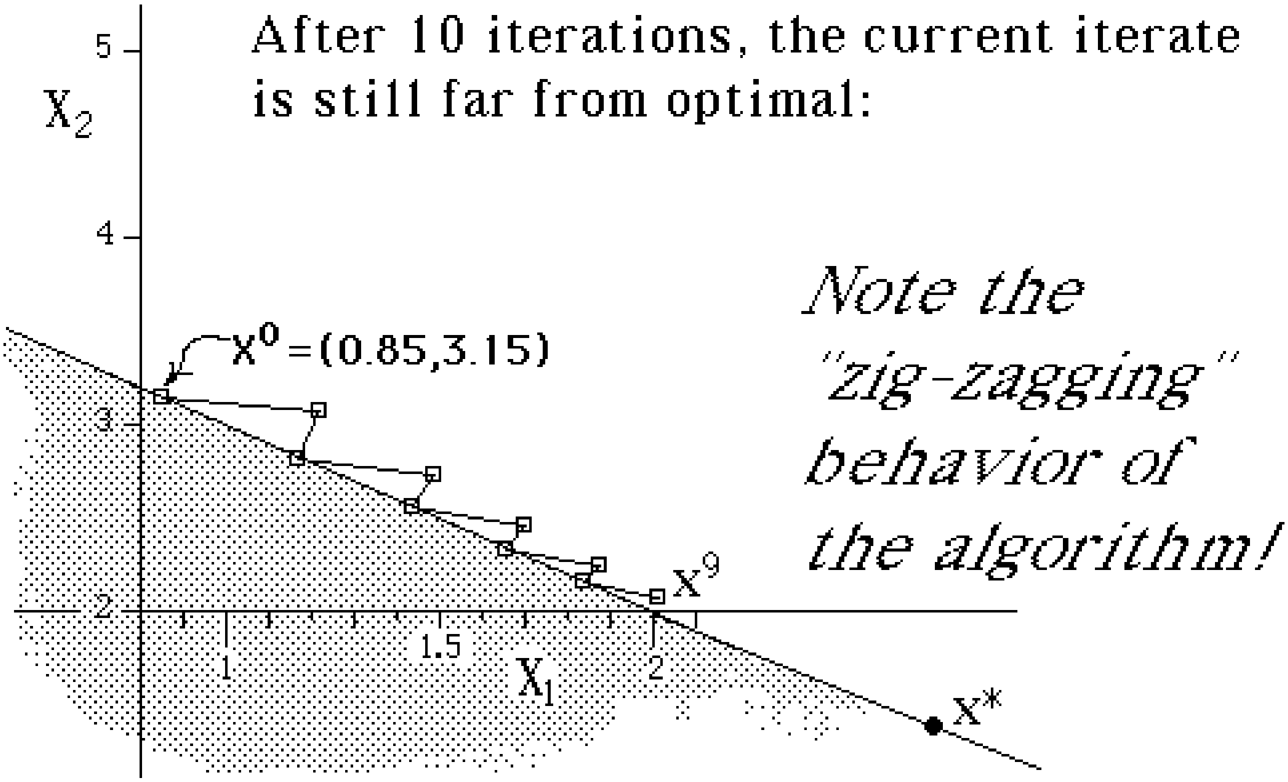
... etc.

```
Iteration 10

X = 2.014252 2.075857222
F(X) = 12.67557753
∇F(X) = 4.028504 8.303428887
G(X) = ¯0.0901092216
Tight Constraints: None

The steepest descent direction, -∇f, is selected.
Search Direction d = ¯0.4851614983 ¯1
Projections of Gradients onto Search Direction :
   Objective         : ¯10.25790392

Computing Max α, starting at estimate α= 0.06067301213
   at which G(X+αd)= 0
Maximum stepsize = 0.06067301213

Optimal stepsize = 0.06067301213
```

The optimal solution can be easily found by solving the KKT conditions:

$x^* = (8/3, 4/3)$





Feasible Directions Algorithm

Weights: 1
3/30/94 12:15

Iteration 1

i	X[i]
1	8.50000E¯1
2	3.15000E0

i	Gi(X)
1	0.00000E0 ←

F(X) = 20.5675
∇F(X) = 1.7 12.6

Tight Constraints: 1
Jacobian of tight constraints =

¯1 ¯1

The Simplex Method with Upper Bounding is used to search for
a direction which is both feasible and improves the objective.

```
LP tableau                 ¯1   ¯1    1.414213562 1 0 ¯1
----------                  1.7 12.6 12.71416533  0 1  1

Costs & Bounds              i     1  2    3   4   5    6
--------------              C[i]  0  0    1   0   0 ¯999
                            L[i] ¯1 ¯1 ¯999   0   0    0
                            U[i]  1  1  999 999 999  999
```

```
Optimal LP objective Z= 0.3569878012
Search Direction d = 1 ¯0.4951430099
Projections of Gradients onto Search Direction :
   Objective        : ¯4.538801925
   Tight Constraints: ¯0.5048569901
No maximum stepsize
Optimal stepsize = 1.522747371
```

Iteration 2

i	X[i]
1	2.37275E0
2	2.39602E0

i	Gi(X)
1	¯7.68770E¯1

```
F(X) = 17.11177565
∇F(X) = 4.745494741 9.584089134

Tight Constraints: None

The steepest descent direction, -∇f, is selected.
Search Direction d = ¯0.4951430099 ¯1
Projections of Gradients onto Search Direction :
   Objective        : ¯11.93378768
Computing Max α, starting at estimate α= 0.5141780078
   at which G(X+αd)= 0
0.5141780078
Maximum stepsize = 0.5141780078
Optimal stepsize = 0.5141780078
```

i	X[i]
1	2.11816E0
2	1.88184E0

i	Gi(X)
1	0.00000E0 ←

F(X) = 11.56925943
∇F(X) = 4.236311448 7.527377103

Tight Constraints: 1
Jacobian of tight constraints =
¯1 ¯1

The Simplex Method with Upper Bounding is used to search for a direction which is both feasible and improves the objective.

LP tableau

¯1	¯1	1.414213562	1	0	¯1
4.236311448	7.527377103	8.637577249	0	1	1

Costs & Bounds

i	1	2	3	4	5	6
C[i]	0	0	1	0	0	¯999
L[i]	¯1	¯1	¯999	0	0	0
U[i]	1	1	999	999	999	999

Optimal LP objective Z= 0.1706727894
Search Direction d = 1 ¯0.7586322265
Projections of Gradients onto Search Direction :
Objective : ¯1.474199403
Tight Constraints: ¯0.2413677735
No maximum stepsize
Optimal stepsize = 0.3426704035

Iteration 4

i	X[i]
1	2.46083E0
2	1.62188E0

i	Gi(X)
1	¯8.27096E¯2

F(X) = 11.31667718

∇F(X) = 4.921652255 6.487533858

Tight Constraints: None

The steepest descent direction, -∇f, is selected.
Search Direction d = ¯0.7586322265 ¯1
Projections of Gradients onto Search Direction :
   Objective : ¯10.22125787
Computing Max α, starting at estimate α= 0.04703063614
   at which G(X+αd)= 0
0.04703063614
Maximum stepsize = 0.04703063614
Optimal stepsize = 0.04703063614

Iteration 5

i	X[i]
1	2.42515E0
2	1.57485E0

i	Gi(X)
1	0.00000E0 ←

F(X) = 10.84166167
∇F(X) = 4.850294343 6.299411314

Tight Constraints: 1
Jacobian of tight constraints =
¯1 ¯1

The Simplex Method with Upper Bounding is used to search for
a direction which is both feasible and improves the objective.

LP tableau

¯1	¯1	1.414213562	1	0	¯1
4.850294343	6.299411314	7.950342013	0	1	1

Costs & Bounds

i	1	2	3	4	5	6
C[i]	0	0	1	0	0	¯999
L[i]	¯1	¯1	¯999	0	0	0

... etc.

After ten iterations, the current iterate is

i	X[i]
1	2.65348E0
2	1.34762E0

i	Gi(X)
1	¯1.10437E¯3

F(X) = 10.67313859

∇F(X) = 5.306969566 5.390478345

Tight Constraints: None