# Problem Set #2

Bennett Hellman & Benjamin Siegel
15.093 - Optimization Methods
MIT

October 12, 2021

## Problem 1: Bertsimas Tsitsiklis, Exercise 2.4

This is false because when converting a polyhedron to standard form, we often increase the dimensionality and force $x_1, x_2, ...x_n \geq 0$, making it impossible for a line to exist without intersecting one of the constraints. This allows a polyhedron that contains a line to be converted into standard form and have a vertex. For example consider the polyhedron $P = \{(x_1, x_2) : 0 \leq x_1 \leq 1, x_2 \geq 0\}$ which is the set $\{x \in R^n : Ax \geq b, x \geq 0\}$ with:

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

$$b = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

Clearly the polyhedron $P$ contains a line and therefore does not contain a vertex. When $P$ is converted into standard form by adding slack variables, the new polyhedron $P = \{x \in R^n : Ax = b, x \geq 0\}$ with:

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

$$b = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

When converted to standard form, this standard form polyhedron contains a vertex because it is nonempty.

## Problem 2: Bertsimas Tsitsiklis, Exercise 4.1

$$\begin{aligned} \min \quad & x_1 - x_2 \\ \text{s.t.} \quad & 2x_1 + 3x_2 - x_3 + x_4 \leq 0 \\ & 3x_1 + x_2 + 4x_3 - 2x_4 \geq 3 \\ & -x_1 - x_2 + 2x_3 + x_4 = 6 \\ & x_1 \leq 0 \\ & x_2, x_3 \geq 0 \end{aligned} \tag{1}$$

In dual form:

$$\begin{aligned}
\max \quad & 0p_1 + 3p_2 + 6p_1 \\
\text{s.t.} \quad & 2p_1 + 3p_2 - p_3 \geq 1 \\
& 3p_1 + p_2 - p_3 \leq -1 \\
& -p_1 + 4p_2 + 2p_3 \leq 0 \\
& p_1 - 2p_2 + p_3 = 0 \\
& p_1 \leq 0 \\
& p_2 \geq 0 \\
& p_3 \; free
\end{aligned} \tag{2}$$

# Problem 3:

(a)
Primal problem:

$$\begin{aligned}
\min \quad & c'x + d'y \\
\text{s.t.} \quad & Ax + By = b \\
& Dx \leq f \\
& Gy \leq g
\end{aligned} \tag{3}$$

Dual form:

$$\begin{aligned}
\max \quad & p_A'b + p_B'f + p_C'g \\
\text{s.t.} \quad & p_A'A_j + p_B'D_j = c_j \\
& p_A'B_j + p_C'G_j = d_j \\
& p_A \; free \\
& p_B \leq 0 \\
& p_C \leq 0
\end{aligned} \tag{4}$$

(b) As seen in the attached Julia code, both the primal problem and the dual problem have the same optimal cost of -236.41.

# Problem 4:

(a)

$$\min_{p \in P_d} ||f(x) - p(x)||_\infty$$

$$= \min_{p \in P_d} \max_{x_i \in [a,b]} |f(x_i) - p(x_i)|$$

Let $z = max_{x_i \in [a,b]} |f(x_i) - p(x_i)|$
Then the linear programming problem becomes:

$$\begin{aligned}
& \min_{p \in P_d} z \\
\text{s.t.} \quad & f(x_i) - p(x_i) \leq z \quad \forall x_i \in [a,b] \\
& -f(x_i) + p(x_i) \leq z \quad \forall x_i \in [a,b]
\end{aligned} \tag{5}$$

Notice the polynomial approximation of degree $d$ can be written as $p(x) = \sum_{j=0}^{d} a_j x^j$. Therefore the final linear programming formulation can be written as follows:

$$\begin{aligned} & \min z \\ \text{s.t.} \quad & z + \sum_{j=0}^{d} a_j x_i^j \geq f(x_i) \quad \forall\, x_i \in [a, b] \\ & z - \sum_{j=0}^{d} a_j x_i^j \geq -f(x_i) \quad \forall\, x_i \in [a, b] \end{aligned} \tag{6}$$

(b)
We introduce the dual variables $p_1$ $p_2$ and formulate the dual as follows:

$$\begin{aligned} & \max \sum_{i=1}^{N} p_{1i} f(x_1) - p_{2i} f(x_1) \\ \text{s.t.} \quad & \sum_{i=1}^{N} p_{1i} + p_{2i} = 1 \\ & \sum_{i=1}^{N} x_i^j p_{1i} - x_i^j p_{2i} = 0 \\ & p_1, p_2 \geq 0 \end{aligned} \tag{7}$$

(c)
The complementary slackness conditions state:

$$\begin{aligned} p_{1i}(z + \sum_{j=0}^{d} a_j x_i^j - f(x_i)) = 0 \quad \forall x_i \in [a, b] \\ p_{2i}(z - \sum_{j=0}^{d} a_j x_i^j + f(x_i)) = 0 \quad \forall x_i \in [a, b] \end{aligned} \tag{8}$$

According to these conditions we have $z = |p(x_i) - f(x_i)|$ for all $x_i$ according to the original formulation, and the dual conditions ensure the value for $z$ is optimally minimized.

# Problem 5: Bertsimas Tsitsiklis, Exercise 4.28

We must demonstrate statements (a) and (b) are equivalent.
$(a \to b)$
Assume statement (a) is true.
$\forall x \geq 0 \quad a'x' \leq \max_i a_i' x$
Let $\lambda_j = 0 \quad \forall j \neq i$ and $\lambda_i = 1$ where $i$ corresponds to $\max_i a_i x$
Therefore the coefficients $\lambda_i$ sum to 1.
Furthermore, $\sum_{j=1}^{m} \lambda_j a_j x = \max_i a_i x$

By substitution:
$a'x \leq \sum_{j=1}^{m} \lambda_j a_j x$
Equivalently:
$a \leq \sum_{j=1}^{m} \lambda_j a_j$
Therefore statement (b) is true

(b $\rightarrow a$)
Assume statement (b) is true.
There exist nonnegative coefficients $\lambda_i$ that sum to 1 and such that $a \leq \sum_{i=1}^{m} \lambda_i a_i$
$a \leq \sum_{i=1}^{m} \lambda_i a_i$ can be written equivalently as $a'x \leq \sum_{i=1}^{m} \lambda_i a_i x$ for all $x \geq 0$
Since a convex combination of a set is always less than or equal to the maximum, we know that $\sum_{i=1}^{m} \lambda_i a_i x \leq \max_i a_i x$
It follows that $a'x \leq \sum_{i=1}^{m} \lambda_i a_i x \leq \max_i a_i x$
By the transitive property, $a'x \leq \max_i a_i x$
Therefore statement (a) is true.

In [94]:

```julia
using Pkg, JLD2, JuMP, Clp
```

In [95]:

```julia
@load "data.jld2" c d A B b D f G g;
```

## Primal Solution

In [98]:

```julia
model = Model(Clp.Optimizer)
### Add variables
@variables(
    model,
    begin
        x[1:50]
        y[1:100]
    end
)
### Add objective
@objective(model,Min,sum(c[i]*x[i] for i in 1:50)+sum(d[i]*y[i] for i in 1:100))
### Add constraints
@constraints(
    model,
    begin
        [i = 1:25], sum(A[i,j]*x[j] for j in 1:50)+sum(B[i,j]*y[j] for j in 1:100) == b[i]
        [i = 1:100], sum(D[i,j]*x[j] for j in 1:50) <= f[i]
        [i = 1:200], sum(G[i,j]*y[j] for j in 1:100) <= g[i]
    end
)
### Optimize
optimize!(model)
```

```
Coin0506I Presolve 325 (0) rows, 150 (0) columns and 28750 (0) elements
Clp0006I 0  Obj 0 Primal inf 120927.15 (168) Dual inf 1.9596901 (150) w.o. free dual inf (0)
Clp0006I 1  Obj -5.0099709e+11 Primal inf 2.0081889e+14 (180) Dual inf 58.390467 (58)
Clp0006I 1  Obj -1.0849018e+12 Primal inf 2.9161142e+14 (190) Dual inf 5.1032289e+15 (150) w.o. free dual inf (1)
Clp0006I 34  Obj -6.0819552e+11 Primal inf 8.6956047e+13 (130) Dual inf 9.6045993e+15 (129) w.o. free dual inf (13)
Clp0006I 87  Obj -2.5466723e+11 Primal inf 3.7086919e+13 (95) Dual inf 2.7712908e+15 (89) w.o. free dual inf (26)
Clp0006I 138  Obj -244.48593 Primal inf 34273.727 (94) Dual inf 3.2151979e+15 (59) w.o. free dual inf (47)
Clp0006I 182  Obj -222.64466 Primal inf 241.53227 (29) Dual inf 3.2276703e+16 (61)
Clp0006I 228  Obj -219.43154 Primal inf 22.297114 (8) Dual inf 3.8065899e+14 (58)
Clp0006I 250  Obj -218.15476 Dual inf 16.036848 (62)
Clp0006I 300  Obj -228.60093 Dual inf 15.17668 (55)
Clp0006I 346  Obj -234.63638 Dual inf 3.9697542 (55)
Clp0006I 402  Obj -236.41355
Clp0000I Optimal - objective value -236.41355
Clp0032I Optimal objective -236.4135515 - 402 iterations time 0.062
```

In [99]:
```
objective_value(model)
```

Out[99]:
```
-236.413551485672
```

## Dual Solution

In [100…
```
dual = Model(Clp.Optimizer)
### Add variables
@variables(
    dual,
    begin
        pA[1:25]
        pB[1:100]
        pC[1:200]
    end
)
### Add objective
@objective(dual,Max,sum(pA[i]*b[i] for i in 1:25)+sum(pB[i]*f[i] for i in 1:100)+sum(pC[i]*g[i] for i in 1:200))
### Add constraints
@constraints(
    dual,
    begin
        [i = 1:50], sum(pA[j]*A[j,i] for j in 1:25) + sum(pB[j]*D[j,i] for j in 1:100) == c[i]
        [i = 1:100], sum(pA[j]*B[j,i] for j in 1:25) + sum(pC[j]*G[j,i] for j in 1:200) == d[i]
        [i = 1:100], pB[i] <= 0
        [i = 1:200], pC[i] <= 0
    end
)
### Optimize
optimize!(dual)
```

```
Coin0506I Presolve 150 (-300) rows, 325 (0) columns and 28750 (-300) elements
Clp0006I 0  Obj -0 Primal inf 1656.4483 (150) Dual inf 16688.358 (168) w.o. free dual inf (143)
Clp0006I 31  Obj 8.8164995e+13 Primal inf 8.2203334e+14 (132)
Clp0006I 67  Obj 5.745542e+13 Primal inf 6.1960457e+14 (117)
Clp0006I 119  Obj 3.0280359e+13 Primal inf 5.2002966e+14 (91)
Clp0006I 156  Obj 1.531674e+13 Primal inf 1.3538655e+14 (82)
Clp0006I 187  Obj 6.7440736e+12 Primal inf 5.7533014e+13 (73)
Clp0006I 218  Obj 1.6080855e+12 Primal inf 3.1041927e+13 (63)
Clp0006I 251  Obj 2.2880244e+11 Primal inf 3.2538525e+12 (63)
Clp0006I 287  Obj 2.0290044e+10 Primal inf 1.061127e+12 (61)
Clp0006I 322  Obj -222.38096 Primal inf 82.972122 (65)
Clp0006I 359  Obj -230.30942 Primal inf 35.984235 (61)
Clp0006I 394  Obj -233.14203 Primal inf 33.517978 (61)
Clp0006I 426  Obj -235.61676 Primal inf 15.443534 (53)
Clp0006I 461  Obj -237.39974 Primal inf 0.93337196 (20)
Clp0006I 469  Obj -237.45307
Clp0006I 469  Obj -6.332118e+10 Primal inf 2.5754025e+12 (63) Dual inf 2.4142535e+16 (92) w.o. free dual inf (88)
```

```
Clp0006I 482  Obj -237.47487 Dual inf 876.79434 (71)
Clp0006I 514  Obj -236.41355
Clp0000I Optimal - objective value -236.41355
Coin0511I After Postsolve, objective -236.41355, infeasibilities - dual 0 (0), primal 0 (0)
Clp0032I Optimal objective -236.4135515 - 514 iterations time 0.052, Presolve 0.00
```

In [101…

```
objective_value(dual)
```

Out[101…

```
-236.41355148567138
```