Mathematical Programming: Modeling and Applications

Giacomo Nannicini

giacomon@lix.polytechnique.fr

Optimal Rocket Control

- A rocket of mass m is launched at sea level and has to reach an altitude H at time T
- Let $y(t)$ be the altitude of the rocket at time t, and $u(t)$ the force acting on t in the vertical direction
- Assume $u(t) \leq b$ for $0 \leq t \leq T$, that the mass of the rocket is constant (i.e. fuel consumption is negligible), and the gravity acceleration g is constant

Optimal Rocket Control

- Discretize the time interval $[0, T]$ in n subintervals
- Formulate a linear program to determine, for each $k \leq n$, the force $u(t_k)$ acting on the rocket at time t_k so that the total consumed energy (i.e. $\int_0^T |u(t)|dt$) is minimum
- Solve the problem with data: m = 2140kg, H = 23km, T = 1min, b = 100000N, n = 20

Solution: Nonlinear Model

- The equation of motion of the rocket is:

$$\forall t \in [0, T] \; m\frac{\partial^2 y(t)}{\partial t^2} + mg = u(t)$$

- Let $v(t) = \frac{\partial y(t)}{\partial t}$ be the velocity at time t
- At time 0, we have $y(0) = v(0) = 0$
- At time T, we have $y(T) = H$
- We also have $|u(t)| \leq b \; \forall t$

Solution: Nonlinear Model

- We obtain:

$$
\begin{aligned}
\min \quad & \int_0^T |u(t)| dt \\
\forall t \in [0, T] \quad & |u(t)| \leq b \\
\forall t \in [0, T] \quad & m \frac{\partial^2 y(t)}{\partial t^2} + mg = u(t) \\
\forall t \in [0, T] \quad & v(t) = \frac{\partial y(t)}{\partial t} \\
& y(0) = 0 \\
& y(T) = H \\
& v(0) = 0
\end{aligned}
$$

Solution: Linearization

- We discretize the interval $[0, T]$ into n subintervals with $t_1 = 0, \Delta t = T/n, t_{n+1} = T = t_n + \Delta t, t_k = t_1 + k\Delta t$ for each $k \leq n$
- Let $y_k = y(t_k), u_k = u(t_k)$ and $v_k = v(t_k)$
- The discretization allows us to approximate the derivative: $v_k = \frac{y_{k+1} - y_k}{\Delta t}$ for each $k \leq n$
- Similarly, $\frac{\partial^2 y(t)}{\partial t^2}|_{t_k}$ can be approximated with $\frac{v_{k+1} - v_k}{\Delta t}$

Solution: Linearization

- Substituting in the model:

$$
\begin{array}{rl}
\min & \sum_{k=1}^{n} |u_k| \\
\forall k \le n & y_{k+1} - y_k = v_k \Delta t \\
\forall k \le n & v_{k+1} - v_k = (\frac{u_k}{m} - g)\Delta t \\
\forall k \le n+1 & |u_k| \le b \\
 & y_1 = 0 \\
 & y_{n+1} = H \\
 & v_1 = 0 \\
\forall k \le n+1 & 0 \le y_k \le H \\
\forall k \le n+1 & v_k \ge 0
\end{array}
$$

Solution: Linearization

- Now we must linearize the objective function: $\sum_{k=1}^{n} |u_k|$
- Introduce n variables w_k and constraints: $w_k \geq u_k$, $w_k \geq -u_k$ for each $k \leq n$
- These constraints enforce $w_k \geq |u_k|$
- The objective function becomes: $\sum_{k=1}^{n} w_k$
- Since we are minimizing, this works: the optimal value will have exactly $w_k = |u_k|$ because $w_k > |u_k|$ has a larger objective value

Solution: Linear Model

$$
\begin{array}{rl}
\min & \displaystyle\sum_{k=1}^{n} w_k \\
\forall k \le n & y_{k+1} - y_k = v_k \Delta t \\
\forall k \le n & v_{k+1} - v_k = (\dfrac{u_k}{m} - g)\Delta t \\
 & y_1 = 0 \\
 & y_{n+1} = H \\
 & v_1 = 0 \\
\forall k \le n & 0 \le w_k \le b \\
\forall k \le n & -w_k \le u_k \le w_k \\
\forall k \le n+1 & 0 \le y_k \le H \\
\forall k \le n+1 & v_k \ge 0
\end{array}
$$

Solution: Model

```
## rocket.mod

# time horizon
param T >= 0, default 60;
# height to reach
param H >= 0, default 23000;
# mass of rocket
param m >= 0, default 2140;
# limit on force
param b >= 0, default 100000;
# number of time intervals
param n >= 0, default 20;
# gravity acceleration
param g default 9.8;
# Delta t
param Dt := T / n;

set N := 1..n+1;
set N1 := 1..n;
```

Solution: Model

```
# height
var y{N} >= 0, <= H;
# velocity
var v{N} >= 0;
# force
var u{N1};
# linearization
var w{N1} >= 0;

minimize energy : sum{k in N1} w[k];

subject to velocity {k in N1} : y[k+1] - y[k] = Dt*v[k];
subject to force {k in N1} : v[k+1] - v[k] = Dt*(u[k]/m - g);
subject to forcelimit {k in N1} : w[k] <= b;
subject to sealevel: y[1] = 0;
subject to height : y[n+1] = H;
subject to still: v[1] = 0;
subject to linearization1 {k in N1}: u[k] + w[k] >= 0;
subject to linearization2 {k in N1}: u[k] - w[k] <= 0;
```

Solution: Run File

```
# rocket.run
model rocket.mod;
option solver cplexamp;
solve;
display energy;
for {i in N1} {
    printf "%d %f %f %f\n", i, y[i], v[i], u[i];
}
printf "%d %f %f %f\n", n+1, y[n+1], v[n+1], 0;
```

Solution: Output

```
energy = 568112
1 0.000000 0.000000 100000.000000
2 0.000000 110.786916 100000.000000
3 332.360748 221.573832 100000.000000
4 997.082243 332.360748 100000.000000
5 1994.164486 443.147664 100000.000000
6 3323.607477 553.934579 68112.063492
7 4985.411215 620.018781 0.000000
8 6845.467557 590.618781 0.000000
9 8617.323899 561.218781 0.000000
10 10300.980240 531.818781 0.000000
11 11896.436582 502.418781 0.000000
12 13403.692924 473.018781 0.000000
13 14822.749266 443.618781 0.000000
14 16153.605607 414.218781 0.000000
15 17396.261949 384.818781 0.000000
16 18550.718291 355.418781 0.000000
17 19616.974633 326.018781 0.000000
18 20595.030975 296.618781 0.000000
19 21484.887316 267.218781 0.000000
20 22286.543658 237.818781 0.000000
21 23000.000000 208.418781 0.000000
```