ChE 597 Computational Optimization

Homework 10

April 5th 11:59 pm

1. Prove the convergence of the outer approximation algorithm for convex MINLP using the hint from the slides. You can assume that the NLP subproblems are always feasible.

2. The Generalized Benders cut we derived from the NLP subproblem

$$
\begin{aligned}
\text{NLP1} \quad \min_{x} \quad & f\left(x, y^k\right) \\
& g_i\left(x, y^k\right) \leq 0 \quad \forall i = 1, \ldots, m
\end{aligned}
$$

has the following form

$$
f(x^k, y^k) + \nabla_y f(x^k, y^k)^T (y - y^k) + \sum_{i=1}^{m} \mu_i^k \nabla_y g_i(x^k, y^k)^T (y - y^k) \leq \gamma
$$

However, if we formulate the NLP as

$$
\begin{aligned}
\text{NLP2} \quad \min_{x,y} \quad & f(x, y) \\
& y = y^k \\
& g_i(x, y) \leq 0 \quad \forall i = 1, \ldots, m
\end{aligned}
$$

It is easy to see that NLP2 is equivalent to NLP1. Now, derive GBD cuts based on NLP2.

Hint: follow the derivation of GBD cuts discussed in class. The stationarity conditions of NLP2 has to do both with the x and the y variables instead of just the x variables. You will see that you can simplify the GBD cuts by writing the NLP in the form of NLP2.

3. Consider the following convex MINLP

$$
\begin{aligned}
\min f = y_1 + 1.5y_2 + 0.5y_3 + {x_1}^2 + {x_2}^2 \\
\text{s.t.} \quad (x_1 - 2)^2 - x_2 \leq 0 \\
x_1 - 2y_1 \geq 0 \\
x_1 - x_2 - 4(1 - y_2) \leq 0 \\
x_1 - (1 - y_1) \geq 0 \\
x_2 - y_2 \geq 0 \\
x_1 + x_2 \geq 3y_3 \\
y_1 + y_2 + y_3 \geq 1 \\
0 \leq x_1 \leq 4, \quad 0 \leq x_2 \leq 4 \\
y_1, y_2, y_3 = 0, 1
\end{aligned}
$$

Implement the outer approximation, generalized Benders decomposition, extended cutting plane, extended supporting hyperplane algorithms to solve this problem using pyomo.

For OA and GBD, you can first solve the NLP subproblem with the y variables fixed as $y_1 = y_2 = y_3 = 1$ to generate cuts before solving the MILP.

For ECP and ESP, you can first generate cuts at starting point $y_1 = y_2 = y_3 = 1$ $x_1 = x_2 = x_3 = 0$.

This can help avoid unboundedness in the first iteration.

Hints: For GBD, you can use the cuts you derived from problem 2, which will make the implementation easy.

You can manually take derivatives.

The MILP master problem and the NLP subproblem can be solved with Gurobi.

Refer to the pyomo user manual to see how to get dual variables of a given constraint. Be careful about the signs of the dual variables you get from pyomo, making sure they are consistent with the way you derive the Lagrangian function.

4. A well-known stochastic programming problem is the farmer's problem introduced in the textbook

   Birge, J. R., & Louveaux, F. (2011). Introduction to stochastic programming. Springer Science & Business Media.

   Read section 1.1 of the textbook about the farmer's problem. Implement the farmer's problem in pyomo and solve it with Gurobi using the data from the textbook.

5. Solve the problem you implemented in problem 4 using Benders decomposition. Compare your solution with what you get from problem 4.