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.

Solution: Consider the case when the MILP master problem returns the same integer solution in iteration K and $K+1$. The bound generated from the NLP subproblem bound will be the same in iterations K and $K+1$. as the NLP subproblem is convex. This also implies that the same cuts will be generated at the end of the iterations K and $K+1$ which implies that the master problem bounds will no longer be improved. We first prove that the upper and lower bounds are the same at the end of the $K+1$ iteration. This would imply that the outer approximation has converged.

Consider the following LP,

$$
\begin{aligned}
v(\hat{y}) = \min_{x,\gamma} \quad & \gamma \\
& f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{pmatrix} x - x^k \\ \hat{y} - y^k \end{pmatrix} \leq \gamma \quad \forall k = 1, \dots, K-1 \\
& g_i(x^k, y^k) + \nabla g_i(x^k, y^k)^T \begin{pmatrix} x - x^k \\ \hat{y} - y^k \end{pmatrix} \leq 0 \quad \forall k = 1, \dots, K-1 \, \forall i \in 1, \dots, m
\end{aligned}
$$

$v(\hat{y})$ is the MILP master problem with y fixed at $\hat{y}$.

The Lagrangian function of the problem is defined below:

$$
\begin{aligned}
\mathcal{L}(x, \gamma, \lambda, \nu) = \gamma + & \sum_{k=1}^{K-1} \alpha_k \left(f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{pmatrix} x - x^k \\ \hat{y} - y^k \end{pmatrix} - \gamma \right) + \\
& \sum_{k=1}^{K-1} \lambda_{i,k} \left(g_i(x^k, y^k) + \nabla g_i(x^k, y^k)^T \begin{pmatrix} x - x^k \\ \hat{y} - y^k \end{pmatrix} \right)
\end{aligned}
$$

The KKT conditions are as follows:

$$\nabla_x \mathcal{L} = 0$$

$$\nabla_\gamma \mathcal{L} = 0$$

$$\alpha_k \left(f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{pmatrix} x - x^k \\ \hat{y} - y^k \end{pmatrix} - \gamma \right) = 0 \forall k = 1, \dots, K-1$$

$$\lambda_{i,k} \left(g_i(x^k, y^k) + \nabla g_i(x^k, y^k)^T \begin{pmatrix} x - x^k \\ \hat{y} - y^k \end{pmatrix} \right) = 0 \forall k = 1, \dots, K-1 \, \forall i \in 1, \dots, m$$

$$\lambda_{i,k}, \alpha_k \geq 0$$

From the above we can derive the following:

$$\nabla_x \mathcal{L} = \sum_{k=1}^{K-1} \alpha_k \nabla_x f(x^k, y^k)^T + \sum_{k=1}^{K-1} \sum_{i=1}^{m} \lambda_{i,k} \nabla_x g(x^k, y^k)^T = 0$$

$$\nabla_\gamma \mathcal{L} = 1 - \sum_{k=1}^{K-1} \nu_k = 0$$

We also know that

$$f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{pmatrix} x - x^k \\ \hat{y} - y^k \end{pmatrix} \leq \gamma \quad \forall k = 1, \ldots, K-1$$
$$g_i(x^k, y^k) + \nabla g_i(x^k, y^k)^T \begin{pmatrix} x - x^k \\ \hat{y} - y^k \end{pmatrix} \leq 0 \quad \forall k = 1, \ldots, K-1 \, \forall i \in 1, \ldots, m$$

Further x^k is the solution to the NLP solved fixing $y = y^k$. The NLP solved is as follows :

$$NLP : \min_x \quad f\left(x, y^k\right)$$
$$g_i\left(x, y^k\right) \leq 0 \quad \forall i = 1, \ldots, m$$

We assume that the NLP is always feasible. Then x_k is a KKT point to the above NLP. We can define the Lagrangian of the above NLP as follows

$$\mathcal{L}_{NLP} = f\left(x, y^k\right) + \sum_{i=1}^{m} \beta_i g_i\left(x, y^k\right)$$

We have the following conditions for $x = x_k, \beta = \beta^*$ being the optimum solution:

$$\nabla_x \mathcal{L}_{NLP} = 0 = \nabla_x f\left(x^k, y^k\right) + \sum_{i=1}^{m} \beta_i^* \nabla_x g_i\left(x^k, y^k\right)$$

$$\beta_i^* \geq 0$$

Let us consider a $k^* \in [1, K-1]$ where $\hat{y} = y_{k^*}$. If the NLP subproblem is feasible we have the following:

$$\nabla_x \mathcal{L}_{NLP} = 0 = \nabla_x f\left(x^{k^*}, y^{k^*}\right) + \sum_{i=1}^{m} \beta_i^* \nabla_x g_i\left(x^{k^*}, y^{k^*}\right)$$

$$\beta_i^* \geq 0$$

In the LP, if we set $\nu_k^* = 1, \nu_k = 0 \forall k = 1, 2 \ldots K-1, k \neq k^*, \lambda_{i,k^*} = \beta_i^* \forall i = 1..m, \lambda_{i,k} = 0 \forall k = 1, 2 \ldots K-1, k \neq k^*, \gamma = f(x^k, y^k)$, we satisfy the KKT conditions:

$$\nabla_x \mathcal{L} = 0$$

$$\nabla_\gamma \mathcal{L} = 0$$

Furthermore since the MINLP is convex, we have the following conditions also satisfied:

$$f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{pmatrix} x^{k^*} - x^k \\ y^{\hat{k}^*} - y^k \end{pmatrix} \leq f(x^{k^*}, y^{k^*}) \quad \forall k = 1, \ldots, K-1$$

$$g_i(x^k, y^k) + \nabla g_i(x^k, y^k)^T \begin{pmatrix} x^{k^*} - x^k \\ y^{\hat{k}^*} - y^k \end{pmatrix} \leq g_i(x^{k^*}, y^{k^*})^T \leq 0 \quad \forall k = 1, \ldots, K-1 \, \forall i \in 1, \ldots, m$$

Therefore $x = x^{k^*}$ is a KKT point of $v(y^{k^*})$. This implies that $x = x^{k^*}$ is the optimal solution of $v(y^{k^*})$ as $v(y^{k^*})$ is an LP.

From the above we can infer that at $y = y^K, x = x^k$ will be the optimal solution to the MILP master problem. Since it is also the solution to the NLP subproblem the bounds generated will be the same

If y^K is not the optimal solution then there exists a solution y' which is feasible and has a better objective than that at y^K. However this means that the we can better solution at the MILP Master problem which is a contradiction. Therefore when the MILP master problem returns the same integer solution in iteration K and $K+1$, we get the optimal solution for the integer variables.

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.

Solution:

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

Let μ_i^k be the optimal dual multiplier of the ith inequality constraint and λ_j^k be the optimal dual multiplier of the the constraint $y_j - y_j^k = 0$ and λ^k being the set of of all λ_j^k.

The stationarity condition is

$$\nabla_x f\left(x^k, y^k\right) + \sum_{i=1}^{m} \mu_i^k \nabla_x g_i\left(x^k, y^k\right) = 0$$

$$\nabla_y f\left(x^k, y^k\right) + \lambda^k + \sum_{i=1}^{m} \mu_i^k \nabla_y g_i\left(x^k, y^k\right) = 0$$

Consider the OA cuts added to the MILP master problem at iteration k

$$\begin{aligned} & f(x^k, y^k) + \nabla_x f(x^k, y^k)^T (x - x^k) + \nabla_y f(x^k, y^k)^T (y - y^k) \leq \gamma \\ & g_i(x^k, y^k) + \nabla_x g_i(x^k, y^k)^T (x - x^k) + \nabla_y g_i(x^k, y^k)^T (y - y^k) \leq 0 \end{aligned}$$

For the OA cuts, we multiply the first constraint by 1, and the constraint corresponding to g_i by μ_i^k

$$\begin{aligned} & f(x^k, y^k) + \nabla_x f(x^k, y^k)^T (x - x^k) + \nabla_y f(x^k, y^k)^T (y - y^k) \leq \gamma \\ & g_i(x^k, y^k) + \nabla_x g_i(x^k, y^k)^T (x - x^k) + \nabla_y g_i(x^k, y^k)^T (y - y^k) \leq 0 \end{aligned}$$

We obtain

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

The terms involving $(x-x^k)$ cancel out due to stationarity condition. The terms involving $(y-y^k)$ simplify to $-(\lambda^k)^T(y-y^k)$ due to stationarity condition. $\mu_i^k g_i(x^k,y^k) = 0$ due to complementarity condition.

We have

$$f(x^k,y^k) - (\lambda^k)^T(y-y^k) \leq \gamma$$

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.

Solution: The code is given in While implementing GBD we use the ipopt solver for the NLP to get the duals. While implementing the ESP algorithm, we don't include the the non-linear objective during the line search and add cuts for the non-linear objective at each iteration. We add the cuts for the objective from the MILP Master problem itself rather than line search. The optimal solution we get is $x_1 = 1, x_2 = 1, y_1 = 0, y_2 = 1, y_3 = 0$ and the objective being 3.5.

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.

Solution: The required optimization model is given in the book. The code is given in .. The optimal objective is -108390.0.

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

   Solution: The code is given in ... We get the same optimum objective from solving problem 4.