Lecture 19 Decomposition Algorithms for MINLP

Can Li

ChE 597: Computational Optimization
Purdue University

Decomposition Algorithm

- Decompose a hard optimization problem into easily solvable problems.

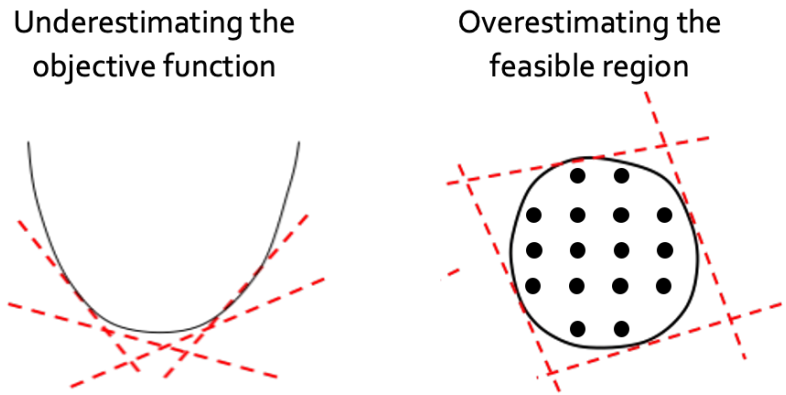
$$\begin{aligned} \text{(MINLP)} \quad \min_{x,y} \quad & f(x,y) \\ \text{s.t.} \quad & g(x,y) \leq 0 \\ & x \in \mathbb{R}^{n^x}, y \in \{0,1\}^{n^y} \end{aligned}$$

f, g are convex functions (convex MINLP).

- Decomposition algorithms for MINLP decompose the (MINLP) into MILP and NLP.
  - Outer Approximation
  - Generalized Benders Decomposition
  - Extended Cutting Planes

Outer Approximation

- Proposed by Duran and Grossmann, 1986.
- Solves convex MINLPs to global optimality



- Key idea: outer approximate the convex nonlinear functions by first-order Taylor series expansion. Lower bound provided by the resulting MILP.
- Where to outer approximate? Determined by solving the NLP (upper bound).

MILP master problem

The MILP master problem at iteration K is defined as,

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

y^k is the optical solution of the MILP.

where k denotes the number of iterations of the OA algorithms. Note that we only need to outer approximate the nonlinear constraints, the linear constraints are kept.

How to initialize the MILP?

- In the first iteration of the OA algorithm, we do not know where to outer approximate the nonlinear constraints since no NLP have been solved.
- Without adding any OA cuts, the MILP could be unbounded.
- One heuristic is to solve the relaxed NLP problem

$$\begin{aligned} \text{(rNLP)} \quad & \min_{x,y} \quad f(x,y) \\ & \text{s.t.} \quad g(x,y) \leq 0 \\ & \qquad x \in \mathbb{R}^{n^x}, y \in [0,1]^{n^y} \end{aligned}$$

Use the solution of (rNLP) (x^0, y^0) to generate an OA cuts.

NLP subproblem

Fix the binary variable y NLP restriction for a fixed y^K where y^K is the optimal solution to the MILP master problem at iteration K:

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

If the NLP is infeasible, then solve the following problem by adding slack variable u to each constraint.

$$\begin{aligned} \min_{u, x} \quad & u \\ & g_i\left(x, y^K\right) \leq u \quad \forall i = 1, \ldots, m \\ & u \in \mathbb{R}_+ \end{aligned}$$

The optimal solution to the NLP is denoted as x^k. OA cuts corresponding to x^k, y^k will be added to the MILP master problem in the next iteration.

OA Algorithm

$K \leftarrow 1$, define an initial MILP relaxation using x^0, y^0, $UB \leftarrow +\infty$, $\text{LB} \leftarrow -\infty$;
while $UB \neq LB$ **do**
    Solve the current MILP relaxation (obtaining y^K) and update LB;
    Solve the current NLP restriction for y^K and obtain x^K;
    if *NLP restriction for* y^K *infeasible* **then**
        Solve the infeasibility subproblem for y^K and obtain x^K;
    else
        if $f(x^K, y^K) < UB$ **then**
            $UB \leftarrow f(x^K, y^K)$, $(x^*, y^*) \leftarrow (x^K, y^K)$;
        end
    end
    Generate linearization cuts with x^K, y^K, update MILP relaxation;
    $K \leftarrow K + 1$;
end
return (x^*, y^*);

Convergence of OA

- OA converges in a finite number of iterations.
- The key to the convergence proof is "if the MILP master problem returns the same integer solution in iteration K and $K+1$, then (x^K, y^K) must be optimal". Because if $y^K = y^{K+1}$, the same cuts will be generated from the NLP subproblem and the master problem bounds will no longer be improved. If y^K is not the optimal solution, the algorithm is stuck at this suboptimal solution.

Convergence of OA

Proof hint: show the following claim is true.
Consider the following LP,

$$v(\hat{y}) = \min_{x} \quad \gamma$$

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

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

The claim is that for $\hat{y} = y^k$, $k = 1, \ldots, K-1$, $v(\hat{y}) = f(x^k, y^k)$.
It suffices to show that x^k is a KKT point.
You will complete the proof in your homework.

Generalized Benders Decomposition

- Proposed by Geoffrion, 1972. The original paper is "unreadable". Let's show an easy derivation.
- Replace the OA cuts with an aggregation of OA cuts such that the MILP master problem only has the binary variables y.
- To this end, we need to "project" the MILP master problem from the (x, y) space to the y space.

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

$$f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} \leq \gamma$$

$$g_i(x^k, y^k) + \nabla g_i(x^k, y^k)^T \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} \leq 0$$

m+1 cuts

which can be written as

$$f(x^k, y^k) + \nabla_x f(x^k, y^k)^T (x - x^k) + \nabla_y f(x^k, y^k)^T (y - y^k) \leq \gamma \quad \times 1$$

$$g_i(x^k, y^k) + \nabla_x g_i(x^k, y^k)^T (x - x^k) + \nabla_y g_i(x^k, y^k)^T (y - y^k) \leq 0 \quad \times \mu_i$$

We would like to remove the terms related to x.

When solving the NLP subproblem

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

Let μ_i^k be the optimal dual multiplier of the ith constraint.
The stationarity condition is

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

Therefore, we also have

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

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

$$f(x^k, y^k) + \nabla_x f(x^k, y^k)^T (x - x^k) + \nabla_y f(x^k, y^k)^T (y - y^k) \leq \gamma$$
$$g_i(x^k, y^k) + \nabla_x g_i(x^k, y^k)^T (x - x^k) + \nabla_y g_i(x^k, y^k)^T (y - y^k) \leq 0$$

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. $\mu_i^k g_i(x^k, y^k) = 0$ due to complementarity condition. We have

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

This is called the Generalized Benders cut.

GBD

- The difference between GBD and OA is the form of the cut. All the other steps are the same.
- Note that we only add one cut per iteration in GBD. The GBD cut is weaker than the OA cuts since it is an aggregation of the OA cuts.
- As a result, GBD usually takes more iterations to converge.

When the NLP subproblem is infeasible, we will also solve

$$\begin{aligned} \min \quad & u \\ & g_i\left(x, y^k\right) \leq u \quad \forall i = 1, \ldots, m \\ & u \in \mathbb{R}_+ \end{aligned}$$

The cuts derived from this problem is called a "feasibility cut". You will derive the feasibility cut in your homework.

Extended Cutting Plane Algorithm

After solving the MILP master problem

$$\min \quad \gamma$$

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

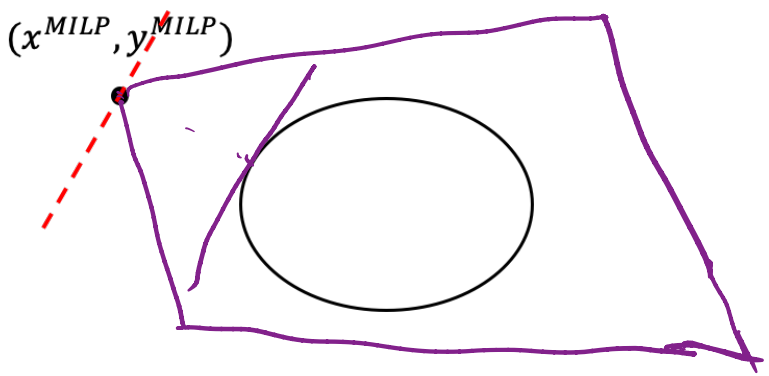
$$g_i(x^k, y^k) + \nabla g_i(x^k, y^k)^T \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} \leq 0 \quad \forall k = 1, \ldots, K-1 \, \forall i \in 1,$$

$$y \in \{0, 1\}^{n_y}$$

Use the solution of the MILP master problem (x^K, y^K) to add cuts.

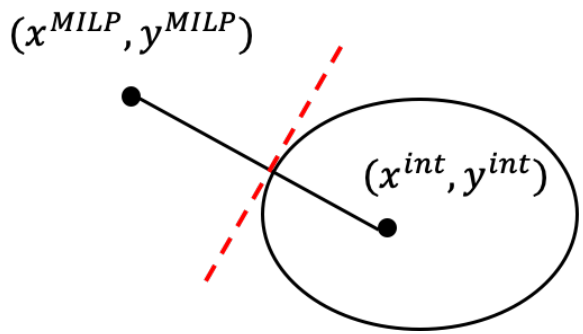
Extended Cutting Plane (ECP)

- The difference between ECP and OA is the way the cuts are generated.
- In OA, y^k is from the solution to the MILP master problem, x^k is from solving the NLP subproblem.
- In ECP, both y^k and x^k is from solving the MILP master problem.
- In ECP method, the x^k, y^k usually do not satisfy the nonlinear constraint $g_i(x, y) \leq 0$. Therefore, these cuts are not "supporting hyperplanes" for the convex feasible region.
- ECP usually takes larger number of iterations than OA.

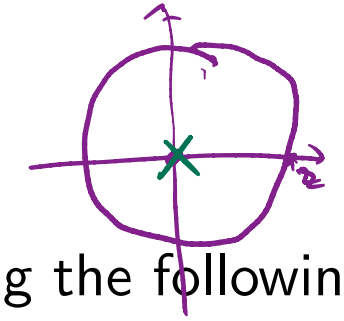


Extended Supporting Hyperplane (ESP)

- To address this issue, Kronqvist et al, 2016 extended ECP to ESP.
- The idea is to do a line search between the MILP solution (x^{MILP}, y^{MILP}) and an interior point (x^{int}, y^{int}) to generate a supporting hyperplane.



Extended Supporting Hyperplane



The interior point (x^{int}, y^{int}) is obtained by solving the following NLP.

$$\begin{aligned} \min_{x,y,u} \quad & u \\ \text{s.t.} \quad & g_i(x,y) \le u \quad \forall i = 1,\dots,m \\ & x \in \mathbb{R}^{n^x}, y \in [0,1]^{n^y}, u \in \mathbb{R}^1 \end{aligned}$$

$x^2 - 4 \le 0$
$x \in X$

$\min_{u \in \mathbb{R}} u$
$x^2 - 4 \le u$
$x \in X$

$u = -2$

Note that here u is unconstrained. It will give a strictly feasible point of the NLP, denoted as (x^{int}, y^{int}).

Extended Supporting Hyperplane

Define the maximum of all the constraints as

$$F(x,y) = \max_{i=1,\ldots,m} \{g_i(x,y)\}$$

The new points $\left(x^k, y^k\right)$ are then determined by,

$$x^k = \lambda^k x^{\text{int}} + \left(1-\lambda^k\right) x^{MILP}$$
$$y^k = \lambda^k y^{\text{int}} + \left(1-\lambda^k\right) y^{MILP}$$

where x^{MILP} and y^{MILP} are the solution of the master MILP in step 2, and λ^k is chosen such that $F(x,y) = 0$, i.e. at the boundary of the feasible region. The points $\left(x^k, y^k\right)$ are then located at the boundary of the feasible region, so that linearizing the active constraints results in supporting hyperplanes. Numerical experience has shown that these linearization improve significantly the convergence of the ECP method.

LP/NLP-based branch and bound (QG)

- Proposed by Quesada and Grossmann, 1992. Typically referred to as the QG algorithm.
- Motivation: The OA algorithm needs to solve MILP for multiple times. Each time the MILP is solved, a branch and bound algorithm needs to start from scratch.
- The QG algorithm can be seen as a "single-tree" OA algorithm where we only have one single branch and bound tree.

LP/NLP branch and bound (QG)

During the single tree branch and bound algorithm, the NLP is solved whenever an integer feasible solution is found at a given node. OA cuts are added to all active nodes after the NLP is solved.

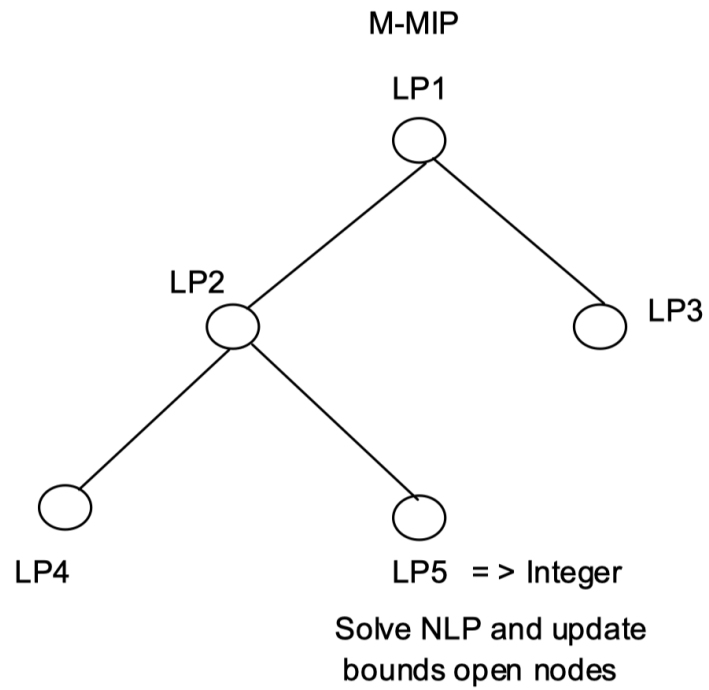
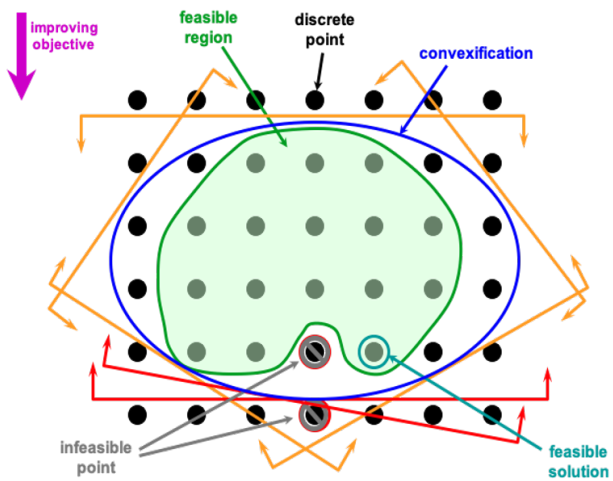


Figure: ref: Quesada and Grossmann, 1992

Extension of OA to nonconvex MINLP

- Proposed by Kesavan et al, 2004.
- Convexify the nonconvex functions. OA cuts added to the convex nonlinear relaxation. The convexification provides a **lower bound**.
- The **upper bound** is provided by fixing the binary variable and solve the nonconvex NLP to global optimality, e.g., with a global solver.
- Must add "no-good" cuts/ integer cuts to cut off the integer point, otherwise, the algorithm might cycle. This is the key difference between OA for convex MINLP and OA for nonconvex MINLP.

$$\sum_{i \in B} y_i - \sum_{i \in N} y_i \leq |B| - 1$$



Solvers based on decomposition algorithms

- DICOPT (the first MINLP solver, OA)
- alphaECP (ECP)
- SHOT (ESP)
- MindtPy (OA, QG, pyomo-based)
- For benchmark of convex MINLP, check the paper Kronqvist, J., Bernal, D. E., Lundell, A., & Grossmann, I. E. (2019). A review and comparison of solvers for convex MINLP. Optimization and Engineering, 20, 397-455.

References

- Grossmann, I. E. (2021). Advanced optimization for process systems engineering. Cambridge University Press.
- Duran, M. A., & Grossmann, I. E. (1986). An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Mathematical programming, 36, 307-339.
- Geoffrion, A. M. (1972). Generalized Benders Decomposition. Journal of optimization theory and applications, 10, 237-260.
- Kronqvist, J., Lundell, A., & Westerlund, T. (2016). The extended supporting hyperplane algorithm for convex mixed-integer nonlinear programming. Journal of Global Optimization, 64, 249-272.
- Quesada, I., & Grossmann, I. E. (1992). An LP/NLP based branch and bound algorithm for convex MINLP optimization problems. Computers & chemical engineering, 16(10-11), 937-947.
- Kesavan, P., Allgor, R. J., Gatzke, E. P., & Barton, P. I. (2004). Outer approximation algorithms for separable nonconvex mixed-integer nonlinear programs. Mathematical Programming, 100, 517-535.