Lecture 18 Branch and Reduce

Can Li

ChE 597: Computational Optimization
Purdue University

Spatial branch-and-bound

Need to perform spatial branching on the continuous variable to obtain global optimality

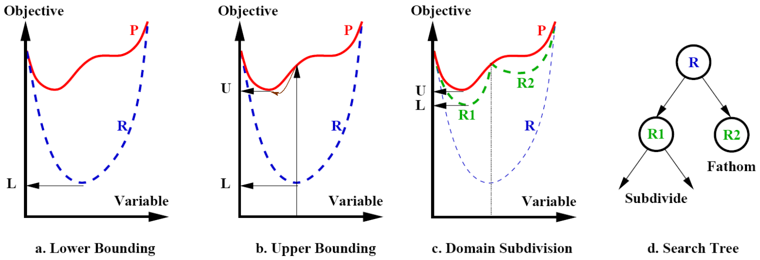
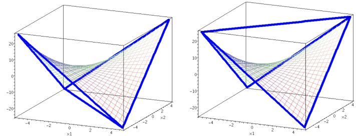


Figure: ref: Tawarmalani & Sahinidis

Branch and Reduce

- First proposed by Ryoo and Sahinidis (1996) in their seminal paper: A Branch-and-Reduce Approach to Global Optimization.
- Used as the basis of the global solver BARON and has been adopted by most global solvers since then.
- Compared with branch and bound for MILP. Branch and reduce for nonconvex NLP has two major differences: (1) it branches on the continuous variables (2) it performs **range reduction (bound tightening)** to reduce the range of the variable bounds.
- **range reduction (bound tightening)** narrow the search space for variables by identifying tighter bounds on.
- For example, the McCormick envelopes are tighter as the variable ranges become smaller.



Branch and reduce for nonconvex MINLP

- Node selection rule: "best bound first", i.e., select the node with the smallest lower bound.
- Branching rule: usually branch on the binary variables first before branching on the continuous variables. Branching rules for binary variables are similar to MILP. Branching rules for continuous variables
  - Typically, select variable with largest underestimating gap. Occasionally, select variable corresponding to largest range.
- Branching point selection: typically at the midpoint of the interval.
- Convex relaxations: modern solvers like BARON and Gurobi uses mostly linear relaxations. Cutting planes such as Gomory cuts and SDP cuts may also be added.

Branch-and-Reduce Algorithm

Initialization Step
Set $k = 0$.
Set the upper bound $U^{(k)} = +\infty$.
Put range $R_1 = R$ in the list ACTIVE of active subproblems with a corresponding lower bound of $L_1 = -\infty$.
Go to the main step.
Main Step (at iteration k)
Step 1: Check Termination Criteria
Set the lower bound $L^{(k)} = \min_{i;R_i \in \text{ACTIVE}}\{L_i\}$.
Set $\text{ACTIVE} = \text{ACTIVE} \setminus \{R_j\}$ for all R_j with $L_j \geq U^{(k)}$. (prune by bound)
If $ACTIVE = \emptyset$,

Stop. The current best solution is optimal.

Otherwise,

Set $k \leftarrow k+1, U^{(k)} \leftarrow U^{(k-1)}$ and $L^{(k)} \leftarrow L^{(k-1)}$.

Go to Step 2.

Step 2: Subproblem Selection

Select R_k from ACTIVE according to a node selection rule.
Set ACTIVE = ACTIVE$\setminus\{R_k\}$.
Go to Step 3.

Step 3: Pre-processing

Tighten variable bounds for R_k, using feasibility-based range reduction.
Go to Step 4.

Step 4: Bounding

Solve R_i, or bound its solution from below. Let L_i be this lower bound ($L_i = +\infty$ if R_i is infeasible.)
If the solution, x^i, found for R_i is feasible to P and $f\left(x^i\right) < U^{(k)}$,
    Update $U^{(k)} \leftarrow f\left(x^i\right)$.
    Make x^i the current best solution: $x^* \leftarrow x^i$.
If $L_i \geq U^{(k)}$,
    Fathom by bound. Go to Step 1 .
Otherwise,
    Go to Step 5 .

Step 5: Optional Upper Bounding

Apply local search heuristics to find a better feasible solution, x^h, for P. If successful,

Update $U^{(k)} \leftarrow f\left(x^h\right)$.

Make x^h the current best solution: $x^* \leftarrow x^h$.

Go to Step 6.

Step 6: Post-processing

Strengthen the bounds of variables using optimality-based and feasibility-based range reduction.

If the range reduction was successful in reducing the range of at least one variable of R_i by at least a prespecified amount $\delta > 0$, then:

Reconstruct R_i, using the new variable bounds.

Go to Step 4.

Otherwise,

Go to Step 7.

Step 7: Partitioning

Apply a branching rule to R_i; obtain a set of new subproblems $R_{i_1}, R_{i_2}, \ldots, R_{i_q}$, and place them on ACTIVE. Go to Step 1.

Convergence of spatial branch and bound

- Not as trivial as branch and bound for MILP since the branching is performed on continuous variables and thus may not be finite.
- A detailed proof of spatial branch and bound can be found in Horst and Tuy's global optimization book.
- In summary, to show finite converge we need
  - **Consistent partitioning** Any open partition can be further refined. As refinement progresses, the lower bound converges to the nonconvex problem value
  - **Bound improving node selection rule** Every finite number of steps, a node with the least lower bound is selected
  - **Exhaustiveness (optional)** The limit of the diameter of a sequence of nested partitions is zero. Not necessary for convergence but most branch-and-bound algorithms satisfy it

Range reduction (bound tightening)

- The smaller the domain, the faster branch-and-bound converges.
  - Tighter lower bounds
  - Fewer nodes
- Range reduction techniques.
  - Based on marginals
  - FBBT: interval arithmetic operations on the problem constraints to tighten bounds.
  - OBBT: solving an optimization problem to tighten bounds.
  - Based on probing

Marginals-based range reduction

Suppose the following convex relaxation solved at the a given node.

$$\min c^T x \quad \text{s.t.} \quad g_i(x) \leq b_i, \forall i \in 1, \ldots, m$$

Suppose a constraint $g_j(x) \leq b_j$ is active at the optimal solution of the relaxation and the corresponding dual multiplier $\lambda_j^* > 0$. Then

$$g_j(x) \geq b_j - (U - L)/\lambda_j^*$$

is valid for all solutions with better (lower) objective function value than U.

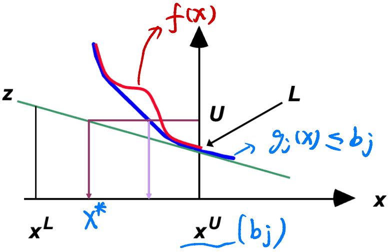


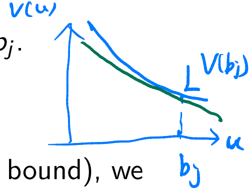
Figure: ref: Tawarmalani & Sahinidis

Marginals-based range reduction

Proof. The idea is to analyze how the optimal objective of the convex relaxation change with respect to b_j. We define a value function as

$$v(u) := \min c^T x \quad \text{s.t.} \quad g_i(x) \leq b_i, \forall i \in 1, \ldots, m, i \neq j, g_j(x) \leq u$$

We claim that $-\lambda_j^*$ is a subgradient of $v(u)$ at $u = b_j$.



$$v(u) \geq v(b^j) - \lambda_j^*(u - b_j)$$

Since $v(b^j)$ is the original convex relaxation (a lower bound), we can denote it as L.
If $L - \lambda_j^*(u - b_j) \geq U$, i.e., $u \leq b_j - (U - L)/\lambda_j^*$, we have $v(u) \geq U$. However, we are only interested in the domain where we can potentially find solutions better than U. Therefore, adding the constraint $g_j(x) \geq b_j - (U - L)/\lambda_j^*$ does not eliminate any solution with bound less than U.

Claim $-\lambda_j^*$ is a subgradient of $v(u)$ at $u = b_j$.
Proof: Consider the Lagrange dual function of the convex relaxation parameterized by u

$$D(\lambda, u) = \min_x c^T x + \sum_{i=1, i\neq j}^{m} \lambda_i(g_i(x) - b_i) + \lambda_j(g_j(x) - u)$$

Due to strong duality, we have $v(u) = \max_{\lambda \geq 0} D(\lambda, u)$. Denote the optimal primal and dual solution at $u = b_j$ as x^*, λ^*, i.e.,

$$x^* = \arg\min_x c^T x + \sum_{i=1}^{m} \lambda_i^* g_i(x)$$

With these definitions, we can observe that

$$\begin{aligned} v(u) = \max_{\lambda \geq 0} D(\lambda, u) \geq D(\lambda^*, u) = c^T x^* + & \sum_{i=1, i\neq j}^{m} \lambda_i^*(g_i(x^*) - b_i) \\ & + \lambda_j^*(g_j(x^*) - u) \\ & = v(b^j) - \lambda_j^*(u - b_j) \end{aligned}$$

Feasibility Based Bound Tightening (FBBT)

FBBT works by inferring tighter bounds on a variable x_i as a result of a changed bound on one or more other variables x_j that depend, directly or indirectly, on x_i.

For example, if $x_j = x_i^3$ and $x_i \in [l_i, u_i]$, then the bound interval of x_j can be tightened to $[l_j, u_j] \cap \left[l_i^3, u_i^3\right]$.

Vice versa, a tightened bound l_j' on x_j implies a possibly tighter bound on x_i, namely, $l_i' = \sqrt[3]{l_j'}$.

FBBT Example with Multiplication

Consider $x_k = x_i x_j$, with $(1,1,0) \le (x_i, x_j, x_k) \le (5,5,2)$. Lower bounds $l_i = l_j = 1$ imply a tighter lower bound $l_k = l_i l_j = 1 > 0$, while the upper bound $u_k = 2$ implies that $x_i \le \frac{u_k}{l_j}$ and $x_j \le \frac{u_k}{l_i}$, hence $u_i' = u_j' = 2 < 5$.

Note that if McCormick relaxation is used. The tightening of bounds of x_i, x_j will yield a tighter relaxation.

FBBT for Affine Functions

Suppose x_k is an auxiliary variable defined as $x_k = a_0 + \sum_{j=1}^{n} a_j x_j$, with $k > n$. For $J^+ = \{j = 1, \ldots, n : a_j > 0\}$ and $J^- = \{j = 1, \ldots, n : a_j < 0\}$, valid bounds on x_k are:

$a_j = -1, \quad u_j = 1, \quad l_j = -1$

$$a_0 + \sum_{j \in J^-} a_j u_j + \sum_{j \in J^+} a_j l_j \le x_k \le a_0 + \sum_{j \in J^-} a_j l_j + \sum_{j \in J^+} a_j u_j.$$

Explicit bounds $[l_k, u_k]$ on x_k imply new (possibly tighter) bounds on x_j:

$l_k \le a_0 + \sum_{j=1}^{n} a_j x_j \le u_k$

$$\forall j : a_j > 0, \quad l'_j = \frac{1}{a_j}\Big(l_k - (a_0 + \sum_{i \in J^+ \setminus \{j\}} a_i u_i + \sum_{i \in J^-} a_i l_i)\Big),$$

$$u'_j = \frac{1}{a_j}\Big(u_k - (a_0 + \sum_{i \in J^+ \setminus \{j\}} a_i l_i + \sum_{i \in J^-} a_i u_i)\Big),$$

$$\forall j : a_j < 0, \quad l'_j = \frac{1}{a_j}\Big(u_k - (a_0 + \sum_{i \in J^+} a_i l_i + \sum_{i \in J^- \setminus \{j\}} a_i u_i)\Big),$$

$$u'_j = \frac{1}{a_j}\Big(l_k - (a_0 + \sum_{i \in J^+} a_i u_i + \sum_{i \in J^- \setminus \{j\}} a_i l_i)\Big).$$

$\frac{a_0}{a_j} + \frac{1}{a_j}\Big[\sum_{\substack{i=1 \\ i \ne j}}^{n} a_i x_i\Big] + x_j \le \frac{u_k}{a_j}$

Convergence of FBBT

- FBBT algorithms allow for fast implementation and are commonly used in problems even of very large size.
- Once the bounds of variables are updated by FBBT in the previous slide, we can use the updated bounds l'_j, u'_j to perform FBBT again.
- This procedure does not terminate unless tolerances or iteration limits are imposed, and it does not achieve its fixed point in finite time.

Optimality Based Bound Tightening (OBBT)

For each variable $x_i, i = 1, 2, \ldots, n$, updated lower and upper bounds can be computed by solving the following optimization problems:

$$l_i' = \min\left\{x_i : x \in S, c^T x \leq U\right\}; \quad u_i' = \max\left\{x_i : x \in S, c^T x \leq U\right\}.$$

where S denotes the feasible region of the original problem. U denotes the best upper bound found so far.

This process involves solving $2n$ optimization problems, which can be as challenging as the original problem due to nonconvexity.

Optimality Based Bound Tightening (OBBT)

A practical approach uses convex relaxation to define a more manageable feasible set $\mathcal{F}(l, u)$. The feasible set $\mathcal{F}(l, u)$ for the convex relaxation includes:

$$
\begin{array}{ll}
a^k x_k + B^k x \geq d^k & k = n+1, n+2, \ldots, n+q \\
l_i \leq x_i \leq u_i & i = 1, 2, \ldots, n+q \\
x \in X &
\end{array}
$$

Using this relaxed set, we compute:

$$
l_i' = \min \left\{ x_i : x \in \mathcal{F}(l, u), c^T x \leq U \right\}
$$

$$
u_i' = \max \left\{ x_i : x \in \mathcal{F}(l, u), c^T x \leq U \right\}.
$$

Optimality Based Bound Tightening (OBBT)

- OBBT can be more effective than FBBT in tightening variable bounds. It considers all the contratins simultaneously rather than rely on one constraint.
- It requires solving $2n$ LPs. Much more expensive than interval arithmetic (FBBT).
- OBBT's use is limited to the root node or to the nodes of small depth.

Probing in MILP

- Probing is a technique originally proposed to solve MILPs and has been extended to solving global optimization problems.
- In MILP, probing is a process of fixing binary variables temporarily to different values (0 or 1) and analyzing the consequences of these assignments on the feasibility and optimality of the solution.
- This technique is especially useful in preprocessing and during the branch-and-bound process to reduce the solution space.

MILP Problem and Probing Technique

MILP Problem:
Maximize $Z = 3x_1 + 5x_2$
Subject to:
$2x_1 + 3x_2 \leq 10$
$4x_1 + x_2 \geq 8$
$x_1, x_2 \in \{0, 1\}$

Probing Technique:
Step 1: Fix x_1 to 0. We have $3x_2 \leq 10$ and $x_2 \geq 8$. The problem becomes infeasible, indicating x_1 must be 1.
Step 2: Fix x_1 to 1. We have $3x_2 \leq 8$ and $x_2 \geq 4$. This leads to $x_2 = 1$ being the only feasible solution.
Through probing, $x_1 = 1$ and $x_2 = 1$ is determined, narrowing down the search space.

Probing in global optimization

- In global optimization, probing is performed on continuous variables.
- Consider adjusting the upper bound of a variable x_i to a fictitious value $u_i' < u_i$, irrespective of u_i' being initially valid, and then applying FBBT. If this results in infeasibility or the lower bound l_{n+q} on x_{n+q} exceeding a cutoff value, it indicates no optimal solution within $[l_i, u_i']$, adjusting x_i bounds to $[u_i', u_i]$. The same logic applies when imposing a fictitious lower bound l_i' and assessing feasibility within $[l_i', u_i]$. This iterative procedure across all variables can significantly narrow down the search space, albeit at a high computational cost.

Global optimization solvers

- BARON
- ANTIGONE
- SCIP
- OCTERACT
- SHOT
- Check Hans Mittelmann's MINLP benchmark for a comparison of global solvers. `https://plato.asu.edu/bench.html`

References

- Tawarmalani, M., & Sahinidis, N. V. (2013). Convexification and global optimization in continuous and mixed-integer nonlinear programming: theory, algorithms, software, and applications (Vol. 65). Springer Science & Business Media.
- Horst, R., & Tuy, H. (2013). Global optimization: Deterministic approaches. Springer Science & Business Media.
- Belotti, P., Kirches, C., Leyffer, S., Linderoth, J., Luedtke, J., & Mahajan, A. (2013). Mixed-integer nonlinear optimization. Acta Numerica, 22, 1-131.