Lecture 22 Lagrangian Relaxation and Decomposition

Can Li

ChE 597: Computational Optimization
Purdue University

Problems with complicating constraints

$$
\begin{aligned}
\min \quad & c_1^T x_1 + c_2^T x_2 + \cdots + c_t^T x_t \\
\text{s.t.} \quad & D_1 x_1 + D_2 x_2 + \cdots + D_t x_t = b_0 \\
& F_i x_i = b_i, \qquad i = 1, 2, \ldots, t \\
& x_1, x_2, \ldots, x_t \geq 0
\end{aligned}
$$

- In the previous lecture, we showed that this problem can be solved by Dantzig Wolfe decomposition. In this lecture, we will describe an algorithm that can be seen as the dual of Dantzig Wolfe decomposition.
- The idea of Lagrangian relaxation is to relax the complicating constraints.

Motivating application: two-stage stochastic programming

$$\min z = c^T x + \sum_{\omega \in \Omega} \tau_\omega \left[q_\omega^T y_\omega \right]$$

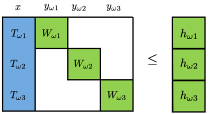
$$\text{s. t.} \quad Ax = b, x \geq 0$$

$$T_\omega x + W_\omega y_\omega = h_\omega, \quad \forall \omega \in \Omega$$

$$y_\omega \geq 0, \quad \forall \omega \in \Omega$$

- Let's duplicate x for each scenario. $x_\omega \in \Omega$.
- We need to add "Nonanticipativity constraints" (NACs)

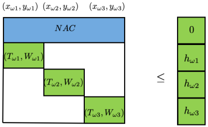
$$x_{\omega 1} = x_{\omega 2}, x_{\omega 1} = x_{\omega 3}, \cdots, x_{\omega 1} = x_{\omega |\Omega|}$$



TSSP formulation with NACs

$$\min z = \sum_{\omega \in \Omega} \tau_\omega \left[c^T x_\omega + q_\omega^T y_\omega \right]$$

$$\begin{aligned}
\text{s. t.} \quad & Ax_{\omega} = b, x_\omega \geq 0 \quad \forall \omega \in \Omega \\
& x_{\omega 1} = x_{\omega 2}, x_{\omega 1} = x_{\omega 3}, \cdots, x_{\omega 1} = x_{\omega |\Omega|} \\
& T_\omega x_{\omega} + W_\omega y_\omega = h_\omega, \quad \forall \omega \in \Omega \\
& y_\omega \geq 0, \quad \forall \omega \in \Omega
\end{aligned}$$



- Convert the TSSP from a problem with "complicating variables" to a problem with "complicating constraints".
- Lagrangian relaxation dualizes the NACs

Generalized Assignment Problem

Assign job j to machine i with capacity b_i. The cost coefficient is c_{ij}.

$$\min_{x \in \{0,1\}^n} \sum_i \sum_j c_{ij} x_{ij}$$

$$\text{s.t.} \quad \sum_j a_{ij} x_{ij} \leq b_i \quad \forall i$$

$$\sum_i x_{ij} = 1 \quad \forall j \qquad \times \lambda_j$$

a_{ij}: resource consumed by job j if assigned to machine i.

Dualize the assignment constraints, the problem becomes

$$\min_{x \in \{0,1\}^n} \sum_i \sum_j c_{ij} x_{ij} + \sum_j \lambda_j \left(1 - \sum_i x_{ij}\right)$$

$$\text{s.t.} \quad \sum_j a_{ij} x_{ij} \leq b_i \quad \forall i$$

It can be solved by any knapsack algorithm like dynamic programming for each machine i.

$$
\begin{aligned}
\min_{x \in \{0,1\}^n} \sum_i \sum_j (c_{ij} - \lambda_j)\, x_{ij} + \sum_j \lambda_j \\
\text{s.t.} \quad \sum_j a_{ij} x_{ij} \le b_i \quad \forall i
\end{aligned}
$$

Rearrange by machine i

$$
\begin{aligned}
Z^i = \min \quad & \sum_j (c_{ij} - \lambda_j)\, x_{ij} \\
\text{s.t.} \quad & \sum_j a_{ij} x_{ij} \le b_i \\
& x_{ij} = 0, 1
\end{aligned}
$$

each machine i involved a separate knapsack problem.

Lagrangian relaxation

We will first develop the theory before we show the algorithm.
Without loss of generality, consider the following MILP

$$
\begin{aligned}
v(P) := \min \quad & c^T x \\
\text{s.t.} \quad & Ax \leq b \\
& Ex \leq d \\
& x \in X \\
& X = \left\{ x_j \in \{0,1\},\ \forall j = 1 \dots p,\ x_j \in \mathbb{R}^+,\ \forall j = p+1, \dots n \right\}
\end{aligned}
$$

- X represents a mixed-integer set.
- $Ax \leq b$ are regarded as the "complicating constraints" NACs, assignment constraints
- $Ex \leq d$ are "easy" constraints, e.g., constraints with decomposable structure.

Lagrangian relaxation

- Idea: relax the complicating constraints through dualizing.

$$\begin{array}{lll} D(\lambda) := & \min_x & c^T x + \lambda^T (Ax - b) \\ \text{s.t.} & & Ex \leq d \\ & & x \in X \end{array}$$

$\lambda = 0 \Leftrightarrow$ removing the constraints $Ax \leq b$.

where $\lambda \geq 0$ non-negative Lagrange multipliers.

- Recall that for any $\lambda \geq 0$, we have $D(\lambda) \leq v(P)$, i.e., the Lagrangian relaxation provides a lower bound for the original problem.

The Lagrangian dual is defined as

$$v(LR) = \max_{\lambda \geq 0} D(\lambda)$$

For convex problems, strong duality holds, we have $v(LR) = v(P)$.

Question: how about nonconvex problems.

Answer: $v(LR) \leq v(P)$ in general.

Primal characterization of Lagrangian relaxation

- Motivation: to have a primal characterization of how tight the Lagrangian relaxation is. How does it compare with the LP relaxation for MILPs?

Theorem (primal characterization of Lagrangian relaxation)

$v(P) = \min_x c^T x$ s.t. $\mathrm{conv}\{x \mid Ax \le b, Ex \le d, x \in X\}$

$$v(LR) = \min_x c^T x \quad \text{s.t. } Ax \le b, x \in \mathrm{conv}\big(\{x | Ex \le d, x \in X\}\big)$$

$v(P) = \min_x c^T x$ s.t. $Ax \le b, Ex \le d, x \in X$.

In words, the Lagrangian relaxation is equal to solving the problem constrained by the convex hull of the constraints that are not dualized intersected with constraints that are dualized.

- Interpretation: LR provides the same bound as a partial convexification of the feasible region.
- This is true in general for any nonconvex set X
- We will show a proof for MILP-representable constraints.

Meyer's Theorem (The Fundamental Theorem of Integer Programming)

We will use Meyer's Theorem in our proof.
Theorem Given rational matrix E and a rational vector d, let $S := \{x : Ex \leq d, x \in X\}$ where $X = \{x_j \in \{0,1\}, \forall j = 1 \ldots p,\ x_j \in \mathbb{R}^+, \forall j = p+1, \ldots n\}$

- There exist rational matrices H and a rational vector h such that $\mathrm{conv}(S) = \{x : Hx \leq h\}$.

In words, the convex hull of an MILP-representable set with rational entries is a polyhedral set.

Proof. Define

$$v(\tilde{P}) = \min_x \left\{ c^T x \mid Ax \leq b, x \in \mathrm{conv}(S) \right\} = \min_x \left\{ c^T x \mid Ax \leq b, Hx \leq h \right\}$$

Take the dual of the LP (assume the feasible region is bounded and strong duality holds)

$$v(\tilde{P}) = \max_{\lambda,\mu \geq 0} \left\{ b^T \lambda + h^T \mu \mid A^T \lambda + H^T \mu \geq c \right\}$$

$$= \max_{\lambda \geq 0} \left\{ b^T \lambda + \max_{\mu \geq 0} \left\{ h^T \mu : H^T \mu \geq c - A^T \lambda \right\} \right\}$$

Take the dual of the inner problem

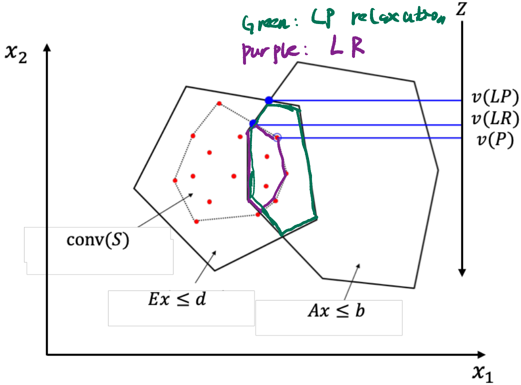
$$= \max_{\lambda \geq 0} \left\{ b^T \lambda + \min_{x \geq 0} \left(c^T - \lambda^T A \right) x : Hx \leq h \right\}$$

$$= \max_{\lambda \geq 0} \left\{ \min_{x \in X} \left\{ c^T x - \lambda^T (Ax - b) : Ex \leq d \right\} \right\}$$

$$= v(LR)$$

Lagrangian Relaxation v.s. LP relaxation

Corollary $v(LP) \le v(LR)$
Proof It is easy to check that the set $\{x \mid Ax \le b, x \in \text{conv}(S)\}$ is contained in the LP relaxation. $\{x \mid Ex \le d,\ x \in X\}$



Finding the Lagrangian dual

- How to find the optimal Lagrangian multipliers?
- For some problems, it is "easy". The Lagrangian function has a closed-form solution.

For example, recall that for QCQPs,

$$D(\lambda) = \begin{cases} -\frac{1}{2} q(\lambda)^T Q^{\dagger}(\lambda) q(\lambda) + r(\lambda), & \text{if } q(\lambda) \in \mathcal{R}\big(Q(\lambda)\big) \text{ and } Q(\lambda) \succeq 0 \\ -\infty, & \text{otherwise} \end{cases}$$

The dual problem $\max_\lambda g(\lambda)$ is equivalent to

$$\max_{\lambda, d} d,$$

$$\begin{bmatrix} r(\lambda) - d & \frac{1}{\sqrt{2}} q(\lambda)^\top \\ \frac{1}{\sqrt{2}} q(\lambda) & Q(\lambda) \end{bmatrix} \succeq 0.$$

Finding the Lagrangian dual

For other problems like MILP,

$$\begin{array}{lll} D(\lambda) := & \min_x & c^T x + \lambda^T (Ax - b) \\ \text{s.t.} & & Ex \leq d \\ & & x \in X \end{array}$$

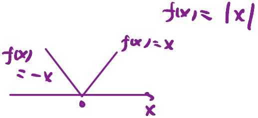
- $D(\lambda)$ is the value function of an MILP, which cannot be (easily) expressed in closed-form.
- In this case, we need an iterative algorithm to find the optimal λ that solves

$$\max_{\lambda \geq 0} D(\lambda)$$

- To this end, let's first analyze the properties of $D(\lambda)$.

Properties of $D(\lambda)$

- **Proposition** $D(\lambda)$ is a
  - concave
  - continuous
  - piecewise linear
  - nonsmooth



function of λ when the problem is an MILP.

Proof

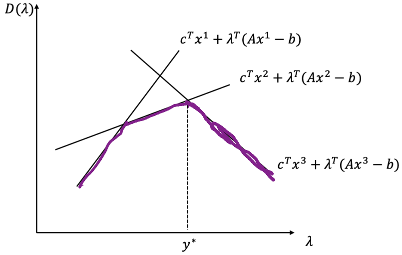
$$\begin{aligned} D(\lambda) &= \min_{x} \quad c^T x + \lambda^T (Ax - b) \quad \text{s.t. } Ex \leq d, x \in X \\ &= \min_{x} \quad c^T x + \lambda^T (Ax - b) \quad \text{s.t. } x \in \text{conv}(S) \\ & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\; \Downarrow \text{ Meyer's theorem} \\ &= \min_{x} \quad c^T x + \lambda^T (Ax - b) \quad \text{s.t. } Hx \leq h \\ &= \min_{k \in K} \quad c^T x^k + \lambda^T (Ax^k - b) \end{aligned}$$

where K is the set of extreme points for the polyhedron set $Hx \leq h$.

Proof continued

$$D(\lambda) = \min_{k \in K} \quad c^T x^k + \lambda^T (Ax^k - b)$$

$D(\lambda)$ is the minimum of linear functions. It is concave by the property, "the maximum over a set of convex functions is convex". Piecewise linear, continuous and nonsmooth properties can be seen easily.



Nonsmooth convex optimization

Since $D(\lambda)$ is nonsmooth concave, we can use any algorithms for solving nonsmooth convex optimization problems. The two most basic algorithms are the cutting plane method and the subgradient method. Other extensions include trust region method, bundle method.

Cutting plane method

$$\max_{\lambda \geq 0} D(\lambda) = \max_{\lambda \geq 0, \eta} \quad \eta$$

$$\text{s.t. } \eta \leq c^T x^k + \lambda^T \left(A x^k - b \right) \quad k \in K$$

where K is the set of extreme points for conv(S). The issue is that there can be combinatorially many extreme points.

Idea: start with a subset of extreme points; then add the "most violated" extreme point.

Suppose we start with a subset of extreme points $K^0 \subseteq K$ and solve the **Lagrangian master problem:**

$$\max_{\lambda \geq 0, \eta} \quad \eta$$

$$\text{s.t. } \eta \leq c^T x^k + \lambda^T \left(A x^k - b \right) \quad k \in K^0$$

Denote the solution to this relaxed problem as η^0, λ^0. The next step is find the extreme point corresponding to the most violated constraint.

Cutting plane method

The most violated constraint is

$$\min_{k \in K} c^T x^k + (\lambda^0)^T \left(Ax^k - b\right) - \eta^0$$

A naive approach is to enumerate all the extreme points. A computationally tractable approach is to solve the MILP

$$\min_{x} c^T x + (\lambda^0)^T (Ax - b) \quad \text{s.t. } Ex \leq d, x \in X$$

This problem is called the **Lagrangian subproblem**. If E is decomposable, this problem is also decomposable.
The solution to the Lagrangian subproblem is an extreme point in the set K. Denote this solution as x^0.
If $c^T x^0 + (\lambda^0)^T \left(Ax^0 - b\right) - \eta^0 \geq 0$, the most violated constraint is still satisfied, cutting plane algorithm has converged. Otherwise, add x^0 to the master problem and keep iterating between the master problem and the subproblem.

Connections with Dantzig Wolfe decomposition

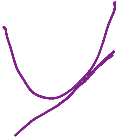
- Both deal with problems with complicating constraints.
- The Lagrangian master problem is the LP dual of the Dantzig Wolfe master problem (leave as a homework question).
- The Lagrangian subproblem is the same as the pricing problem in Dantzig Wolfe decomposition.

Subgradient method

The cutting plane method can take many iterations to converge in practice. Another method is the subgradient method.

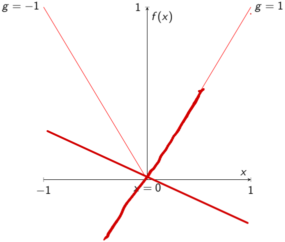
Definition: For a function $f : \mathbb{R}^n \to \mathbb{R}$, a vector $g \in \mathbb{R}^n$ is a *subgradient* of f at a point x_0 if for all $x \in \mathbb{R}^n$,

$$f(x) \geq f(x_0) + g^T(x - x_0).$$



Subgradient at a point x can be a set of vectors. Usually they are represented as $\partial f(x)$

Example $f(x) = |x|$. At $x = 0$, $\partial f(x) = [-1, 1]$

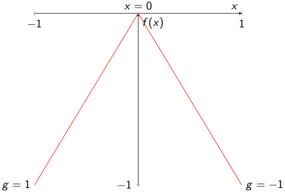


Supergradient

If we have

$$f(x) \leq f(x_0) + g^T(x - x_0).$$

then g is called a supergradient.
Example $f(x) = -|x|$. At $x = 0$, $\partial f(x) = [-1, 1]$



- Subgradient is for convex functions.
- Supergradient is for concave functions.

Subgradient method

Intuition We already know that $D(\lambda)$ is piecewise linear concave. If we can find a supergradient of $D(\lambda)$, we can do "supergradient ascent (subgradient descent)" just like we do gradient ascent (descent) for smooth functions.

Proposition For any $\lambda^0 \geq 0$, denote the solution to the Lagrangian subproblem at λ^0 as x^0

$$D(\lambda^0) = \min_x c^T x + (\lambda^0)^T (Ax - b) \quad \text{s.t. } Ex \leq d, x \in X$$

Then $s^0 = Ax^0 - b$ is a supergradient of the function $D(\lambda)$ at λ^0

Proof

$$\begin{aligned} D(\lambda) =& \min_{k \in K} \quad c^T x^k + \lambda^T (Ax^k - b) \\ \leq& c^T x^0 + \lambda^T (Ax^0 - b) \\ =& c^T x^0 + (\lambda^0)^T (Ax^0 - b) + (\lambda - \lambda^0)^T (Ax^0 - b) \\ =& D(\lambda^0) + (s^0)^T (\lambda - \lambda^0) \end{aligned}$$

Subgradient method

Since s^0 is a supergradient, we can perform a "supergradient ascent" by

$$\lambda^1 = \lambda^0 + \mu s^0$$

where μ is the step size.

1. At iteration λ^k, we solve the Lagrangian subproblem $D(\lambda^k)$ to get the supergradient $s^k = Ax^k - b$ where x^k is the optimal solution to the Lagrangian subproblem.
2. Update by $\lambda^{k+1} = \lambda^k + \mu s^k$
3. repeat

Choosing the step size μ

We typically cannot afford to do line search. One reason is that evaluating $D(\lambda)$ is too expensive (solving an MILP). A heuristic to choose the step size is setting

$$\mu^k = \frac{\alpha^k \left(Z^u - \eta^k\right)}{\|s^k\|_2^2}$$

where $\eta^k = c^T x^k + \left(\lambda^k\right)^T \left(Ax^k - b\right)$, Z^u is an upper bound the Lagrangian relaxation obtained from a heuristic.
α^0 is typically chosen between [0, 2] and is reduced by multiplying $0 < \beta < 1$ if the bounds fail to improve for a given number of iterations.

Lagrangian decomposition

$$
\begin{aligned}
v(P) = \min \quad & c^T x \\
\text{s.t.} \quad & Ax \leq b \\
& Ex \leq d \\
& x \in X
\end{aligned}
$$

Reformulate as

$$
\begin{aligned}
v(P) = \min \quad & \frac{1}{2} c^T (x + y) \\
\text{s.t.} \quad & Ax \leq b \\
& Ey \leq d \\
& x \in X, y \in X \\
& x = y
\end{aligned}
$$

Dualize $(x = y)$ by multiplying Lagrangian multiplier $\nu \in \mathbb{R}^n$

Lagrangian decomposition (dual decomposition)

We end up with two decomposable subproblems.

$$D_x(\nu) := \min \quad \frac{1}{2}c^T x + \nu^T x \qquad D_y(\nu) := \min \quad \frac{1}{2}c^T y - \nu^T y$$
$$\text{s.t. } Ax \le b \qquad\qquad \text{s.t. } Ey \le d$$
$$x \in X \qquad\qquad y \in X$$

Let dual bound be

$$v(LD) := \max_\nu D_x(\nu) + D_y(\nu)$$

Proposition $v(LD) \ge v(LR)$. The bound from Lagrangian decomposition is tighter than Lagrangian relaxation.

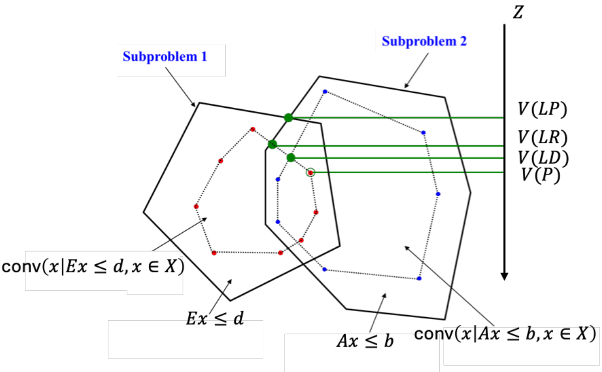
Proof Using the primal characterization.

$$v(LR) = \min_x \left\{ c^T x \mid x \in \{x \mid Ax \le b\} \cap \mathrm{conv}(\{x \mid Ex \le d, x \in X\}) \right\}$$

$$\begin{aligned} v(LD) = \min_{x,y} \Big\{ \frac{1}{2} c^T (x+y) \mid (x,y) \in \{(x,y) \mid x = y\} \cap \\ \mathrm{conv}(\{(x,y) \mid Ax \le b, x \in X,\ Ey \le d, y \in X\}) \Big\} \\ = \min_x \Big\{ c^T x \mid x \in \mathrm{conv}(\{x \mid Ax \le b, x \in X\}) \cap \\ \mathrm{conv}(\{x \mid Ex \le d, x \in X\}) \end{aligned}$$

Since $\mathrm{conv}(\{x \mid Ax \le b, x \in X\}) \subseteq \{x \mid Ax \le b\}$, we have $v(LD) \ge v(LR)$

Geometric intuition



How to obtain a feasible solution?

The Lagrangian subproblems will give a lower bound for the original problem. But we also need a feasible solution (upper bound)

$$D_x(\nu) := \min \quad \frac{1}{2}c^T x + \nu^T x \qquad D_y(\nu) := \min \quad \frac{1}{2}c^T y - \nu^T y$$
$$\text{s.t. } Ax \le b \qquad\qquad\qquad\qquad \text{s.t. } Ey \le d$$
$$x \in X \qquad\qquad\qquad\qquad\qquad y \in X$$

Suppose the solution to the two subproblems are denoted as x^0 and y^0. Note that for MILP we have $x^0 \neq y^0$. Heuristics are used to find a solution that is feasible. For example, we can take $\bar{x} = \frac{1}{2}(x^0 + y^0)$ and round the fractional integer variables. If we are lucky, $\bar{x}$ might be a feasible solution to the original problem. The other commonly used heuristic is to check whether x^0 and y^0 themselves are feasible.

Convergence

- Lagrangian decomposition applied to MILP does not provide zero gap. This is due to the duality gap as well as the arbitrariness of the heuristic. In practice, you can run it for several iterations and hope for the best.
- To guarantee convergence, we need to do some branch and bound on the variables involved in the complicating constraints. See the paper by Carøe and Schultz, 1999.
- Similar property hold for Dantzig Wolfe decomposition when the subproblem are integer constrained. We need to do branch and bound to guarantee convergence. This is called "branch and price" in the literature.

References

- Grossmann, I. E. (2021). Advanced optimization for process systems engineering. Cambridge University Press.
- Conforti, M., Cornuéjols, G., Zambelli, G (2014). Integer programming. Graduate Texts in Mathematics
- Guignard, M. (2003). Lagrangean relaxation. Top, 11, 151-200.
- Carøe, C. C., & Schultz, R. (1999). Dual decomposition in stochastic integer programming. Operations Research Letters, 24(1-2), 37-45.