Solving the Lagrangian Dual

\[ w_{CD} = \min_{\mu \geq 0} \ z(\mu) \]

\[
= \min \left\{ \max_{\mu \geq 0} \right\}_{t=1, \ldots, T} \left( c x_t + \mu (d - D x_t) \right)
\]

i.e. the Lagrangian Dual consists of minimizing a piecewise linear convex but non-differentiable function \( z(\mu) \):

\[ z(\mu) \]

\[ c x_4 + \mu (d - D x_4) \]

\[ c x_3 + \mu (d - D x_3) \]

\[ c x_2 + \mu (d - D x_2) \]

\[ c x_1 + \mu (d - D x_1) \]

The case of a single multiplier \( \mu \) and

\[ X = x_1, x_2, x_3, x_4 \]
How to minimize $z(u)$ over $u \geq 0$?

Solving the corresponding LP is not easy, since it has many (depending on $T$) constraints: constraint generation approach.

An example: (CSP)

Note that if $(P) \quad z = \min c^T x$

$$D x \leq d$$

$$x \in X$$

then $(P_u) \quad z(u) = \min c^T x + u(D x - d)$

for $u \geq 0$

$$x \in X$$

and $(L_u) \quad w_{L,D} = \max \quad z(u)$

for $u \geq 0$

$z(u)$ is now a piecewise linear concave function, to be maximized.
Consider the previous Lagrangian relaxation.

\[ \chi(m) = -\Lambda m + \text{shortest path cost w.r.t. } c_{ij} + \epsilon_{ij} m \]

- Paths

\[ X \text{ is composed of } 4 \text{ solutions: } \]

\[ P_1 = (1, 4) \]
\[ P_2 = (1, 3, 4) \]
\[ P_3 = (1 2 4) \]
\[ P_4 = (1 2 3 4) \]

\[ \text{modified cost } [-2m] \]
\[
Z(\mu) = \min \{ 6 - \mu, 4, x, 3 + \mu \}
\]

\[\mu \in X = \{ P_1, P_2, P_3, P_4 \} \]

* Piecewise linear concave function to be maximize over \( \mu \geq 0 \)*

Geometrically: the maximum (best) Lagrangian bound is 4, obtained for \( \mu \in [1, 2] \);

for \( \mu = 1 \): \( P_4 \) and \( P_2 \) are optimal Lagrangian solutions

for \( \mu \in (0, 1) \): \( P_2 \) is optimal Lagrangian solution

for \( \mu = 2 \): \( P_2 \) and \( P_4 \) are optimal Lagrangian solutions
How to maximize $z(u)$?

(for minimization optimization problems, such as CSP).

Constraint generation approach

Let $B \subseteq X$

$$z_B(u) = \min_{x \in B} c x + u (D x - d)$$

$NB$: $z_B(u)$ dominates $z(u)$ "from above"

\[\text{do}\]

\[\text{MASTER PROBLEM}\]

\[\text{maximize } z_B(u) \text{ over } u \geq 0; \quad \text{let } z^*_B \text{ the maximum value and } u^*_B \text{ the optimal solution;}\]

$NB$: $z^*_B \geq w_{LD}$

\[\text{compute } z(u^*_B) \text{ and let } x^*_B \text{ an optimal solution;}\]

$NB$: $z(u^*_B)$ is a lower bound to $w_{LD}$ (which is the maximum logarithmic bound); $z(u^*_B) < w_{LD}$

$B := B \cup \{x^*_B\}$

$NB$: $x^*_B \in X$

while $z(u^*_B) < z^*_B$
Example (ESP cont.)

Let \( B = \{ p_1, p_2, p_4 \} \)

1) \( z_B(u) \) vs. \( u \)

\[
\begin{align*}
\zeta_B(u) &= \min \left\{ 6 - u, 3 + u \right\} \\
\zeta_B(4.5) &= 4
\end{align*}
\]

\( p_1, p_2, p_4 \)

\( 6 - u \)

\( 3 + u \)

\( u \)

\( u^*_B = 1.5 \)

\( z^*_B = 4.5 \)

- Maximize \( \zeta_B(u) \) over \( u \geq 0 \)

- Compute \( z(1.5) : z(1.5) = 4 \)

- Found \( u^*_B \) corresponds to \( p_2 \)
  \( \text{Separation: add } z \leq 4 \)

- \( B = \{ p_1, p_2, p_4 \} \) \( u + p_2 \cdot y = \gamma \cdot p_1, p_2, p_4 \cdot y \)

Since \( z(1.5) = 4 \leq z^*_B = 4.5 \)

we iterate (with a refined approximation)

2) Now \( \zeta_B(u) = z(u) \), and after its maximization the algorithm stop
Number of iterations: \( \leq |X| \)

Computational efficiency: avoid to consider solutions which are not necessary to compute \( w_{\leq d} \).

*Example:* start with \( B = \{ P_2, P_4 \} \):

In what cases the line corresponding to \( P_3 \) is not generated?

*Example:* LP corresponding to \( B = \{ P_4, P_4 \} \)

\[
\zeta^*_B = \max \eta \quad \omega_{\leq d} = \max \eta
\]

\[
\eta \leq 6 - \mu \quad \eta \leq 6 - \mu
\]

\[
\eta \leq 3 + \mu \quad \text{instead} \quad \eta \leq 3 + \mu
\]

\[
\eta \leq 4
\]

**Master problem**

\[
\mu \geq 0
\]

The separation adds \( \eta \leq 4 \)

\( \mu \geq 0 \)

Relaxation of the Lagrangian Dual!
An alternative approach (without using LP solvers) is to use a Subgradient Algorithm, which is designed to minimize a piecewise linear convex function (or maximize a piecewise linear concave function).

Consider the Lagrangian Dual formulation:

$$\omega_{\text{D}} = \min_{\mu \geq 0} \max_{\epsilon = 1, \ldots, T} \left( c^\top \epsilon + \mu (d - D \epsilon) \right)$$

As already observed, \( z(\mu) \) is piecewise linear convex, but generally it is not differentiable. Therefore, it is possible to use a Subgradient Algorithm.
A subgradient is a generalization of gradient:

**Definition:** Given a convex function $f : \mathbb{R}^m \to \mathbb{R}$, a subgradient of $f$ at $u$ is a vector $\gamma(u) \in \mathbb{R}^m$ such that:

$$f(v) \geq f(u) + \gamma(u)^T(v - u) \quad \forall v \in \mathbb{R}^m.$$ 

If $f$ is a continuously differentiable convex function, then $\gamma(u) = \nabla f(u) = \left( \frac{\partial f}{\partial u_1}, \ldots, \frac{\partial f}{\partial u_m} \right)$ is the gradient of $f$ at $u$.

**Example**

Graphically, $\gamma(u)$ is simply the line segment that connects $u$ to $v$ and lies on or below the tangent plane at $u$. It defines the set of subgradients at $u$. 
slope of a supporting hyperplane to the graph of $f$ at $u$; if $f$ is differentiable at $u$, the only such plane is the tangent plane, and so $f(u) = \nabla f(u)$. Property 0 is a subgradient of $f$ at $u$ if and only if $u$ minimizes $f$.

Proof:
0 is subgradient at $u$ if and only if
$$f(v) \geq f(u) + 0(v - u) + v, \text{ i.e. } f(v) \geq f(u) + v.$$ 

Definition: The (one-sided) directional derivative of $f$ at $u$ in the direction $s$ is
$$Df(u;s) = \lim_{x \to 0^+} \frac{f(u + xs) - f(u)}{x}.$$
So, if \( Df(u; s) > 0 \), then we increase \( f(u) \) by performing a step along \( s \).

In particular:

**Theorem:** \( u \) minimizes \( f \) if and only if \( Df(u; s) > 0 \) for all directions \( s \).

What is the relationship between directional derivatives and subgradients:

**Theorem:**

\[
Df(u; s) = \max \{ f(u)^T s : f(u) \text{ subgradient of } f \text{ at } u \}.
\]

No proof.

See the figure: if \( s \) is the direction in the figure \( (s = (1)) \), then
$Df(u; s)$ is the right-derivative of $f$ at $u$, which is the slope of the supporting line at $u$ turned counterclockwise as far as possible; i.e., $\max g'(u); -g'(u)$ subgradient at $u$. $s$

Similarly, if $\mathbf{s} = (-1)$, $Df(u; s) =$

$= \max g'(u)s; -g'(u)$ subgradient at $u$$

= \max g' - g'(u); -g'(u)$

$g'(u)$ subgradient at $u$.

Obs: If $f$ is differentiable at $u$;

$$Df(u; s) = \nabla f(u)s$$
unique subgradient at $u$

Algorithmic implications: in order to minimize $f$, choose the direction $s = -g(u)$, since $g(u)^T s = -g(u)^T g(u) \leq 0$,
and so it is possible that $Df(u; s) < 0$, allowing to decrease $g(u)$.

Subgradient Algorithm.
Subgradient Algorithm

Initialization: \( u = u^0 \).

Iteration \( k: \) \( u = u^k \)

- Solve the Lagrangian relaxation \( (P_{uk}) \), and find an optimal solution \( x(u^k) \) of \( (P_{uk}) \).
- \( u^{k+1} = \max f(u^k - \lambda_k (d - Dx(u^k))) \), \( \lambda \geq 0 \).
- \( k = k + 1 \)

That is: at each iteration the algorithm moves from the current point \( u^k \) along the direction opposite to \( (d - Dx(u^k)) \), which is a subgradient of function \( z(u) \) at \( u^k \):

- \( \max f(u^k - \lambda_k (d - Dx(u^k))) \), \( \lambda \geq 0 \) since \( u^{k+1} \) must be \( \geq 0 \);
- the difficulty is to choose the step lengths \( \lambda_k \).
Property: \((d - D x(u^k))\) is a subgradient of \(z(u)\) at \(u^k\).

Proof:

\[
\begin{align*}
z(u^k) &= \max_{\ell = 1, \ldots, T} x_{\ell} + u^k(d - D x_{\ell}) \\
&= c x(u^k) + u^k(d - D x(u^k))
\end{align*}
\]

In fact, \(x(u^k)\) is an optimal solution of \(P(u^k)\).

For any \(v \geq 0\):

\[
\begin{align*}
z(v) &= \max_{\ell = 1, \ldots, T} x_{\ell} + v(d - D x_{\ell}) \\
&= c x(u^k) + v(d - D x(u^k)).
\end{align*}
\]

So

\[
z(v) \geq c x(u^k) + v(d - D x(u^k)) + u^k(d - D x(u^k)) - u^k(d - D x(u^k))
\]

\[
= z(u^k) + (d - D x(u^k))(v - u^k)
\]

subgradient

\[
at u^k
\]

\[\Box\]
How to choose \( \lambda_k \):

- If they are too small, the algorithm can stay "near" the current point and not converge.
- If they are too large, the algorithm may skip the optimal solution.

Compromise:

1) \( \sum_k \lambda_k \to 0 \) when \( k \to +\infty \)
2) \( \sum_{k=1}^{K} \lambda_k \to +\infty \) when \( K \to +\infty \)

These conditions ensure that the Subgradient Algorithm converges to \( \omega_LD \) (no proof).

Example: \( \lambda_k = \frac{1}{k} \quad \forall \ K \geq 1 \)
example (ESP) cont.

Let \( u^0 = 3 \)

Iteration (\( u = 3 \))

- Solve (\( P_3 \)), finding \( z(3) = (6 - 3) = 3 \)
- and the optimal solution \( x(3) \) which is the path \( P_1 \) (see the figure in (50))

- The complicating constraint \( D x \leq d \) is
\[
\sum_{(i,j) \in A} e_{ij} x_{ij} \leq \frac{L}{2} ;
\]

(CSP) is a minimization optimization problem, and we want to maximize \( z(u) \), so we move along \( (D x(u^k) - d) \) which is a subgradient of the concave function \( z(u) \) (or simply a subgradient):

\[
\sum_{(i,j) \in A} e_{ij} x_{ij}(3) - L = \sum_{(i,j) \in P_3 = (1, 3)} e_{ij} - 2 = 1 - 2 = -1
\]

So \( u_1 = \max \left[ u^0 + \lambda_1 (-1), 0 \right] \) at \( u = 3 \)

\[
= \max \left[ 3 - \lambda_1, 0 \right] \]

\( = 3 \)
If \( k_1 = 1 \), then:

\[
\begin{align*}
u^1 &= 2 \\
z(1) &= 3
\end{align*}
\]

\[\mathbf{K}_1 = 1\]

Iteration (\( \mu = 2 \))

- Solve \((P_2)\), finding \( z(2) = 4 \) and an optimal solution \( x(2) \);
- If \( x(2) \) is the path \( P_2 \), then \( \sum_{e_{1d} \in P_2} e_{1d} - 2 = 0 \) \( \underline{\mathbf{Maximum!}} \)
- \( \mu^2 = \mu^1 \) and the algorithm STOP

In general: the algorithm is often terminated before \( \mu_{1c} \) is attained.
Exercise:

\[ \text{If } (P) \quad \min_{x} \quad c_{x} \]
\[ \text{subject to } \quad d \times x \leq d \]
\[ x \in X \]

prove that \((D \times (u^{k}) - d)\) is a supergradient of \(z(u)\) at \(u^{k}\), i.e.,

\[ z(v) \leq z(u^{k}) + (D \times (u^{k}) - d)(v - u^{k}) \]

\[ \neq u^{k} \in \mathbf{0} \]

recall that \(x(u^{k})\) denotes an optimal solution of the Lagrangian relaxation \(P_{u^{k}}\).