Decomposition Methods

- separable problems, complicating variables
- primal decomposition
- dual decomposition
- complicating constraints
- general decomposition structures

Prof. S. Boyd, EE364b, Stanford University
Separable problem

minimize \[ f_1(x_1) + f_2(x_2) \]
subject to \( x_1 \in C_1, \quad x_2 \in C_2 \)

- we can solve for \( x_1 \) and \( x_2 \) separately (in parallel)
- even if they are solved sequentially, this gives advantage if computational effort is superlinear in problem size
- called separable or trivially parallelizable
- generalizes to any objective of form \( \Psi(f_1, f_2) \) with \( \Psi \) nondecreasing (\( e.g., \) max)
Complicating variable

consider unconstrained problem,

\[
\text{minimize } f(x) = f_1(x_1, y) + f_2(x_2, y)
\]

\[x = (x_1, x_2, y)\]

- \(y\) is the \textbf{complicating variable} or \textbf{coupling variable}; when it is fixed the problem is separable in \(x_1\) and \(x_2\)

- \(x_1, x_2\) are \textbf{private} or \textbf{local} variables; \(y\) is a \textbf{public} or \textbf{interface} or \textbf{boundary} variable between the two subproblems
Primal decomposition

fix \( y \) and define

\[
\text{subproblem 1: } \min_{x_1} f_1(x_1, y) \\
\text{subproblem 2: } \min_{x_2} f_2(x_2, y)
\]

with optimal values \( \phi_1(y) \) and \( \phi_2(y) \)

original problem is equivalent to master problem

\[
\min_{y} \phi_1(y) + \phi_2(y)
\]

with variable \( y \)

called primal decomposition since master problem manipulates primal (complicating) variables
• if original problem is convex, so is master problem

• can solve master problem using
  – bisection (if \( y \) is scalar)
  – gradient or Newton method (if \( \phi_i \) differentiable)
  – subgradient, cutting-plane, or ellipsoid method

• each iteration of master problem requires solving the two subproblems

• if master algorithm converges fast enough and subproblems are sufficiently easier to solve than original problem, we get savings
Primal decomposition algorithm

(using subgradient algorithm for master)

repeat

1. Solve the subproblems.
   - Find $x_1$ that minimizes $f_1(x_1, y)$, and a subgradient $g_1 \in \partial \phi_1(y)$.
   - Find $x_2$ that minimizes $f_2(x_2, y)$, and a subgradient $g_2 \in \partial \phi_2(y)$.

2. Update complicating variable.
   $y := y - \alpha_k (g_1 + g_2)$.

step length $\alpha_k$ can be chosen in any of the standard ways
Example

• $x_1, x_2 \in \mathbb{R}^{20}, y \in \mathbb{R}$

• $f_i$ are PWL (max of 100 affine functions each); $f^* \approx 1.71$
primal decomposition, using bisection on $y$
Dual decomposition

Step 1: introduce new variables $y_1$, $y_2$

$$\text{minimize } f(x) = f_1(x_1, y_1) + f_2(x_2, y_2)$$
$$\text{subject to } y_1 = y_2$$

- $y_1, y_2$ are local versions of complicating variable $y$
- $y_1 = y_2$ is consistency constraint
Step 2: form dual problem

\[ L(x_1, y_1, x_2, y_2) = f_1(x_1, y_1) + f_2(x_2, y_2) + \nu^T(y_1 - y_2) \]

separable; can minimize over \((x_1, y_1)\) and \((x_2, y_2)\) separately

\[
\begin{align*}
    g_1(\nu) & = \inf_{x_1, y_1} (f_1(x_1, y_1) + \nu^T y_1) = -f_1^*(0, -\nu) \\
    g_2(\nu) & = \inf_{x_2, y_2} (f_2(x_2, y_2) - \nu^T y_2) = -f_2^*(0, \nu)
\end{align*}
\]

dual problem is: maximize \( g(\nu) = g_1(\nu) + g_2(\nu) \)

- computing \( g_i(\nu) \) are the **dual subproblems**
- can be done in parallel
- a subgradient of \(-g\) is \( y_2 - y_1 \) (from solutions of subproblems)
Dual decomposition algorithm

(using subgradient algorithm for master)

repeat
1. Solve the dual subproblems.
   Find $x_1, y_1$ that minimize $f_1(x_1, y_1) + \nu^T y_1$.
   Find $x_2, y_2$ that minimize $f_2(x_2, y_2) - \nu^T y_2$.
2. Update dual variables (prices).
   $\nu := \nu - \alpha_k (y_2 - y_1)$.

• step length $\alpha_k$ can be chosen in standard ways
• at each step we have a lower bound $g(\nu)$ on $p^*$
• iterates are generally infeasible, i.e., $y_1 \neq y_2$
Finding feasible iterates

• reasonable guess of feasible point from \((x_1, y_1), (x_2, y_2)\):

\[
(x_1, \bar{y}), \quad (x_2, \bar{y}), \quad \bar{y} = (y_1 + y_2)/2
\]

  – projection onto feasible set \(y_1 = y_2\)
  – gives upper bound \(p^* \leq f_1(x_1, \bar{y}) + f_2(x_2, \bar{y})\)

• a better feasible point: replace \(y_1, y_2\) with \(\bar{y}\) and solve primal subproblems minimize\(_{x_1} f_1(x_1, \bar{y}), \text{ minimize}_{x_2} f_2(x_2, \bar{y})\)

  – gives (better) upper bound \(p^* \leq \phi_1(\bar{y}) + \phi_2(\bar{y})\)
(Same) example
dual decomposition convergence (using bisection on $\nu$)
Interpretation

- $y_1$ is resources consumed by first unit, $y_2$ is resources generated by second unit
- $y_1 = y_2$ is **consistency** condition: supply equals demand
- $\nu$ is a set of resource prices
- master algorithm adjusts prices at each step, rather than allocating resources directly (primal decomposition)
Recovering the primal solution from the dual

• iterates in dual decomposition:

\[ \nu^{(k)}, \quad (x_1^{(k)}, y_1^{(k)}), \quad (x_2^{(k)}, y_2^{(k)}) \]

- \( x_1^{(k)}, y_1^{(k)} \) is minimizer of \( f_1(x_1, y_1) + \nu^{(k)T}y_1 \) found in subproblem 1
- \( x_2^{(k)}, y_2^{(k)} \) is minimizer of \( f_2(x_2, y_2) - \nu^{(k)T}y_2 \) found in subproblem 2

• \( \nu^{(k)} \to \nu^* \) (i.e., we have price convergence)

• subtlety: we need not have \( y_1^{(k)} - y_2^{(k)} \to 0 \)

• the hammer: if \( f_i \) strictly convex, we have \( y_1^{(k)} - y_2^{(k)} \to 0 \)

• can fix allocation (i.e., compute \( \phi_i \)), or add regularization terms \( \epsilon \|y_i\|^2 \)
Decomposition with constraints

can also have **complicating constraints**, as in

\[
\begin{align*}
\text{minimize} & \quad f_1(x_1) + f_2(x_2) \\
\text{subject to} & \quad x_1 \in C_1, \quad x_2 \in C_2 \\
& \quad h_1(x_1) + h_2(x_2) \leq 0
\end{align*}
\]

- \( f_i, h_i, C_i \) convex

- \( h_1(x_1) + h_2(x_2) \leq 0 \) is a set of \( p \) complicating or coupling constraints, involving both \( x_1 \) and \( x_2 \)

- can interpret coupling constraints as limits on resources shared between two subproblems
Primal decomposition

fix $t \in \mathbb{R}^p$ and define

subproblem 1: minimize $f_1(x_1)$
subject to $x_1 \in \mathcal{C}_1$, $h_1(x_1) \preceq t$

subproblem 2: minimize $f_2(x_2)$
subject to $x_2 \in \mathcal{C}_2$, $h_2(x_2) \preceq -t$

• $t$ is the quantity of resources allocated to first subproblem
  ($-t$ is allocated to second subproblem)

• master problem: minimize $\phi_1(t) + \phi_2(t)$ (optimal values of
  subproblems) over $t$

• subproblems can be solved separately when $t$ is fixed
Primal decomposition algorithm

repeat
  1. Solve the subproblems.
     Solve subproblem 1, finding $x_1$ and $\lambda_1$.
     Solve subproblem 2, finding $x_2$ and $\lambda_2$.
  2. Update resource allocation.
     $t := t - \alpha_k(\lambda_2 - \lambda_1)$.

• $\lambda_i$ is an optimal Lagrange multiplier associated with resource constraint in subproblem $i$

• $\lambda_2 - \lambda_1 \in \partial(\phi_1 + \phi_2)(t)$

• $\alpha_k$ is an appropriate step size

• all iterates are feasible (when subproblems are feasible)
Example

- $x_1, x_2 \in \mathbb{R}^{20}$, $t \in \mathbb{R}^2$; $f_i$ are quadratic, $h_i$ are affine, $C_i$ are polyhedra defined by 100 inequalities; $p^* \approx -1.33$; $\alpha_k = 0.5/k$
resource allocation $t$ to first subsystem (second subsystem gets $-t$)
Dual decomposition

form (separable) partial Lagrangian

\[ L(x_1, x_2, \lambda) = f_1(x_1) + f_2(x_2) + \lambda^T(h_1(x_1) + h_2(x_2)) \]
\[ = (f_1(x_1) + \lambda^T h_1(x_1)) + (f_2(x_2) + \lambda^T h_2(x_2)) \]

fix dual variable \( \lambda \) and define

subproblem 1: \[ \text{minimize} \quad f_1(x_1) + \lambda^T h_1(x_1) \]
subject to \[ x_1 \in C_1 \]

subproblem 2: \[ \text{minimize} \quad f_2(x_2) + \lambda^T h_2(x_2) \]
subject to \[ x_2 \in C_2 \]

with optimal values \( g_1(\lambda), g_2(\lambda) \)
• $h_i(\bar{x}_i) \in \partial(-g_i)(\lambda)$, where $\bar{x}_i$ is any solution to subproblem $i$

• $h_1(\bar{x}_1) + h_2(\bar{x}_2) \in \partial(-g)(\lambda)$

• the master algorithm updates $\lambda$ using this subgradient
Dual decomposition algorithm

(using projected subgradient method)

repeat
  1. Solve the subproblems.
     - Solve subproblem 1, finding an optimal \( \bar{x}_1 \).
     - Solve subproblem 2, finding an optimal \( \bar{x}_2 \).
  2. Update dual variables (prices).
     \[ \lambda := (\lambda + \alpha_k (h_1(\bar{x}_1) + h_2(\bar{x}_2)))_+ \].

• \( \alpha_k \) is an appropriate step size

• iterates need not be feasible

• can again construct feasible primal variables using projection
Interpretation

- $\lambda$ gives prices of resources
- Subproblems are solved separately, taking income/expense from resource usage into account
- Master algorithm adjusts prices
- Prices on over-subscribed resources are increased; prices on undersubscribed resources are reduced, but never made negative
(Same) example

subgradient method for master; resource prices $\lambda$

Prof. S. Boyd, EE364b, Stanford University
dual decomposition convergence; $\hat{f}$ is objective of projected feasible allocation
$$p^* - g(\lambda)$$
$$\tilde{f} - g(\lambda)$$
General decomposition structures

- multiple subsystems

- (variable and/or constraint) coupling constraints between subsets of subsystems

- represent as hypergraph with subsystems as vertices, coupling as hyperedges or nets

- without loss of generality, can assume all coupling is via consistency constraints
Simple example

• 3 subsystems, with private variables $x_1$, $x_2$, $x_3$, and public variables $y_1$, $(y_2, y_3)$, and $y_4$

• 2 (simple) edges

\[
\begin{align*}
\text{minimize} & \quad f_1(x_1, y_1) + f_2(x_2, y_2, y_3) + f_3(x_3, y_4) \\
\text{subject to} & \quad (x_1, y_1) \in C_1, \quad (x_2, y_2, y_3) \in C_2, \quad (x_3, y_4) \in C_3 \\
& \quad y_1 = y_2, \quad y_3 = y_4
\end{align*}
\]
A more complex example
General form

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{K} f_i(x_i, y_i) \\
\text{subject to} & \quad (x_i, y_i) \in C_i, \quad i = 1, \ldots, K \\
& \quad y_i = E_i z, \quad i = 1, \ldots, K
\end{align*}
\]

- private variables \(x_i\), public variables \(y_i\)
- net (hyperedge) variables \(z \in \mathbb{R}^N; z_i\) is common value of public variables in net \(i\)
- matrices \(E_i\) give netlist or hypergraph
  row \(k\) is \(e_p\), where \(k\)th entry of \(y_i\) is in net \(p\)
Primal decomposition

$\phi_i(y_i)$ is optimal value of subproblem

\[
\begin{align*}
\text{minimize} & \quad f_i(x_i, y_i) \\
\text{subject to} & \quad (x_i, y_i) \in C_i
\end{align*}
\]

repeat

1. Distribute net variables to subsystems.
   \[y_i := E_i z, \quad i = 1, \ldots, K.\]
2. Optimize subsystems (separately).
   Solve subproblems to find optimal \(x_i, g_i \in \partial \phi_i(y_i), \quad i = 1, \ldots, K.\)
3. Collect and sum subgradients for each net.
   \[g := \sum_{i=1}^{K} E_i^T g_i.\]
4. Update net variables.
   \[z := z - \alpha_k g.\]
Dual decomposition

\[ g_i(\nu_i) \] is optimal value of subproblem

\[
\begin{align*}
\text{minimize} & \quad f_i(x_i, y_i) + \nu_i^T y_i \\
\text{subject to} & \quad (x_i, y_i) \in C_i
\end{align*}
\]

given initial price vector \( \nu \) that satisfies \( E^T \nu = 0 \) (e.g., \( \nu = 0 \)).

repeat

1. Optimize subsystems (separately).
   Solve subproblems to obtain \( x_i, y_i \).
2. Compute average value of public variables over each net.
   \[ \hat{z} := (E^T E)^{-1} E^T y. \]
3. Update prices on public variables.
   \[ \nu := \nu + \alpha_k (y - E \hat{z}). \]
A more complex example

subsystems: quadratic plus PWL objective with 10 private variables; 9 public variables and 4 nets; $p^* \approx 11.1; \alpha = 0.5$
consistency constraint residual $\|y - E\hat{z}\|$ versus iteration number