Subgradients

- subgradients
- strong and weak subgradient calculus
- optimality conditions via subgradients
- directional derivatives

Basic inequality

recall basic inequality for convex differentiable f:

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

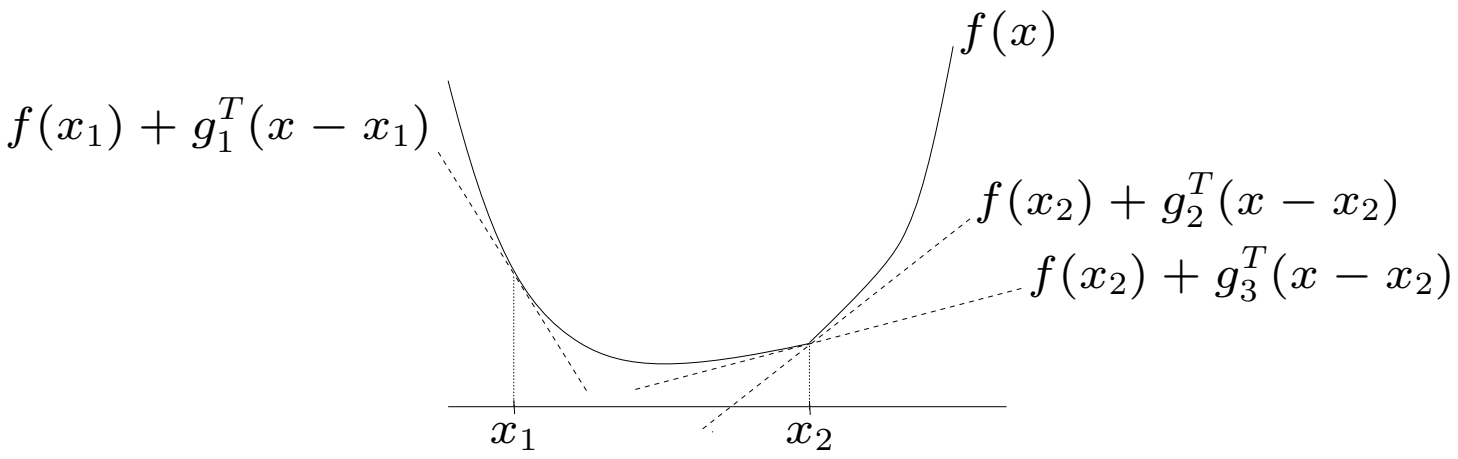
- first-order approximation of f at x is global underestimator
- $(\nabla f(x), -1)$ supports $\mathbf{epi}\, f$ at $(x, f(x))$

what if f is not differentiable?

Subgradient of a function

g is a **subgradient** of f (not necessarily convex) at x if

$$f(y) \geq f(x) + g^T(y - x) \quad \text{for all } y$$



g_2, g_3 are subgradients at x_2; g_1 is a subgradient at x_1

- g is a subgradient of f at x iff $(g, -1)$ supports $\mathbf{epi}\, f$ at $(x, f(x))$
- g is a subgradient iff $f(x) + g^T(y - x)$ is a global (affine) underestimator of f
- if f is convex and differentiable, $\nabla f(x)$ is a subgradient of f at x

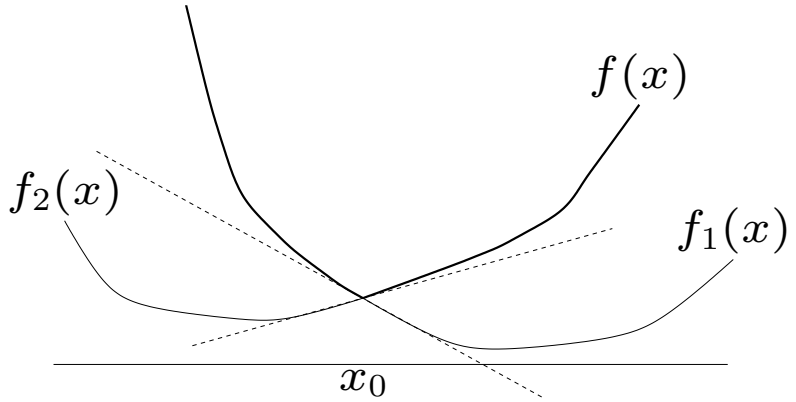
subgradients come up in several contexts:

- algorithms for nondifferentiable convex optimization
- convex analysis, $e.g.$, optimality conditions, duality for nondifferentiable problems

(if $f(y) \leq f(x) + g^T(y - x)$ for all y, then g is a **supergradient**)

Example

$f = \max\{f_1, f_2\}$, with f_1, f_2 convex and differentiable



- $f_1(x_0) > f_2(x_0)$: unique subgradient $g = \nabla f_1(x_0)$
- $f_2(x_0) > f_1(x_0)$: unique subgradient $g = \nabla f_2(x_0)$
- $f_1(x_0) = f_2(x_0)$: subgradients form a line segment $[\nabla f_1(x_0), \nabla f_2(x_0)]$

Subdifferential

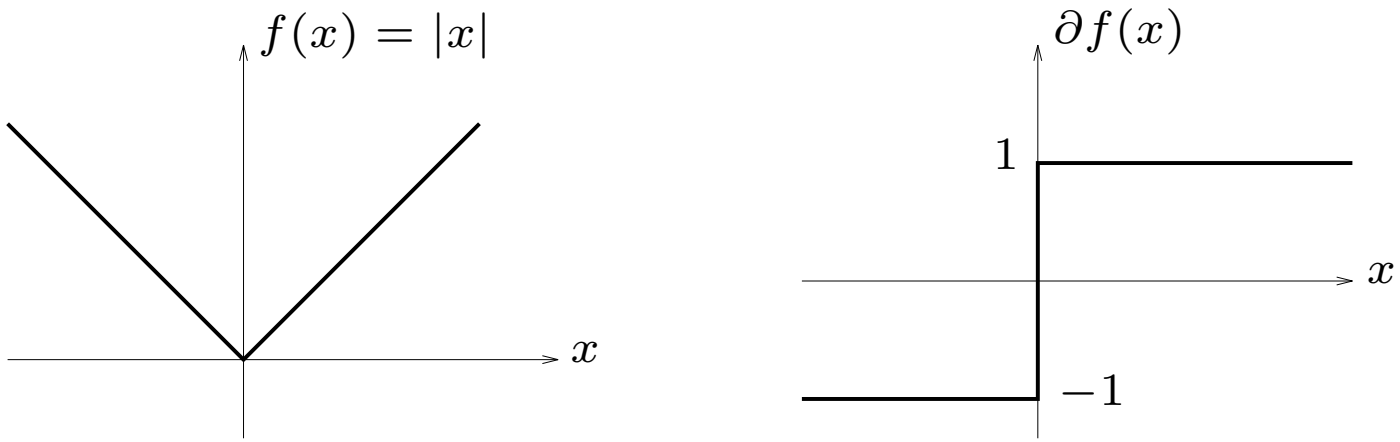
- set of all subgradients of f at x is called the **subdifferential** of f at x, denoted $\partial f(x)$
- $\partial f(x)$ is a closed convex set (can be empty)

if f is convex,

- $\partial f(x)$ is nonempty, for $x \in \mathbf{relint}\,\mathbf{dom}\, f$
- $\partial f(x) = \{\nabla f(x)\}$, if f is differentiable at x
- if $\partial f(x) = \{g\}$, then f is differentiable at x and $g = \nabla f(x)$

Example

$f(x) = |x|$



righthand plot shows $\bigcup \{(x, g) \mid x \in \mathbf{R},\ g \in \partial f(x)\}$

Subgradient calculus

- **weak subgradient calculus**: formulas for finding *one* subgradient $g \in \partial f(x)$
- **strong subgradient calculus**: formulas for finding the whole subdifferential $\partial f(x)$, *i.e.*, *all* subgradients of f at x
- many algorithms for nondifferentiable convex optimization require only *one* subgradient at each step, so weak calculus suffices
- some algorithms, optimality conditions, etc., need whole subdifferential
- roughly speaking: if you can compute $f(x)$, you can usually compute a $g \in \partial f(x)$
- we'll assume that f is convex, and $x \in \mathbf{relint}\,\mathbf{dom}\, f$

Some basic rules

- $\partial f(x) = \{\nabla f(x)\}$ if f is differentiable at x
- **scaling:** $\partial(\alpha f) = \alpha \partial f$ (if $\alpha > 0$)
- **addition:** $\partial(f_1 + f_2) = \partial f_1 + \partial f_2$ (RHS is addition of sets)
- **affine transformation of variables:** if $g(x) = f(Ax + b)$, then $\partial g(x) = A^T \partial f(Ax + b)$
- **finite pointwise maximum:** if $f = \max_{i=1,\ldots,m} f_i$, then

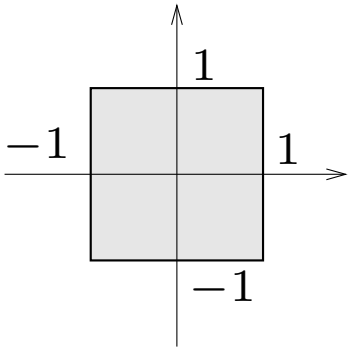
$$\partial f(x) = \mathbf{Co} \bigcup \{\partial f_i(x) \mid f_i(x) = f(x)\},$$

  $i.e.$, convex hull of union of subdifferentials of 'active' functions at x

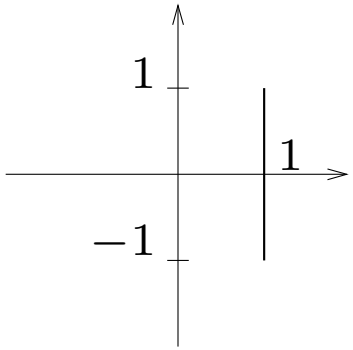
$f(x) = \max\{f_1(x), \ldots, f_m(x)\}$, with $f_1, \ldots, f_m$ differentiable

$$\partial f(x) = \mathbf{Co}\{\nabla f_i(x) \mid f_i(x) = f(x)\}$$

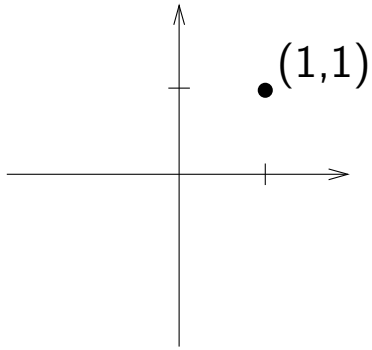
example: $f(x) = \|x\|_1 = \max\{s^T x \mid s_i \in \{-1, 1\}\}$



$\partial f(x)$ at $x = (0, 0)$



at $x = (1, 0)$

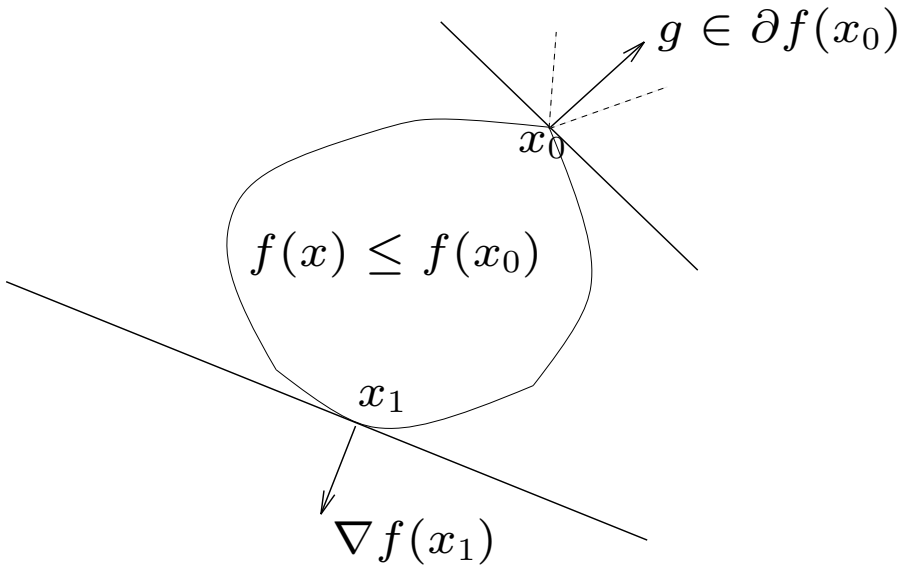


at $x = (1, 1)$

Subgradients and sublevel sets

g is a subgradient at x means $f(y) \geq f(x) + g^T(y - x)$

hence $f(y) \leq f(x) \Longrightarrow g^T(y - x) \leq 0$



- f differentiable at x_0: $\nabla f(x_0)$ is normal to the sublevel set $\{x \mid f(x) \leq f(x_0)\}$

- f nondifferentiable at x_0: subgradient defines a supporting hyperplane to sublevel set through x_0

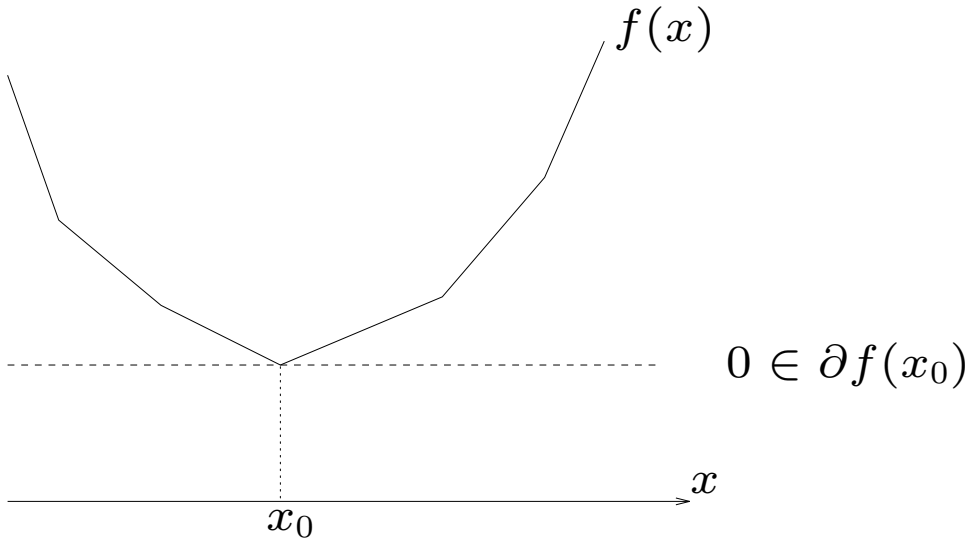
Optimality conditions — unconstrained

recall for f convex, differentiable,

$$f(x^\star) = \inf_x f(x) \Longleftrightarrow 0 = \nabla f(x^\star)$$

generalization to nondifferentiable convex f:

$$f(x^\star) = \inf_x f(x) \Longleftrightarrow 0 \in \partial f(x^\star)$$



proof. by definition (!)

$$f(y) \geq f(x^\star) + 0^T(y - x^\star) \text{ for all } y \iff 0 \in \partial f(x^\star)$$

. . . seems trivial but isn't

Example: piecewise linear minimization

$f(x) = \max_{i=1,\ldots,m}(a_i^T x + b_i)$

$x^\star$ minimizes $f \iff 0 \in \partial f(x^\star) = \mathbf{Co}\{a_i \mid a_i^T x^\star + b_i = f(x^\star)\}$

$\iff$ there is a λ with

$$\lambda \succeq 0, \qquad \mathbf{1}^T \lambda = 1, \qquad \sum_{i=1}^{m} \lambda_i a_i = 0$$

where $\lambda_i = 0$ if $a_i^T x^\star + b_i < f(x^\star)$

. . . but these are the KKT conditions for the epigraph form

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & a_i^T x + b_i \leq t, \quad i = 1, \ldots, m \end{array}$$

with dual

$$\begin{array}{lll} \text{maximize} & b^T \lambda & \\ \text{subject to} & \lambda \succeq 0, \qquad A^T \lambda = 0, & \mathbf{1}^T \lambda = 1 \end{array}$$

Optimality conditions — constrained

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0,\ i = 1, \ldots, m \end{array}$$

we assume

- f_i convex, defined on $\mathbf{R}^n$ (hence subdifferentiable)
- strict feasibility (Slater's condition)

$x^\star$ is primal optimal ($\lambda^\star$ is dual optimal) iff

$$\begin{aligned} & f_i(x^\star) \leq 0, \quad \lambda_i^\star \geq 0 \\ & 0 \in \partial f_0(x^\star) + \textstyle\sum_{i=1}^m \lambda_i^\star \partial f_i(x^\star) \\ & \lambda_i^\star f_i(x^\star) = 0 \end{aligned}$$

. . . generalizes KKT for nondifferentiable f_i

Directional derivative

directional derivative of f at x in the direction δx is

$$f'(x;\delta x) \stackrel{\Delta}{=} \lim_{h \searrow 0} \frac{f(x+h\delta x) - f(x)}{h}$$

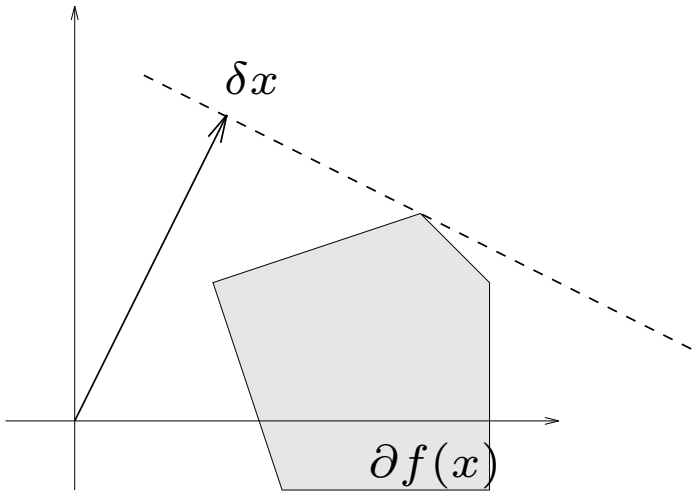
can be $+\infty$ or $-\infty$

- f convex, finite near $x \Longrightarrow f'(x;\delta x)$ exists

- f differentiable at x if and only if, for some g $(= \nabla f(x))$ and all δx, $f'(x;\delta x) = g^T \delta x$ ($i.e.$, $f'(x;\delta x)$ is a linear function of δx)

Directional derivative and subdifferential

general formula for convex f: $f'(x;\delta x) = \sup_{g \in \partial f(x)} g^T \delta x$



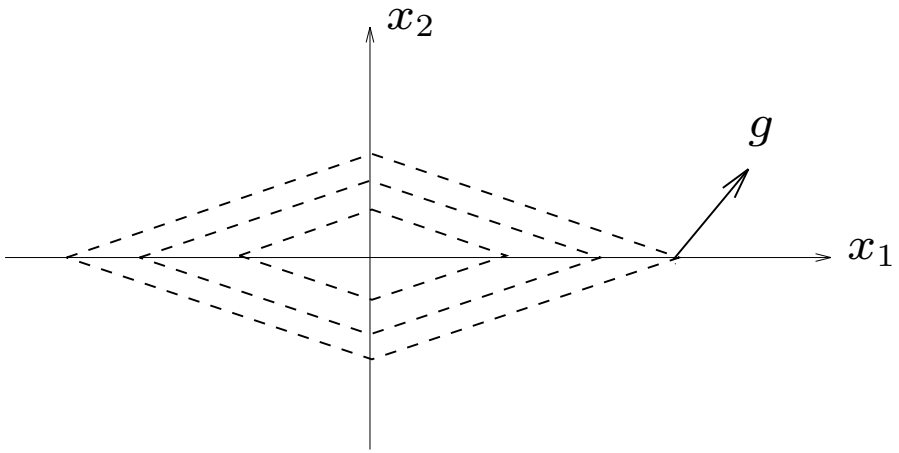
Descent directions

δx is a **descent direction** for f at x if $f'(x;\delta x) < 0$

for differentiable f, $\delta x = -\nabla f(x)$ is always a descent direction (except when it is zero)

warning: for nondifferentiable (convex) functions, $\delta x = -g$, with $g \in \partial f(x)$, need not be descent direction

example: $f(x) = |x_1| + 2|x_2|$



Subgradients and distance to sublevel sets

if f is convex, $f(z) < f(x)$, $g \in \partial f(x)$, then for small $t > 0$,

$$\|x - tg - z\|_2 < \|x - z\|_2$$

thus $-g$ is descent direction for $\|x - z\|_2$, for **any** z with $f(z) < f(x)$ ($e.g.$, $x^\star$)

negative subgradient is descent direction for distance to optimal point

proof:
$$\begin{aligned} \|x - tg - z\|_2^2 &= \|x - z\|_2^2 - 2tg^T(x - z) + t^2\|g\|_2^2 \\ &\leq \|x - z\|_2^2 - 2t(f(x) - f(z)) + t^2\|g\|_2^2 \end{aligned}$$

Descent directions and optimality

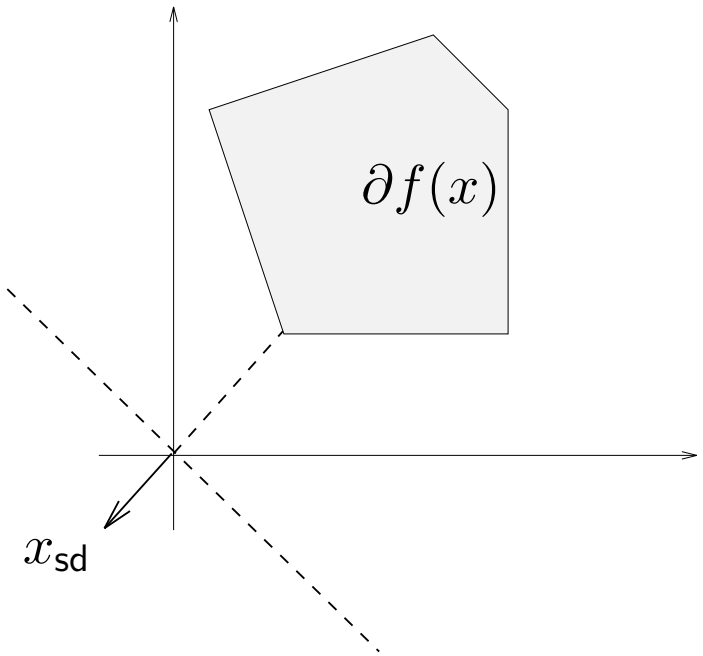
fact: for f convex, finite near x, either

- $0 \in \partial f(x)$ (in which case x minimizes f), or
- there is a descent direction for f at x

i.e., x is optimal (minimizes f) iff there is no descent direction for f at x

proof: define $\delta x_{\mathrm{sd}} = -\operatorname*{argmin}_{z \in \partial f(x)} \|z\|_2$

if $\delta x_{\mathrm{sd}} = 0$, then $0 \in \partial f(x)$, so x is optimal; otherwise
$f'(x; \delta x_{\mathrm{sd}}) = -\left(\inf_{z \in \partial f(x)} \|z\|_2\right)^2 < 0$, so δx_{sd} is a descent direction



idea extends to constrained case (feasible descent direction)

Subgradient method

subgradient method is simple algorithm to minimize nondifferentiable convex function f

$$x^{(k+1)} = x^{(k)} - \alpha_k g^{(k)}$$

- $x^{(k)}$ is the kth iterate
- $g^{(k)}$ is **any** subgradient of f at $x^{(k)}$
- $\alpha_k > 0$ is the kth step size

not a descent method, so we keep track of best point so far

$$f_{\rm best}^{(k)} = \min_{i=1,\ldots,k} f(x^{(i)})$$

Step size rules

step sizes are fixed ahead of time

- *constant step size:* $\alpha_k = \alpha$ (constant)
- *constant step length:* $\alpha_k = \gamma/\|g^{(k)}\|_2$ (so $\|x^{(k+1)} - x^{(k)}\|_2 = \gamma$)
- *square summable but not summable:* step sizes satisfy

$$\sum_{k=1}^{\infty} \alpha_k^2 < \infty, \qquad \sum_{k=1}^{\infty} \alpha_k = \infty$$

- *nonsummable diminishing:* step sizes satisfy

$$\lim_{k\to\infty} \alpha_k = 0, \qquad \sum_{k=1}^{\infty} \alpha_k = \infty$$

Subgradient Methods for Constrained Problems

- projected subgradient method
- projected subgradient for dual
- subgradient method for constrained optimization

Projected subgradient method

solves constrained optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathcal{C}, \end{array}$$

where $f : \mathbf{R}^n \to \mathbf{R}$, $\mathcal{C} \subseteq \mathbf{R}^n$ are convex

projected subgradient method is given by

$$x^{(k+1)} = P(x^{(k)} - \alpha_k g^{(k)}),$$

P is (Euclidean) projection on $\mathcal{C}$, and $g^{(k)} \in \partial f(x^{(k)})$

same convergence results:

- for constant step size, converges to neighborhood of optimal (for f differentiable and h small enough, converges)

- for diminishing nonsummable step sizes, converges

key idea: projection does not increase distance to $x^\star$

Linear equality constraints

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax = b \end{array}$$

projection of z onto $\{x \mid Ax = b\}$ is

$$\begin{aligned} P(z) &= z - A^T(AA^T)^{-1}(Az - b) \\ &= (I - A^T(AA^T)^{-1}A)z + A^T(AA^T)^{-1}b \end{aligned}$$

projected subgradient update is (using $Ax^{(k)} = b$)

$$\begin{aligned} x^{(k+1)} &= P(x^{(k)} - \alpha_k g^{(k)}) \\ &= x^{(k)} - \alpha_k(I - A^T(AA^T)^{-1}A)g^{(k)} \\ &= x^{(k)} - \alpha_k P_{\mathcal{N}(A)}(g^{(k)}) \end{aligned}$$

Example: Least l_1-norm

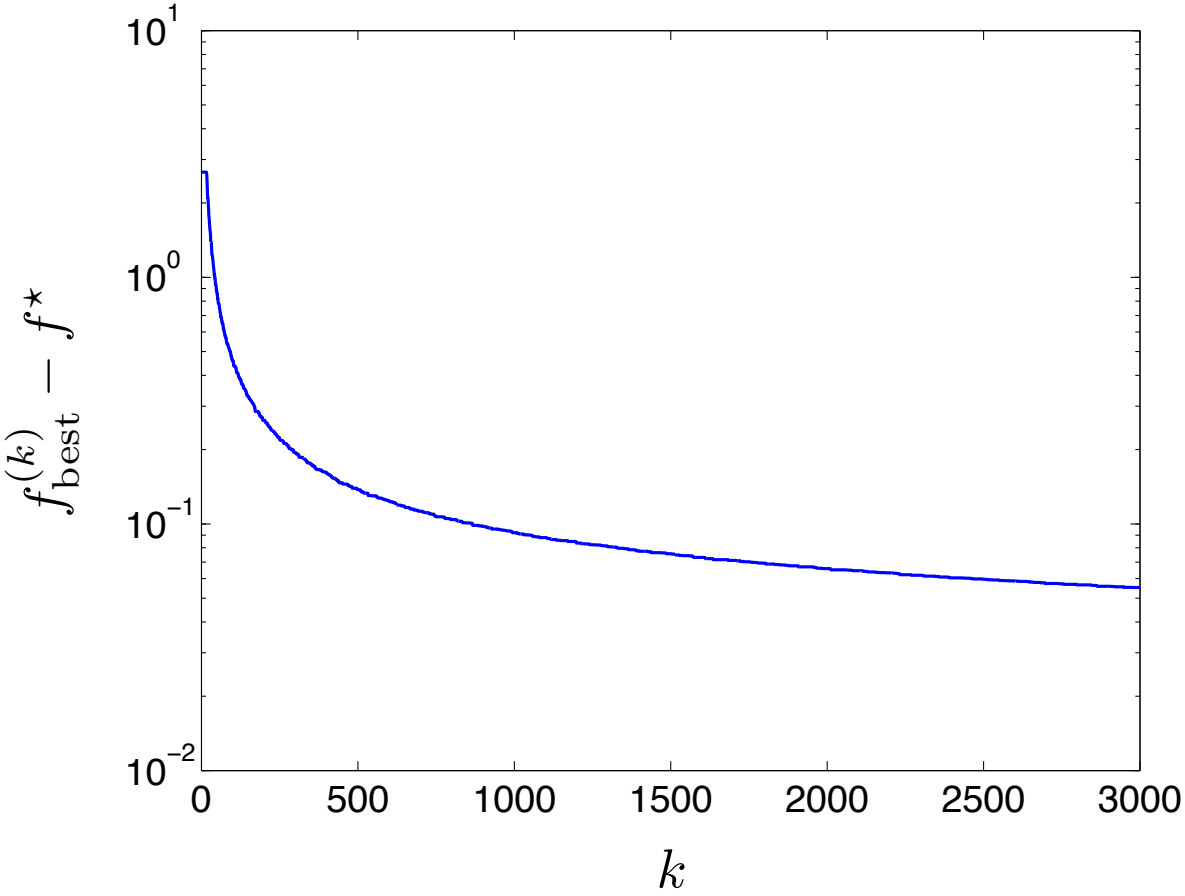
$$\begin{array}{ll} \text{minimize} & \|x\|_1 \\ \text{subject to} & Ax = b \end{array}$$

subgradient of objective is $g = \mathbf{sign}(x)$

projected subgradient update is

$$x^{(k+1)} = x^{(k)} - \alpha_k (I - A^T (AA^T)^{-1} A)\, \mathbf{sign}(x^{(k)})$$

problem instance with $n = 1000$, $m = 50$, step size $\alpha_k = 0.1/k$, $f^\star \approx 3.2$



Projected subgradient for dual problem

(convex) primal:

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \end{array}$$

solve dual problem

$$\begin{array}{ll} \text{maximize} & g(\lambda) \\ \text{subject to} & \lambda \succeq 0 \end{array}$$

via projected subgradient method:

$$\lambda^{(k+1)} = \left(\lambda^{(k)} - \alpha_k h\right)_+, \qquad h \in \partial(-g)(\lambda^{(k)})$$

Subgradient of negative dual function

assume f_0 is strictly convex, and denote, for $\lambda \succeq 0$,

$$x^*(\lambda) = \operatorname*{argmin}_z \left(f_0(z) + \lambda_1 f_1(z) + \cdots + \lambda_m f_m(z)\right)$$

so $g(\lambda) = f_0(x^*(\lambda)) + \lambda_1 f_1(x^*(\lambda)) + \cdots + \lambda_m f_m(x^*(\lambda))$

a subgradient of $-g$ at λ is given by $h_i = -f_i(x^*(\lambda))$

projected subgradient method for dual:

$$x^{(k)} = x^*(\lambda^{(k)}), \qquad \lambda_i^{(k+1)} = \left(\lambda_i^{(k)} + \alpha_k f_i(x^{(k)})\right)_+$$

- primal iterates $x^{(k)}$ are not feasible, but become feasible in limit (sometimes can find feasible, suboptimal $\tilde{x}^{(k)}$ from $x^{(k)}$)
- dual function values $g(\lambda^{(k)})$ converge to $f^\star = f_0(x^\star)$

interpretation:

- λ_i is price for 'resource' $f_i(x)$
- price update $\lambda_i^{(k+1)} = \left(\lambda_i^{(k)} + \alpha_k f_i(x^{(k)})\right)_+$
  - increase price λ_i if resource i is over-utilized ($i.e.$, $f_i(x) > 0$)
  - decrease price λ_i if resource i is under-utilized ($i.e.$, $f_i(x) < 0$)
  - but never let prices get negative

Example

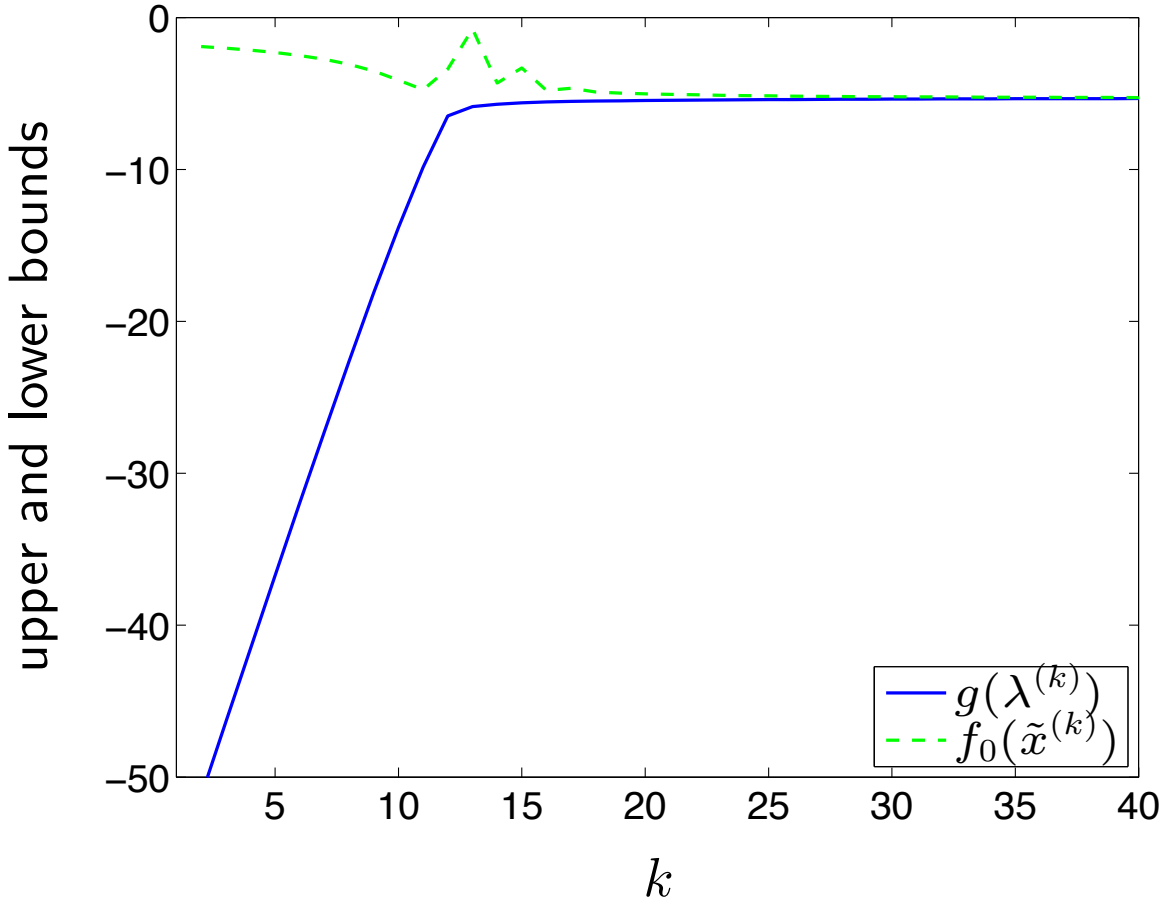
minimize strictly convex quadratic ($P \succ 0$) over unit box:

$$\begin{array}{ll} \text{minimize} & (1/2)x^T P x - q^T x \\ \text{subject to} & x_i^2 \leq 1, \quad i = 1, \ldots, n \end{array}$$

- $L(x, \lambda) = (1/2)x^T(P + \mathbf{diag}(2\lambda))x - q^T x - \mathbf{1}^T\lambda$
- $x^*(\lambda) = (P + \mathbf{diag}(2\lambda))^{-1}q$
- projected subgradient for dual:

$$x^{(k)} = (P + \mathbf{diag}(2\lambda^{(k)}))^{-1}q, \quad \lambda_i^{(k+1)} = \left(\lambda_i^{(k)} + \alpha_k((x_i^{(k)})^2 - 1)\right)_+$$

problem instance with $n = 50$, fixed step size $\alpha = 0.1$, $f^\star \approx -5.3$; $\tilde{x}^{(k)}$ is a nearby feasible point for $x^{(k)}$



Subgradient method for constrained optimization

solves constrained optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m, \end{array}$$

where $f_i : \mathbf{R}^n \to \mathbf{R}$ are convex

same update $x^{(k+1)} = x^{(k)} - \alpha_k g^{(k)}$, but we have

$$g^{(k)} \in \begin{cases} \partial f_0(x) & f_i(x) \leq 0, \quad i = 1, \ldots, m, \\ \partial f_j(x) & f_j(x) > 0 \end{cases}$$

define $f_{\text{best}}^{(k)} = \min\{f_0(x^{(i)}) \mid x^{(i)} \text{ feasible}, \ i = 1, \ldots, k\}$

Convergence

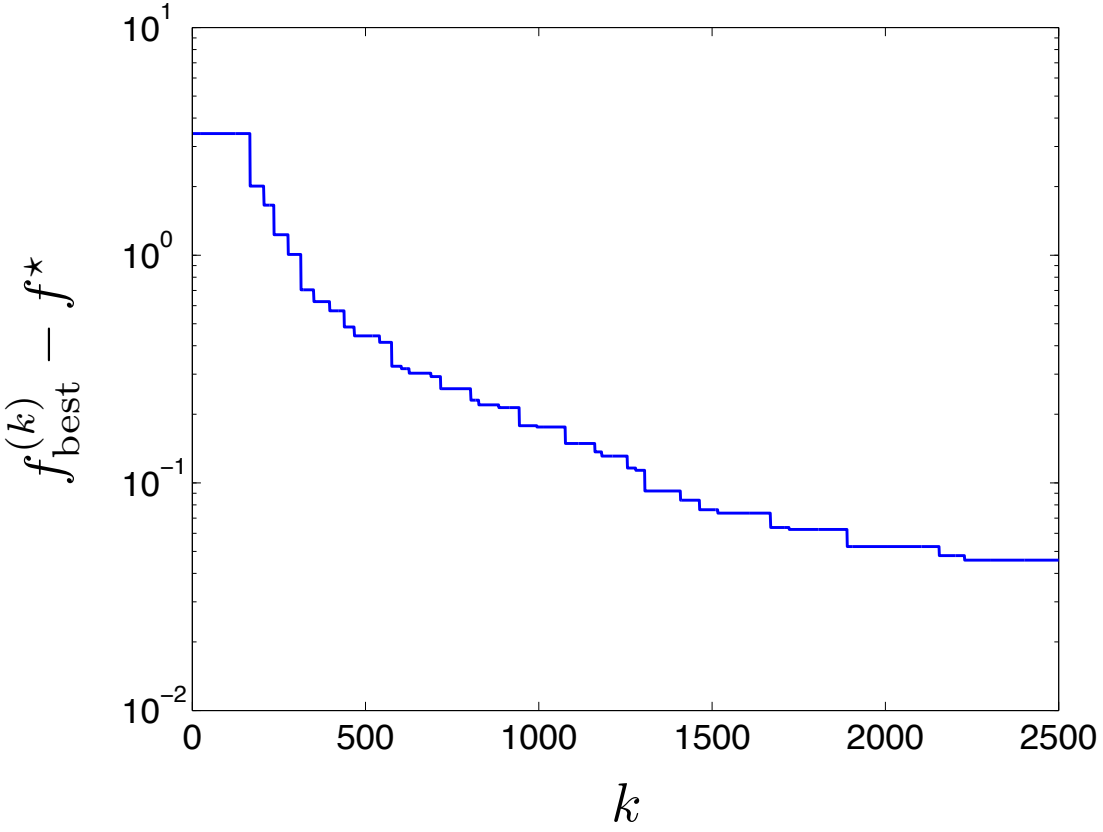
assumptions:

- there exists an optimal $x^\star$; Slater's condition holds
- $\|g^{(k)}\|_2 \leq G$; $\|x^{(1)} - x^\star\|_2 \leq R$

typical result: for $\alpha_k > 0$, $\alpha_k \to 0$, $\sum_{i=1}^{\infty} \alpha_i = \infty$, we have $f_{\text{best}}^{(k)} \to f^\star$

Example: Inequality form LP

LP with $n = 20$ variables, $m = 200$ inequalities, $f^\star \approx -3.4$;
$\alpha_k = 1/k$ for optimality step, Polyak's step size for feasibility step



Decomposition Methods

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

Separable problem

$$\begin{array}{ll} \text{minimize} & f_1(x_1) + f_2(x_2) \\ \text{subject to} & x_1 \in \mathcal{C}_1, \quad x_2 \in \mathcal{C}_2 \end{array}$$

- 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 Ψ nondecreasing ($e.g.$, $\max$)

Complicating variable

consider unconstrained problem,

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

$x = (x_1, x_2, y)$

- y is the **complicating variable** or **coupling variable**; when it is fixed the problem is separable in x_1 and x_2

- x_1, x_2 are **private** or **local** variables; y is a **public** or **interface** or **boundary** variable between the two subproblems

Primal decomposition

fix y and define

$$\begin{array}{lll} \text{subproblem 1:} & \text{minimize}_{x_1} & f_1(x_1, y) \\ \text{subproblem 2:} & \text{minimize}_{x_2} & f_2(x_2, y) \end{array}$$

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

original problem is equivalent to **master problem**

$$\text{minimize}_y \quad \phi_1(y) + \phi_2(y)$$

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 ϕ_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)

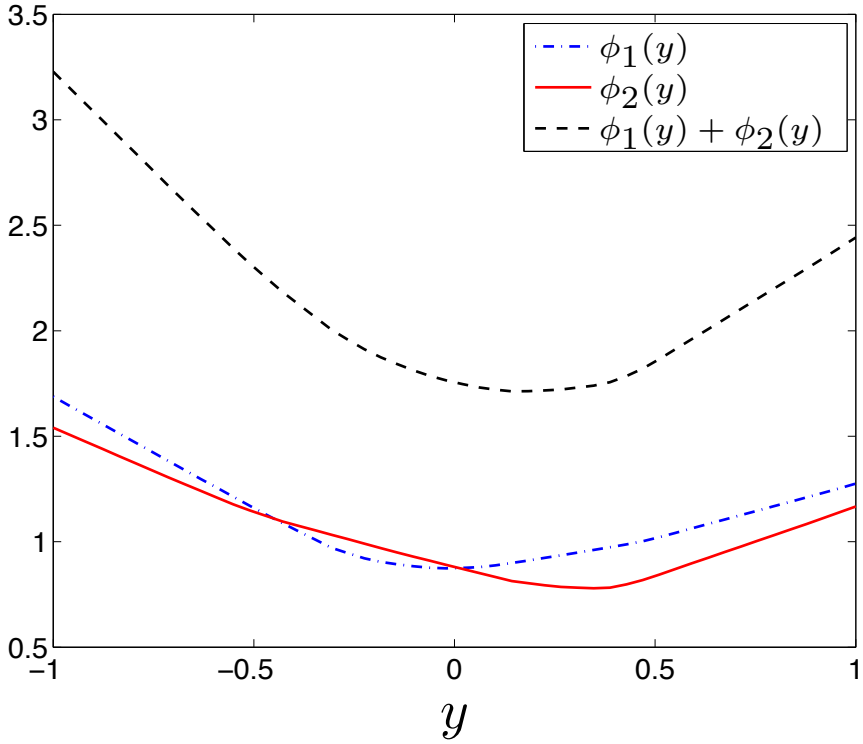
repeat

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.
   $y := y - \alpha_k(g_1 + g_2)$.

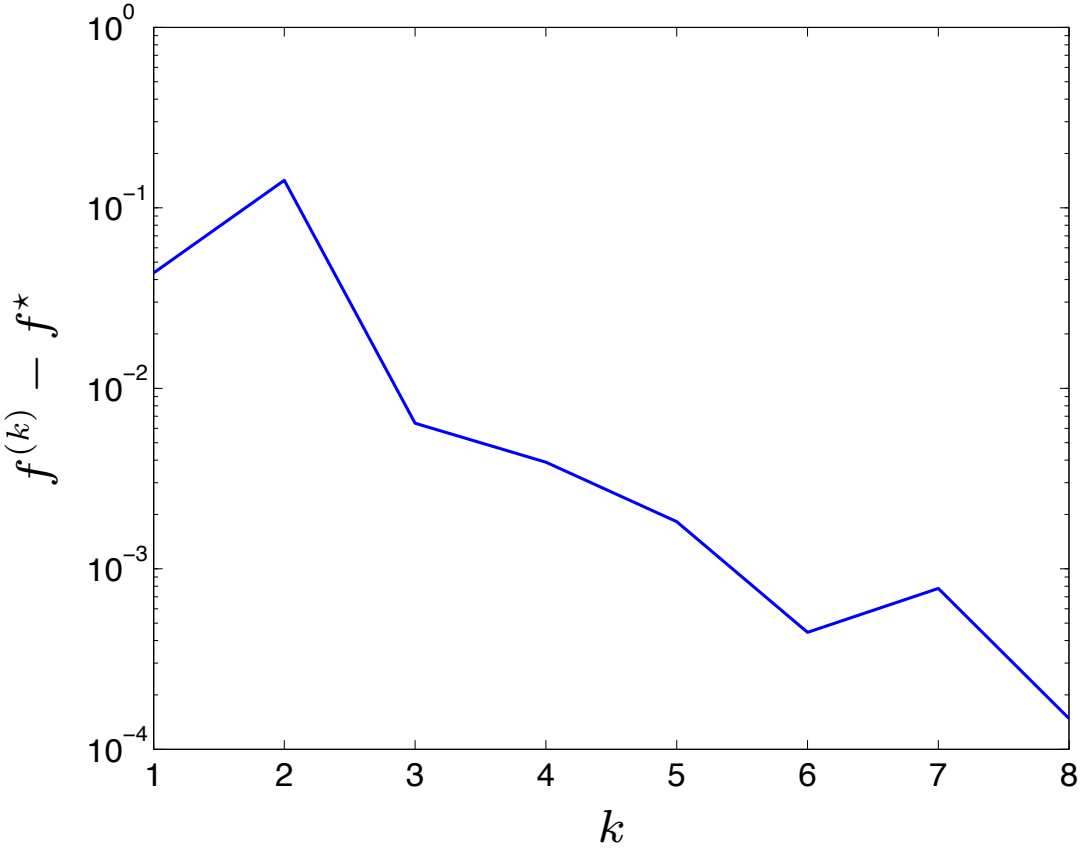
step length α_k can be chosen in any of the standard ways

Example

- $x_1, x_2 \in \mathbf{R}^{20}$, $y \in \mathbf{R}$
- f_i are PWL (max of 100 affine functions each); $f^\star \approx 1.71$



primal decomposition, using bisection on y



Dual decomposition

Step 1: introduce new variables y_1, y_2

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

- y_1, y_2 are **local** versions of complicating variable y
- $y_1 = y_2$ is **consensus 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

$$\begin{aligned} g_1(\nu) &= \inf_{x_1, y_1} \left(f_1(x_1, y_1) + \nu^T y_1\right) = -f_1^*(0, -\nu) \\ g_2(\nu) &= \inf_{x_2, y_2} \left(f_2(x_2, y_2) - \nu^T y_2\right) = -f_2^*(0, \nu) \end{aligned}$$

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 α_k can be chosen in standard ways
- at each step we have a lower bound $g(\nu)$ on $p^\star$
- 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):

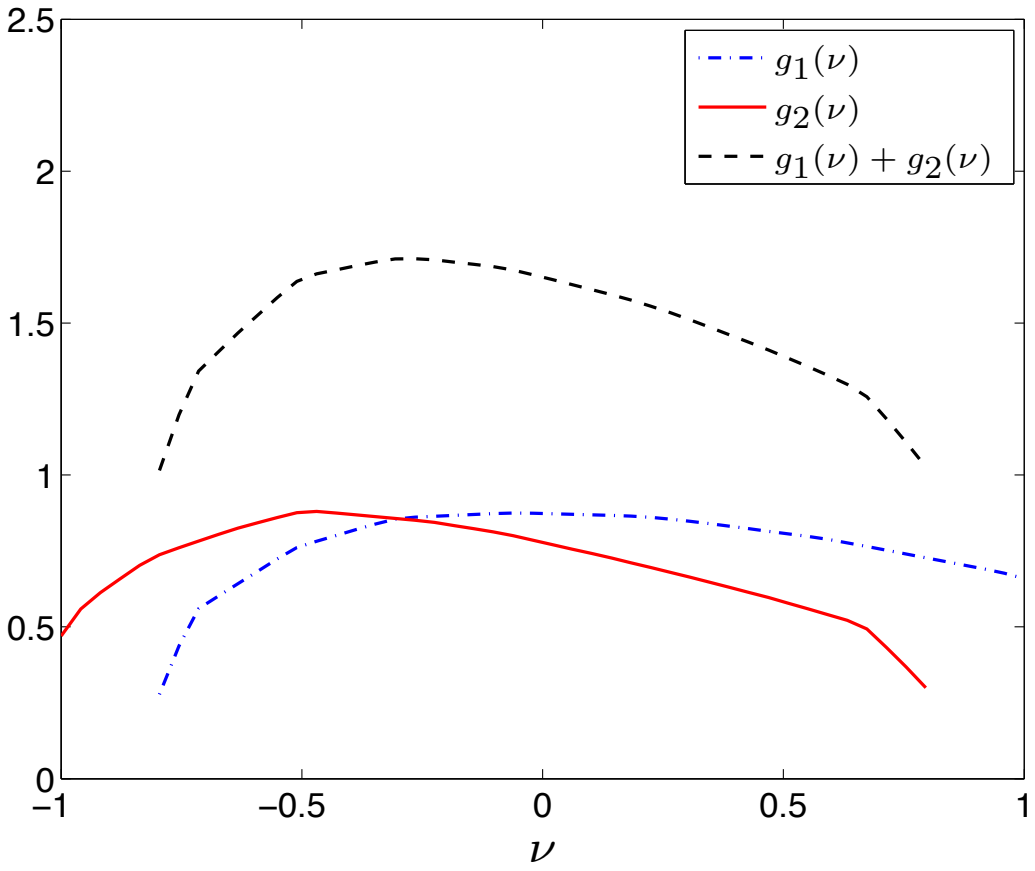
$$(x_1, \bar{y}), \qquad (x_2, \bar{y}), \qquad \bar{y} = (y_1 + y_2)/2$$

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

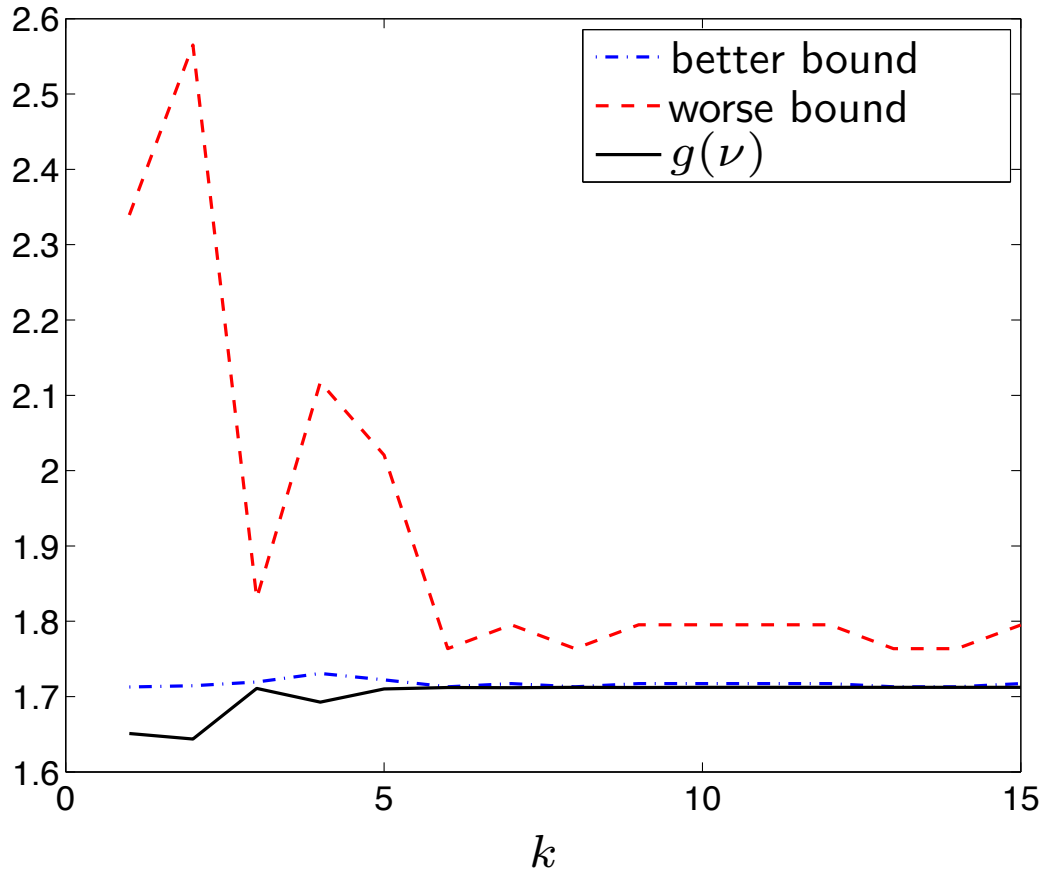
- a better feasible point: replace y_1, y_2 with $\bar{y}$ and solve *primal* subproblems minimize$_{x_1} f_1(x_1, \bar{y})$, minimize$_{x_2} f_2(x_2, \bar{y})$

  - gives (better) upper bound $p^\star \leq \phi_1(\bar{y}) + \phi_2(\bar{y})$

(Same) example



dual decomposition convergence (using bisection on ν)



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
- ν 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:

$$\nu^{(k)}, \qquad (x_1^{(k)}, y_1^{(k)}), \qquad (x_2^{(k)}, y_2^{(k)})$$

  - $x_1^{(k)}, y_1^{(k)}$ 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

- $\nu^{(k)} \to \nu^\star$ ($i.e.$, 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 ϕ_i), or add regularization terms $\epsilon \|x_i\|^2$

Decomposition with constraints

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

$$\begin{array}{ll} \text{minimize} & f_1(x_1) + f_2(x_2) \\ \text{subject to} & x_1 \in \mathcal{C}_1, \quad x_2 \in \mathcal{C}_2 \\ & h_1(x_1) + h_2(x_2) \preceq 0 \end{array}$$

- f_i, h_i, $\mathcal{C}_i$ convex
- $h_1(x_1) + h_2(x_2) \preceq 0$ 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 $t \in \mathbf{R}^p$ and define

subproblem 1:

$$\begin{array}{ll} \text{minimize} & f_1(x_1) \\ \text{subject to} & x_1 \in \mathcal{C}_1, \quad h_1(x_1) \preceq t \end{array}$$

subproblem 2:

$$\begin{array}{ll} \text{minimize} & f_2(x_2) \\ \text{subject to} & x_2 \in \mathcal{C}_2, \quad h_2(x_2) \preceq -t \end{array}$$

- 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

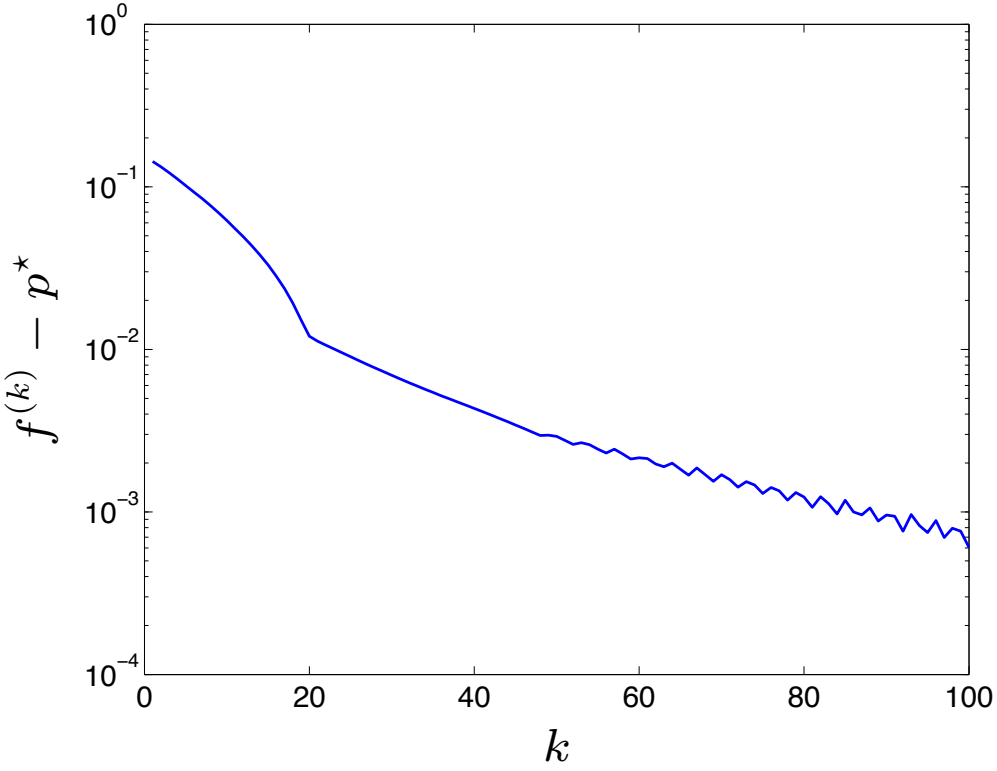
repeat

1. Solve the subproblems.
   Solve subproblem 1, finding x_1 and λ_1.
   Solve subproblem 2, finding x_2 and λ_2.
2. Update resource allocation.
   $t := t - \alpha_k(\lambda_2 - \lambda_1)$.

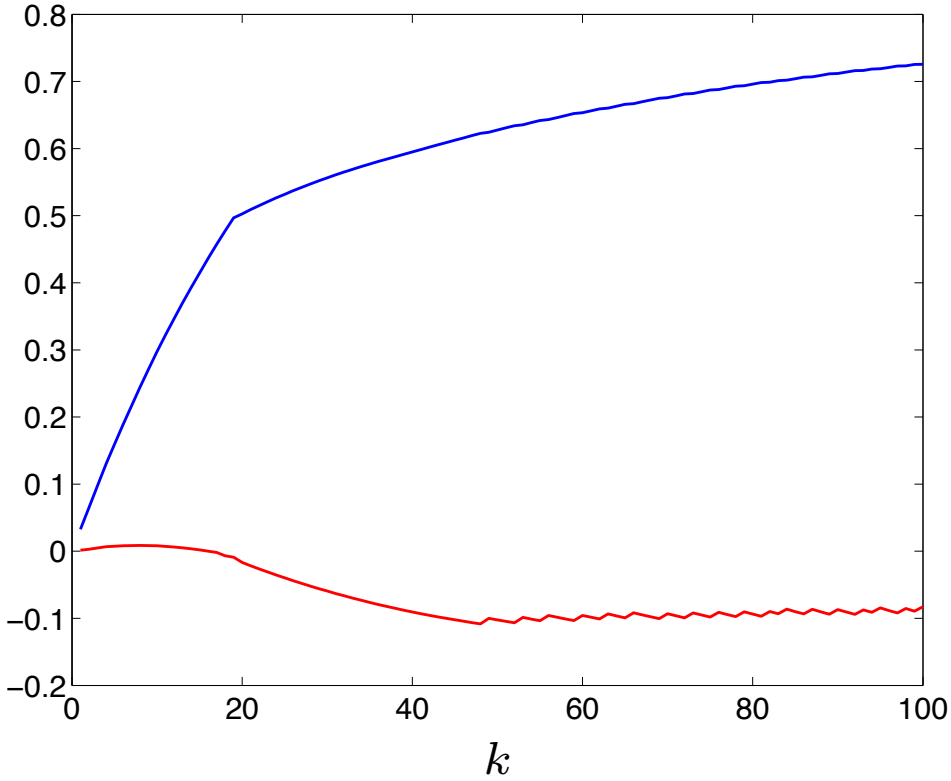
- λ_i is an optimal Lagrange multiplier associated with resource constraint in subproblem i
- $\lambda_2 - \lambda_1 \in \partial(\phi_1 + \phi_2)(t)$
- α_k is an appropriate step size
- all iterates are feasible (when subproblems are feasible)

Example

- $x_1, x_2 \in \mathbf{R}^{20}$, $t \in \mathbf{R}^2$; f_i are quadratic, h_i are affine, $\mathcal{C}_i$ are polyhedra defined by 100 inequalities; $p^\star \approx -1.33$; $\alpha_k = 0.5/k$



resource allocation t to first subsystem (second subsystem gets $-t$)



Dual decomposition

form (separable) partial Lagrangian

$$\begin{aligned} L(x_1, x_2, \lambda) &= f_1(x_1) + f_2(x_2) + \lambda^T(h_1(x_1) + h_2(x_2)) \\ &= \big(f_1(x_1) + \lambda^T h_1(x_1)\big) + \big(f_2(x_2) + \lambda^T h_2(x_2)\big) \end{aligned}$$

fix dual variable λ and define

subproblem 1:

$$\begin{array}{ll} \text{minimize} & f_1(x_1) + \lambda^T h_1(x_1) \\ \text{subject to} & x_1 \in \mathcal{C}_1 \end{array}$$

subproblem 2:

$$\begin{array}{ll} \text{minimize} & f_2(x_2) + \lambda^T h_2(x_2) \\ \text{subject to} & x_2 \in \mathcal{C}_2 \end{array}$$

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 λ 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).
   $\lambda := (\lambda + \alpha_k(h_1(\bar{x}_1) + h_2(\bar{x}_2)))_+$.

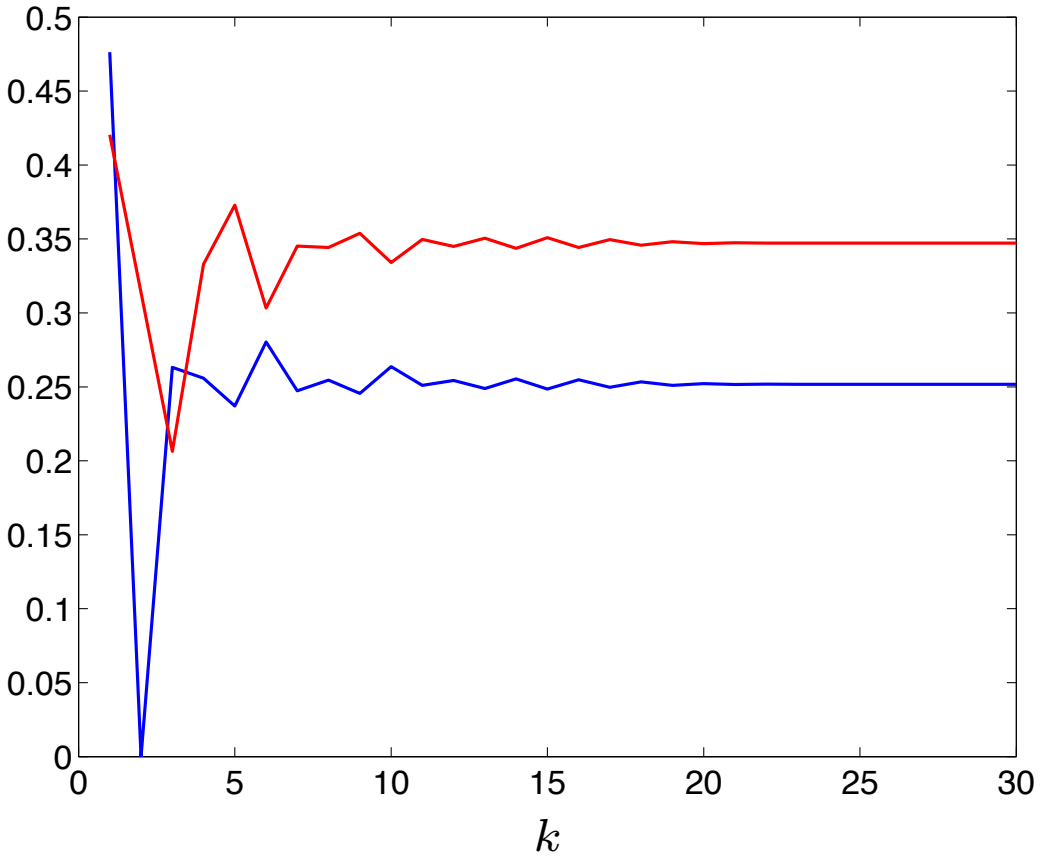
- α_k is an appropriate step size
- iterates need not be feasible
- can again construct feasible primal variables using projection

Interpretation

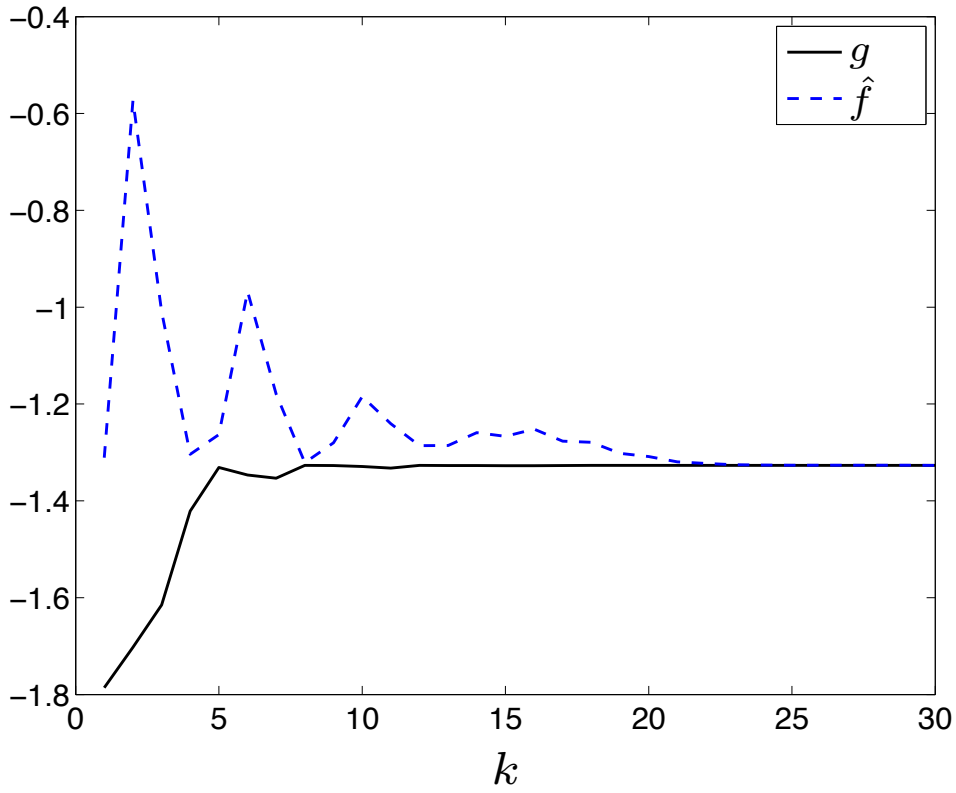
- λ gives prices of resources
- subproblems are solved separately, taking income/expense from resource usage into account
- master algorithm adjusts prices
- prices on over-subscribed resources are increased; prices on undersubscribed resources are reduced, but never made negative

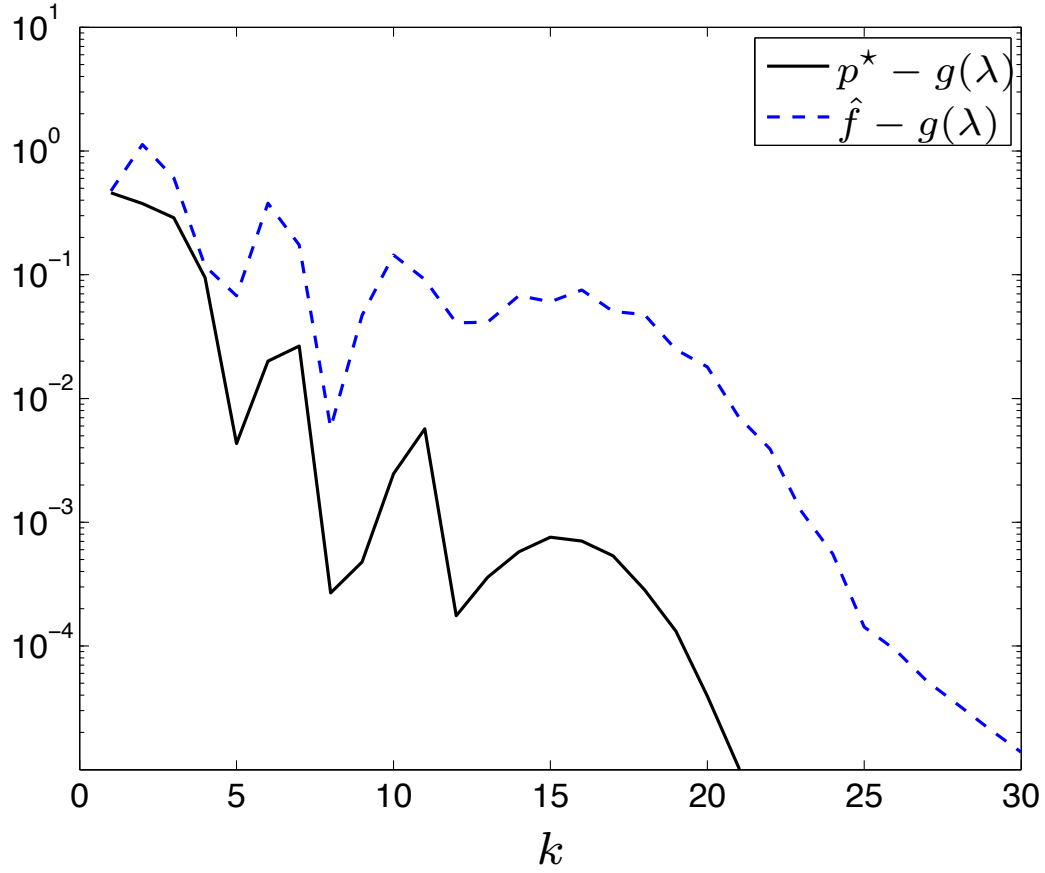
(Same) example

subgradient method for master; resource prices λ



dual decomposition convergence; $\hat{f}$ is objective of projected feasible allocation

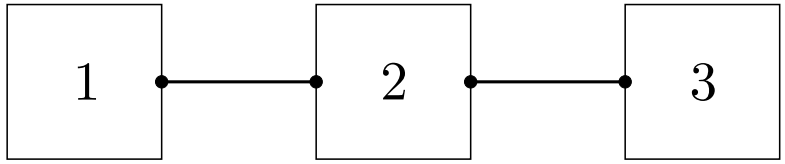



10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
0
5
10
15
20
25
30
k
$p^\star - g(\lambda)$
$\hat{f} - g(\lambda)$

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

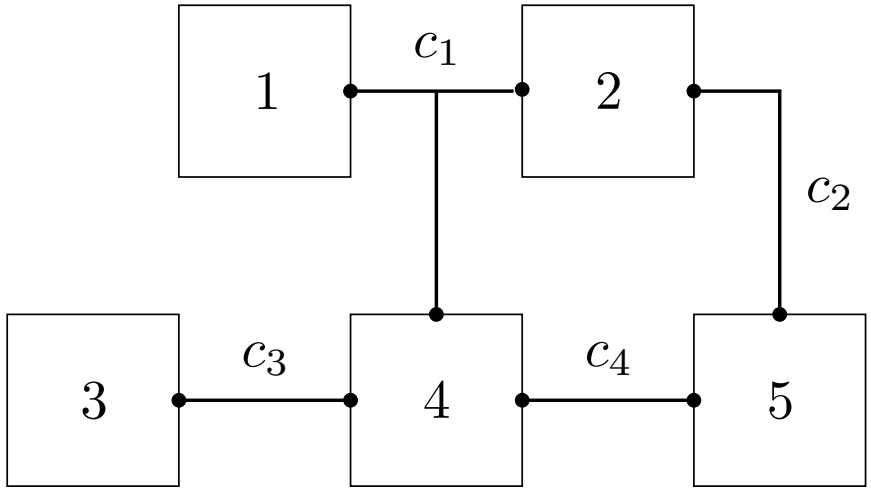
Simple example



- 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

$$\begin{array}{ll} \text{minimize} & f_1(x_1, y_1) + f_2(x_2, y_2, y_3) + f_3(x_3, y_4) \\ \text{subject to} & (x_1, y_1) \in \mathcal{C}_1, \quad (x_2, y_2, y_3) \in \mathcal{C}_2, \quad (x_3, y_4) \in \mathcal{C}_3 \\ & y_1 = y_2, \quad y_3 = y_4 \end{array}$$

A more complex example



General form

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{K} f_i(x_i, y_i) \\ \text{subject to} & (x_i, y_i) \in \mathcal{C}_i, \quad i = 1, \ldots, K \\ & y_i = E_i z, \quad i = 1, \ldots, K \end{array}$$

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

Primal decomposition

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

$$\begin{array}{ll} \text{minimize} & f_i(x_i, y_i) \\ \text{subject to} & (x_i, y_i) \in \mathcal{C}_i \end{array}$$

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

$$\begin{array}{ll} \text{minimize} & f_i(x_i, y_i) + \nu_i^T y_i \\ \text{subject to} & (x_i, y_i) \in \mathcal{C}_i \end{array}$$

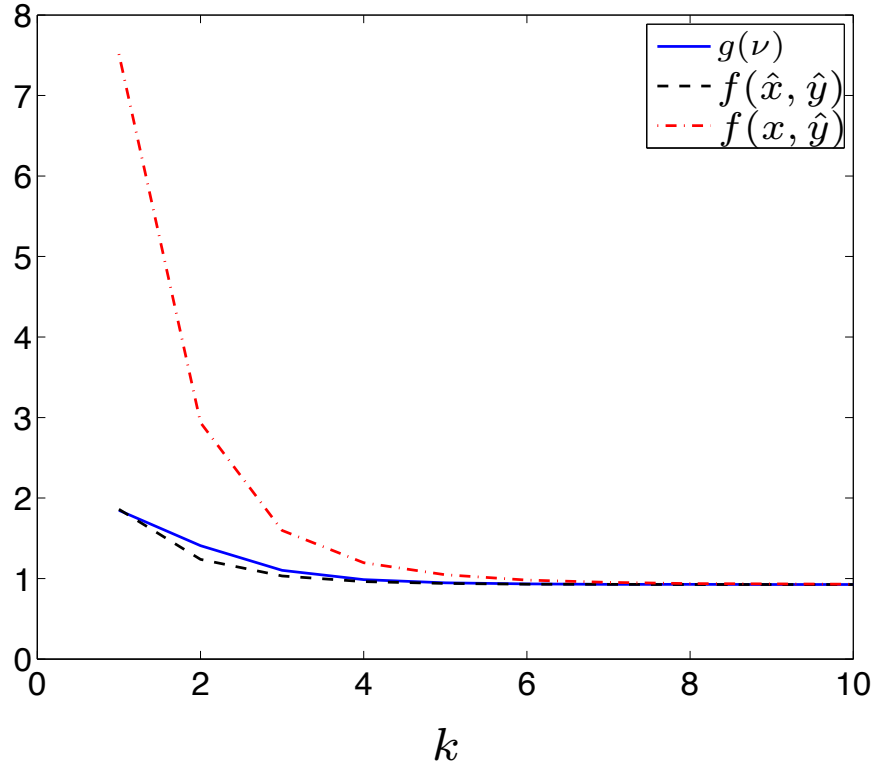
given initial price vector ν that satisfies $E^T\nu = 0$ ($e.g.$, $\nu = 0$).

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; $p^\star \approx 11.1$; $\alpha = 0.5$



consistency constraint residual $\|y - E\hat{z}\|$ versus iteration number