

Distributed Optimization: Analysis and Synthesis via Circuits

Stephen Boyd

Prof. S. Boyd, EE364b, Stanford University

Outline

- canonical form for distributed convex optimization
- circuit interpretation
- primal decomposition
- dual decomposition
- prox decomposition
- momentum terms

Distributed convex optimization problem

- convex optimization problem partitioned into coupled subsystems
- divide variables, constraints, objective terms into two groups
 - **local** variables, constraints, objective terms appear in only one subsystem
 - **complicating** variables, constraints, objective terms appear in more than one subsystem
- describe by hypergraph
 - subsystems are nodes
 - complicating variables, constraints, objective terms are hyperedges

Conditional separability

- **separable problem**: can solve by solving subsystems separately, *e.g.*,

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

- in distributed problem, two subsystems are **conditionally separable** if they are separable when all other variables are fixed
- two subsystems not connected by a net are **conditionally separable**
- cf. conditional independence in Bayes net: two variables not connected by hyperedge are conditionally independent, given all other variables

Examples

- minimize $f_1(z_1, x) + f_2(z_2, x)$, with variables z_1, z_2, x
 - x is the **complicating variable**; when fixed, problem is separable
 - z_1, z_2 are **private** or **local** variables
 - x is a **public** or **interface** or **boundary** variable between the two subproblems
 - hypergraph: two nodes connected by an edge
- optimal control problem
 - state is the complicating variable between past and future
 - hypergraph: simple chain

Transformation to standard form

- introduce slack variables for complicating inequality constraints
- introduce local copies of complicating variables
- implicitly minimize over private variables (preserves convexity)
- represent local constraints in domain of objective term
- we are left with
 - all variables are public, associated with a single node
 - all constraints are **consistency constraints**, *i.e.*, equality of two or more variables

Example

- minimize $f_1(z_1, x) + f_2(z_2, x)$, with variables z_1, z_2, x
- introduce local copies of complicating variable:

$$\begin{array}{ll}\text{minimize} & f_1(z_1, x_1) + f_2(z_2, x_2) \\ \text{subject to} & x_1 = x_2\end{array}$$

- eliminate local variables:

$$\begin{array}{ll}\text{minimize} & \tilde{f}_1(x_1) + \tilde{f}_2(x_2) \\ \text{subject to} & x_1 = x_2\end{array}$$

$$\text{with } \tilde{f}_i(x_i) = \inf_{z_i} f_i(z_i, x_i)$$

General form

- n subsystems with variables x_1, \dots, x_n
- m nets with common variable values z_1, \dots, z_m

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^n f_i(x_i) \\ \text{subject to} & x_i = E_i z, \quad i = 1, \dots, n \end{array}$$

- matrices E_i give **netlist** or **hypergraph**
(row k is e_p , where k th entry of x_i is in net p)

Optimality conditions

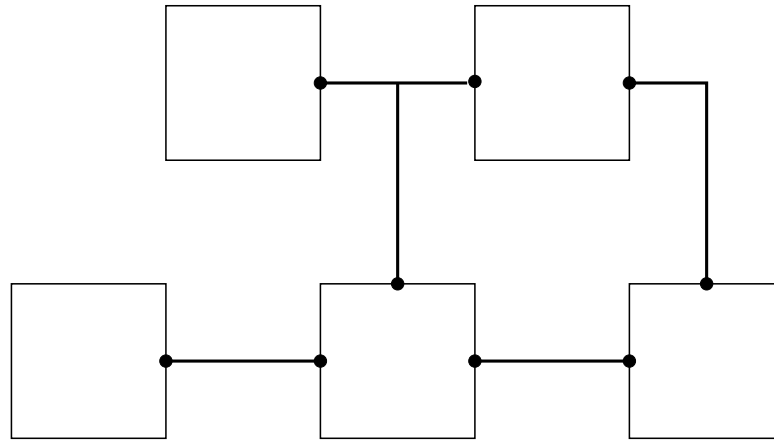
- introduce dual variable y_i associated with $x_i = E_i z$
- optimality conditions are

$$\begin{array}{ll} \nabla f_i(x_i) = y_i & \text{(subsystem relations)} \\ x_i = E_i z & \text{(primal feasibility)} \\ \sum_{i=1}^n E_i^T y_i = 0 & \text{(dual feasibility)} \end{array}$$

(for nondifferentiable case, replace $\nabla f_i(x_i)$ with $g_i \in \partial f_i(x_i)$)

- primal condition: (primal) variables on each net are the same
- dual condition: dual variables on each net sum to zero

Circuit interpretation (primal/voltages)



- subsystems are (grounded) nonlinear resistors
- nets are wires (nets); consistency constraint is KVL
- z_j is voltage on net j
- x_i is vector of pin voltages for resistor i

Circuit interpretation (dual/currents)

- y_i is vector of currents entering resistor i
- dual feasibility is KCL: sum of currents leaving net j is zero
- V-I characteristic for resistor i : $y_i = \nabla f_i(x_i)$
- $f_i(x)$ is **content function** of resistor i
- convexity of f_i is **incremental passivity** of resistor i :

$$(x_i - \tilde{x}_i)^T (y_i - \tilde{y}_i) \geq 0, \quad y_i = \nabla f_i(x_i), \quad \tilde{y}_i = \nabla f_i(\tilde{x}_i)$$

- **optimality conditions are exactly the circuit equations**

Decomposition methods

- solve distributed problem iteratively
 - algorithm state maintained in nets
- each step consists of
 - (parallel) update of subsystem primal and dual variables, based only on adjacent net states
 - update of the net states, based only on adjacent subsystems
- algorithms differ in
 - interface to subsystems
 - state and update

Primal decomposition

repeat

1. distribute net variables to adjacent subsystems

$$x_i := E_i z$$

2. optimize subsystems (separately)

solve subproblems to evaluate $y_i = \nabla f_i(x_i)$

3. collect and sum dual variables for each net

$$w := \sum_{i=1}^n E_i^T y_i$$

4. update net variables

$$z := z - \alpha_k w.$$

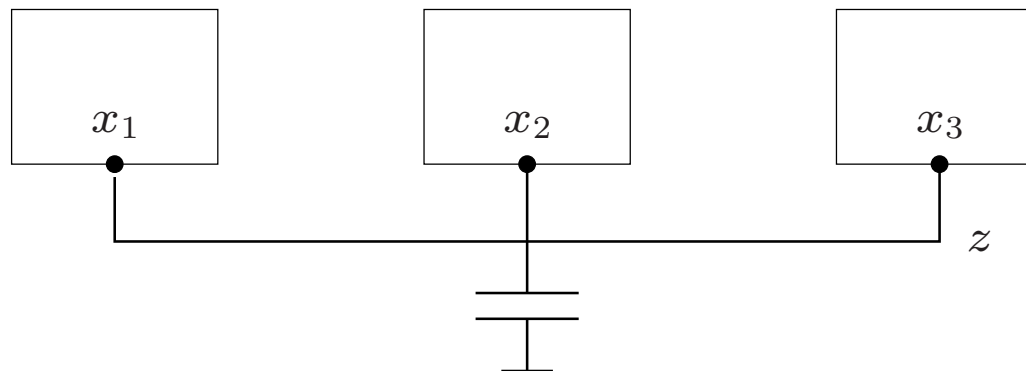
- step factor α_k chosen by standard gradient or subgradient rules

Primal decomposition

- algorithm state is net variable z (net voltages)
- $w = \sum_{i=1}^n E_i^T y_i$ is dual residual (net current residuals)
- primal feasibility maintained; dual feasibility approached in limit
- subsystems are **voltage controlled**:
 - voltage x_i is asserted at subsystem pins
 - pin currents y_i are then found

Circuit interpretation

- connect capacitor to each net; system relaxes to equilibrium
- forward Euler update is primal decomposition
- incremental passivity implies convergence to equilibrium



Dual decomposition

initialize y_i so that $\sum_{i=1}^n E_i^T y_i = 0$
(dual variables sum to zero on each net)

repeat

1. optimize subsystems (separately)

find x_i with $\nabla f_i(x_i) = y_i$, *i.e.*, minimize $f_i(x_i) - y_i^T x_i$

2. collect and average primal variables over each net

$$z := (E^T E)^{-1} E^T x$$

3. update dual variables

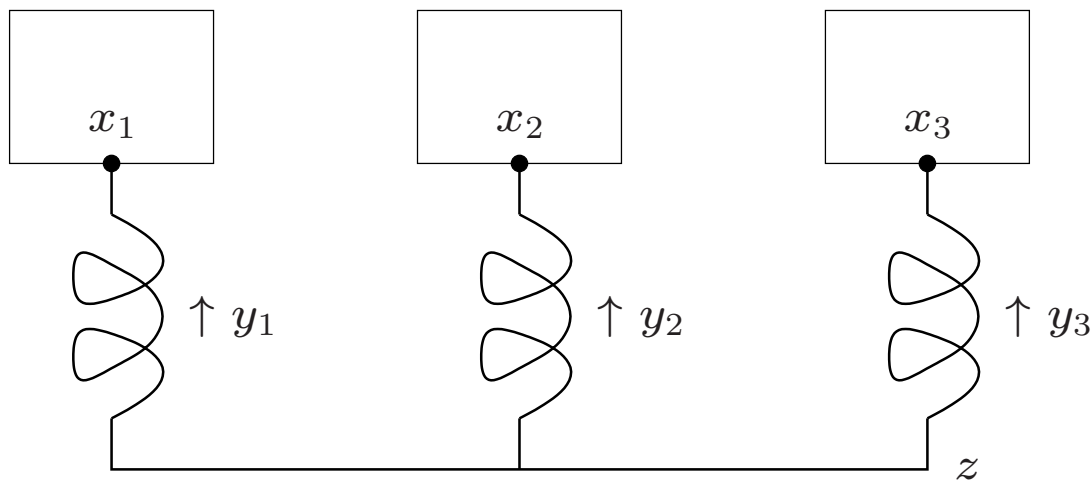
$$y := y - \alpha_k (x - Ez)$$

Dual decomposition

- algorithm state is dual variable y
- $x - Ez$ is consistency residual
- dual feasibility maintained; primal feasibility approached in limit
- subsystems are **current controlled**:
 - pin currents y_i are asserted
 - pin voltages x_i are then found

Circuit interpretation

- connect inductor to each pin; system relaxes to equilibrium
- forward Euler update is dual decomposition
- incremental passivity implies convergence to equilibrium

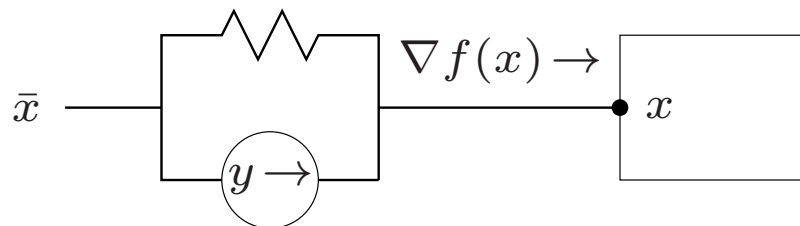


Prox(imal) interface

- prox operator:

$$P_\rho(y, \bar{x}) = \operatorname{argmin}_x \left(f(x) - y^T x + (\rho/2) \|x - \bar{x}\|_2^2 \right)$$

- contains usual dual term $y^T x$ and ‘proximal regularization’ term
- amounts to solving $\nabla f(x) + \rho(x - \bar{x}) = y$
- circuit interpretation: drive via resistance $R = 1/\rho$
cf. voltage (primal) drive or current (dual) drive



Prox decomposition

initialize y_i so that $\sum_{i=1}^n E_i^T y_i = 0$

repeat

1. optimize subsystems (separately)

$$\text{minimize } f_i(x_i) - y_i^T x_i + (\rho/2) \|x_i - E_i z\|^2$$

2. collect and average primal variables over each net

$$z := (E^T E)^{-1} E^T x$$

3. update dual variables

$$y := y - \rho(x - Ez)$$

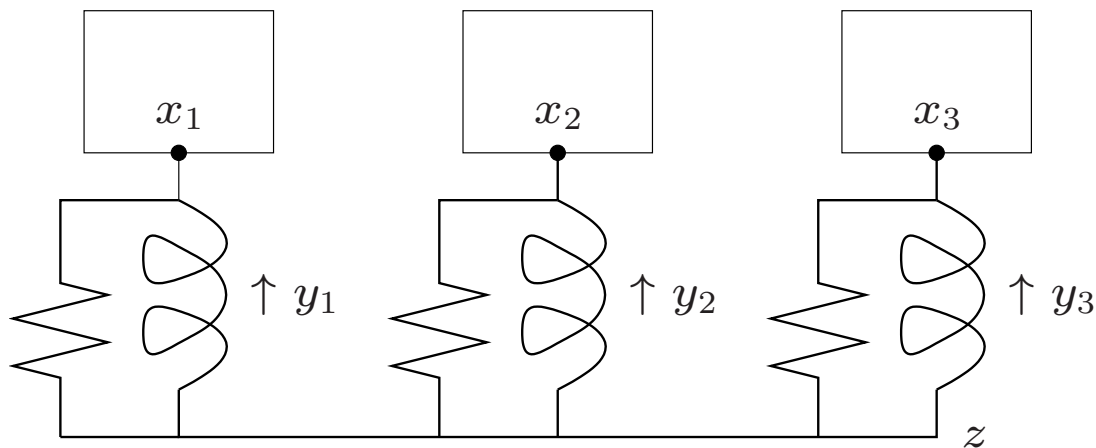
- step size ρ in dual update guaranteed to work

Prox decomposition

- has **many** other names . . .
- algorithm state is dual variable y
- $y - \rho(x - \bar{x})$ is dual feasible
- primal and dual feasibility approached in limit
- subsystems are resistor driven; must support prox interface
- interpretations
 - regularized dual decomposition
 - PI feedback (as opposed to I only feedback)

Circuit interpretation

- connect inductor \parallel resistor to each pin; system relaxes to equilibrium
- forward Euler update is prox decomposition
- incremental passivity implies convergence to equilibrium

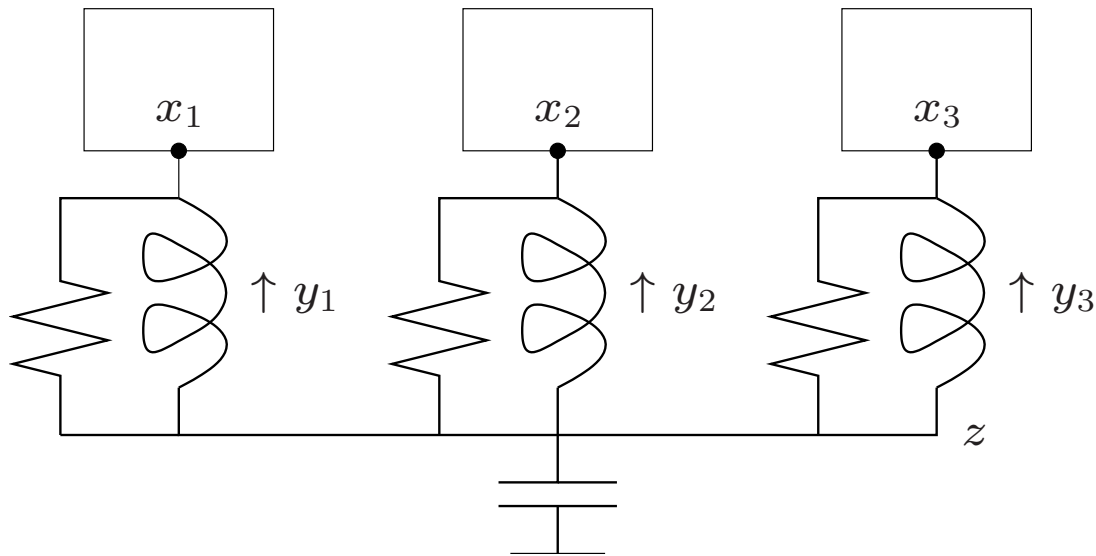


Momentum terms

- in optimization method, current search direction is
 - standard search direction (gradient, subgradient, prox . . .)
 - plus last search direction, scaled
- interpretations/examples
 - smooth/low pass filter/average search directions
 - add momentum to search algorithm ('heavy-ball method')
 - two term method (CG)
 - Nesterov optimal order method
- often dramatically improves convergence

You guessed it

- algorithm: prox decomposition with momentum
- just add capacitor to prox LR circuit



Conclusions

to get a distributed optimization algorithm:

- represent as circuit with interconnecting wires
- replace interconnect wires with passive circuits that reduce to wires at equilibrium
- discretize circuit dynamics
- subsystem interfaces depend on circuit drive (current, voltage, via resistor)
- convergence hinges on incremental passivity