Enriching the art of engineering design via convex optimization

Tim Davidson

McMaster University
Hamilton, ON, Canada
`http://www.ece.mcmaster.ca/~davidson`

EURASIP Seminar
SPAWC 2010

Slides with magenta headings have been added since the seminar.

Engineering Design

- Typically a multi-stage process with two key stages
  - Choose a configuration and identify free parameters
  - Choose values for the free parameters
- Example: digital filter design
  - Choose a configuration:
    FIR or IIR? fixed order? discrete coefficients?
  - Choose values for the filter coefficients
- Parameter choice
  - Typically requires judicious trade-offs, or
    showing no suitable parameters exist for current config.
  - Design experience is often distilled into guidelines
  - This tutorial: enriching process of parameter design by harnessing the perspective of optimization theory; and in particular, that of convex optimization

Goals (and caveats)

- Help you to harness the perspective of optimization to enrich the common sense of good design practice
- This is not an introduction to convex optimization; more a taste of how optimization can be leveraged for design
- Many of you know convexity opens door to reliable algo's
  Emphasis here is on other doors that convexity opens and impact on the design process

- Rigor is important in practice, but I will be sloppy; e.g.,
  - Affine functions $\mathbf{a}^T\mathbf{x} + b$ described as being linear
  - Implicit assumptions of full column rank in linear eq'ns
- Associated literature can fill technical gaps;
  List of 'entry points' at the end

Parameter Optimization

- Given configuration, how to choose free parameters?
- Consider taking a structured approach
  - Identify the design variables: $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n$
  - identify req'd characteristics: $f_m(\mathbf{x}) \leq \xi_m$; $g_q(\mathbf{x}) = \zeta_q$
    Note: $f_m(\mathbf{x}) \geq \xi_m \longleftrightarrow -f_m(\mathbf{x}) \leq -\xi_m$
  - Identify cost function: $f_0(\mathbf{x})$;
    locally decreases with increasing merit
  - Find the best of the satisficing parameter vectors

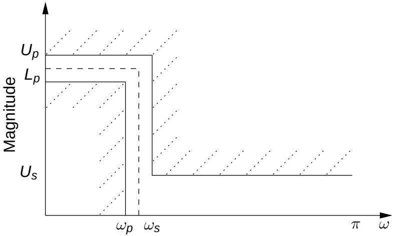
$$\begin{aligned} \min_{\mathbf{x} \in \mathcal{X}} \quad & f_0(\mathbf{x}) \\ \text{subject to} \quad & f_m(\mathbf{x}) \leq \xi_m \\ & g_q(\mathbf{x}) = \zeta_q \end{aligned}$$

    or show that no satisficing parameter vector exists.
    In latter case need to revisit configuration

- This process often enlightening in and of itself

Simple example

FIR low pass filter with L discrete coefficients



- Identify variables:
  - the L filter coefficients; can take on only discrete values
- Identify required characteristics:
  - magnitude response lies within mask
- Identify objective:
  - stop band energy

Formal Optimization

- Write the design problem at hand as

$$\begin{aligned} \min_{\mathbf{x}\in\mathcal{X}} \quad & f_0(\mathbf{x}) \\ \text{subject to} \quad & f_m(\mathbf{x}) \leq \xi_m \\ & g_q(\mathbf{x}) = \zeta_q \end{aligned}$$

  or show that there is no feasible $\mathbf{x}$
- Does this help?
- Maybe not! Problem may be fundamentally difficult
- This tutorial will help you identify how it can help
- and for cases where it initially appears that it does not
  - we will provide some suggestions for things to try, and
  - help you manage expectations of impact on design

Desirable Properties I

- **Model accuracy:**
  - Is global optimum really the best design?
  - Is it even good?

  Knowledge of application is important
- **Reliable solution method:**
  - no tweaking of parameters of algorithm
  - unsupervised; perhaps even embeddable
  - detection of infeasibility
  - easy to program
- **Computational efficiency:**
  Assessment depends on application; might want
  - ‘real time’, or
  - graceful (polynomial) increase with problem size

Desirable Properties II

- **Insight:**
  - Structure of the solution
  - Inherent trade-offs between competing design criteria
- **Robustness/sensitivity of solution:**
  - Extent of neighbouring $\mathbf{x}$'s that are feasible? good?
    - Design: enables secondary objective
    - Estimation: evaluates specificity of criterion
  - Sensitivity of solution to changes in $f_m(\mathbf{x})$ or $g_q(\mathbf{x})$
    - What if these functions are only partially known?

In practice?

- Typically, on your first try, the problem will have few of these desirable properties, if any
- What to do? grid search? random search?
- Key steps in proposed approach
  - Study application and optimization problem to identify an underlying problem with better properties
    - still want reasonable model accuracy, but reliability, comp. efficiency, insight given greater weight
    - this ‘nicer’ problem may have different variables, or even different dimensions
  - Solve the ‘nicer’ problem
  - Use that solution to generate good sol’n to orig. prob. or to obtain insight into the original problem
  - Iterate, if necessary

An alternative approach

- Proposed approach is a “problem first” approach
  - Describe the actual design problem first, then
  - try to approximate with a ‘nice’ optimization problem
- An alternative approach: “optimization first”
  - Consider all the ‘nice’ opt. problems that you know
  - Pick the one that best suits the problem
  - Add on ‘features’ while retaining ‘nicety’

‘Nice’ problems

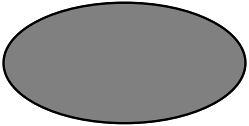
- So which problems are ‘nice’?
- Some that have been known for some time:
  - Least-squares problems: $\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$
    closed-form solution: $\mathbf{x}^\star = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$; $\mathbf{A}$ fcr
  - least-squares with linear equality constraints;
    also closed-form solution
  - problems with linear objective and linear constrs;
    computationally efficient algo’s; optimality conditions
- For much of that time, “approx. by nice problem” meant approx. by one of these, or a few others
- Clearly that could incur large “modelling error”
- Good news: the list of ‘nice’ problems has been substantially expanded over last 15–20 years; an enabling step

Convexity

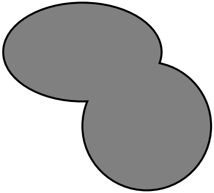
To help be more specific about ‘nice’, let’s look at convexity

Convex set: contains all line segments between any pair of points in the set

Convex

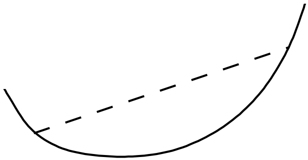


Non-convex



Convex functions

- Convex function: for any two points in the domain, function lies below the line segment joining the functional values



- Epigraph: set $(t, \mathbf{x})$ such that $t \geq f(\mathbf{x})$
- A function is convex iff its epigraph is a convex set
- A function $f(\mathbf{x})$ is concave if $-f(\mathbf{x})$ is convex

Convex problems

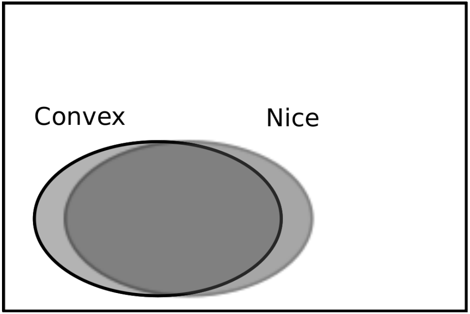
- Recall generic problem

$$
\begin{aligned}
\min_{\mathbf{x}\in\mathcal{X}} \quad & f_0(\mathbf{x}) \\
\text{subject to} \quad & f_m(\mathbf{x}) \leq \xi_m \\
& g_q(\mathbf{x}) = \zeta_q
\end{aligned}
$$

- If $f_0(\mathbf{x})$ and all $f_m(\mathbf{x})$ are convex and all $g_q(\mathbf{x})$ are linear then problem is convex
- Least-squares and linear programs are convex
- For symmetric matrix $\mathbf{Q}$ with non-negative eigenvalues $f(\mathbf{x}) = \mathbf{x}^T\mathbf{Q}\mathbf{x} + 2\mathbf{p}^T\mathbf{x} + r$ is convex
- Note: Maximizing a concave function $\tilde{f}_0(\mathbf{x})$ equiv. to minimizing $-\tilde{f}_0(\mathbf{x})$, which is a convex function

Convex and ‘nice’ problems

A coarse categorization



Convex problems

- Reliable algo. for global optimum; most v. efficient
- Easily implementable general purpose tools that can handle many cases; e.g., CVX
- but there's more than just a good algorithm
- Enable efficient/reliable computation of trade-offs
- Optimality conditions; insight into structure
- Bounds obtained using duality can reliably determine when no suitable set of parameters exists for the current configuration
- Also, Lagrange multipliers may give some insight into how to modify configuration
- Sometimes convexity is obscured,
  but when discovered, it is well worth it

Non-convex problems

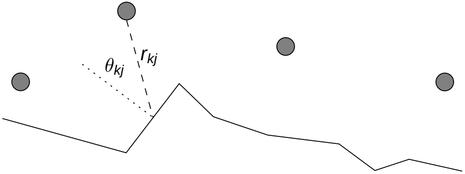
- Often “what you see is all you’ve got” (apologies to Brians Reid and Kernighan, and Leslie Lamport)
- In a few cases careful analysis yields specialized algorithms that have desirable features
- For smooth problems, reasonable general purpose software, e.g., fmincon (matlab), lancelot.
  Driving force: sequence of local convex approximations
- However, typically, for anything other than truly small problems all we can expect to do in a reasonable amount of time is find a locally optimal solution
- Therefore, even when you can’t find a sol’n, hard to decide if problem is infeasible for current config.
- Insight from convex approx’s can sometimes
  - help you understand some features of problem
  - guide you to good local solutions
  - help you evaluate local solutions

An example (Boyd)

- Your consulting company gets a call from a ski operator
- They installed light towers for night skiing
- Customers complaining about illumination; insufficient and uneven
- They ask: Do we need to move towers or install more?
- What do you tell them?

An example (Boyd)

- Construct a model: n small flat patches



- Configuration: m towers in fixed positions
- Free parameters: power used in each lamp, $p_j \in [0, p_{\max}]$
- Quantity of interest: Intensity on each patch
  - Easy. Free space propagation: $I_k(\mathbf{p}) = \sum_{j=1}^{m} a_{kj} p_j$, $a_{kj} = \frac{1}{r_{kj}^2}\cos(\theta_{kj})$? No! $a_{kj} = \max\left\{\frac{1}{r_{kj}^2}\cos(\theta_{kj}), 0\right\}$
  - Note that $I_k(\mathbf{p})$ is linear in $\mathbf{p}$
- Obj: Make intensity on each surface close to I_{des}
- Let's begin with the "optimization first" approach

"Optimization first" approach

- Try uniform allocation $p_j = p$, and
  try to find a better p than currently used
- Try least squares

  $$\min_{\mathbf{p}\in\mathbb{R}^m} \sum_{k=1}^{n} \left(I_k(\mathbf{p}) - I_{\text{des}}\right)^2$$

  closed-form solution; round solution to $[0, p_{\max}]$
- Try regularizing

  $$\min_{\mathbf{p}\in\mathbb{R}^m} \sum_{k=1}^{n} \left(I_k(\mathbf{p}) - I_{\text{des}}\right)^2 + \sum_{j=1}^{m} w_j \left(p_j - p_{\max}/2\right)^2$$

  closed-form; iteratively adjust w_j until opt. $p_j \in [0, p_{\max}]$
- Try linear programming

  $$\begin{array}{rll} \min_{\mathbf{p}\in\mathbb{R}^m,\,\delta\in\mathbb{R}} & \delta & \\ \text{subject to} & -\delta \le I_k(\mathbf{p}) - I_{\text{des}} \le \delta & k = 1, 2, \ldots, n \\ & 0 \le p_j \le p_{\max} & j = 1, 2, \ldots, m \end{array}$$

  convex; efficiently solvable; no tweaking

“Problem first” approach

- Response of eye to intensity is approx. logarithmic
- Suggests:

$$\begin{array}{ll} \min_{\mathbf{p}\in\mathbb{R}^m} & \max_{k\in[1,n]} \left|\log\big(I_k(\mathbf{p})\big) - \log\big(I_{\text{des}}\big)\right| \\ \text{subject to} & 0 \le p_j \le p_{\max} \quad j = 1, 2, \ldots, m \end{array}$$

- Looks intimidating
- Analyze: $\left|\log(a) - \log(b)\right| \le \tau \longleftrightarrow \max\{\frac{a}{b}, \frac{b}{a}\} \le e^\tau$
- Equivalent problem:

$$\begin{array}{ll} \min_{\mathbf{p}\in\mathbb{R}^m} & \max_{k\in[1,n]} h\big(I_k(\mathbf{p})/I_{\text{des}}\big) \\ \text{subject to} & 0 \le p_j \le p_{\max} \quad j = 1, 2, \ldots, m \end{array}$$

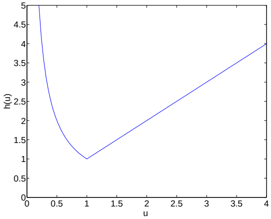
where $h(u) = \max\{u, 1/u\}$.

Problem first approach

Equivalent problem:

$$\min_{\mathbf{p}} \ \max_k \ h\left(I_k(\mathbf{p})/I_{\text{des}}\right)$$
$$\text{s.t. } 0 \leq p_j \leq p_{\max}$$

$h(u) = \max\{u, 1/u\}$

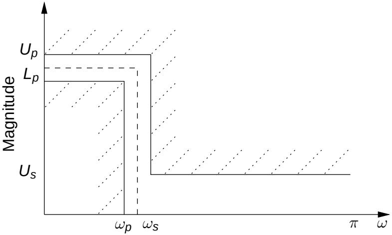


- $h(u)$ is convex; max of convex functions is convex
- Equivalent problem is convex; can write as linear obj. with linear and second order cone constrs
- Reliably solvable for global opt. with modest effort
- So what do you tell the ski operator?
- Since we can reliably obtain global optimum, we can confidently say that if that solution is not good enough, must change configuration (move/more towers)

Another example

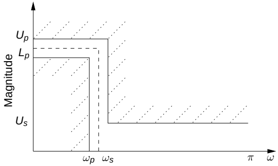
- Previous problem was reformulated as a convex one
- In this case we won't be so lucky
- However, we will show that convex opt. still has an important role to play

Consider the previous FIR filter design problem:



- Identify variables: L discrete-valued filter coefficients
- Constraints: magnitude response lies within mask
- Objective: stop band energy

Initial formulation



- Let $\mathbf{v}(\omega) = [1, e^{j\omega}, e^{j2\omega}, \ldots, e^{j(L-1)\omega}]^T$; $\quad X(e^{j\omega}) = \mathbf{v}(\omega)^H\mathbf{x}$
- Formulation:

$$\begin{aligned} \min_{\mathbf{x}\in\mathcal{X}} \quad & E_s = \frac{1}{\pi}\int_{\omega_s}^{\pi} \left|\mathbf{v}(\omega)^H\mathbf{x}\right|^2 d\omega = \mathbf{x}^T\mathbf{Q}\mathbf{x} \\ \text{subject to} \quad & |\mathbf{v}(\omega)^H\mathbf{x}| \le U_p \quad \forall\omega \in [0, \omega_s) \\ & |\mathbf{v}(\omega)^H\mathbf{x}| \ge L_p \quad \forall\omega \in [0, \omega_p] \\ & |\mathbf{v}(\omega)^H\mathbf{x}| \le U_s \quad \forall\omega \in [\omega_s, \pi] \end{aligned}$$

- Design question:
  - What is the inherent trade-off between E_s and U_s?
  - i.e., What is the region of all achievable (U_s, E_s) pairs?

Analysis of init. formulation

- Initial formulation

$$\begin{aligned} \min_{\mathbf{x}\in\mathcal{X}} \quad & \mathbf{x}^T\mathbf{Q}\mathbf{x} \\ \text{subject to} \quad & |\mathbf{v}(\omega)^H\mathbf{x}| \leq U_p \quad \forall\omega \in [0, \omega_s) \\ & |\mathbf{v}(\omega)^H\mathbf{x}| \geq L_p \quad \forall\omega \in [0, \omega_p] \\ & |\mathbf{v}(\omega)^H\mathbf{x}| \leq U_s \quad \forall\omega \in [\omega_s, \pi] \end{aligned}$$

- Coefficients are discrete: non-convex
- Relax that constraint to allow real coefficients
- Will give outer bound on set of achievable (U_s, E_s) pairs
- Relaxed formulation, with squared constraints:

$$\begin{aligned} \min_{\mathbf{h}\in\mathbb{R}^L} \quad & \mathbf{h}^T\mathbf{Q}\mathbf{h} \\ \text{subject to} \quad & |\mathbf{v}(\omega)^H\mathbf{h}|^2 \leq U_p^2 \quad \forall\omega \in [0, \omega_s) \\ & |\mathbf{v}(\omega)^H\mathbf{h}|^2 \geq L_p^2 \quad \forall\omega \in [0, \omega_p] \\ & |\mathbf{v}(\omega)^H\mathbf{h}|^2 \leq U_s^2 \quad \forall\omega \in [\omega_s, \pi] \end{aligned}$$

Analyze relaxed formulation

- Relaxed formulation, return to integral objective:

$$\begin{aligned}\min_{\mathbf{h}\in\mathbb{R}^L} \quad & \frac{1}{\pi}\int_{\omega_s}^{\pi} \left|\mathbf{v}(\omega)^H\mathbf{h}\right|^2 d\omega \\ \text{subject to} \quad & |\mathbf{v}(\omega)^H\mathbf{h}|^2 \le U_p^2 \quad \forall\omega\in[0,\omega_s) \\ & |\mathbf{v}(\omega)^H\mathbf{h}|^2 \ge L_p^2 \quad \forall\omega\in[0,\omega_p] \\ & |\mathbf{v}(\omega)^H\mathbf{h}|^2 \le U_s^2 \quad \forall\omega\in[\omega_s,\pi]\end{aligned}$$

- Second constraint: lower bound on convex quadratic; non-convex; what to do?
- Observation: Everything is a function of $|H(e^{j\omega})|^2$
- Observation: $|H(e^{j\omega})|^2 = R_h(e^{j\omega})$, where $R_h(e^{j\omega})$ is the Fourier Transform of the autocorrelation of $h[k]$
- Observation: $R_h(e^{j\omega}) = \tilde{\mathbf{v}}(\omega)^T\tilde{\mathbf{r}}_h$;
  $\tilde{\mathbf{v}}(\omega) = [1, 2\cos(\omega), \ldots, 2\cos((L-1)\omega)]$;
  $\tilde{\mathbf{r}}_h$ contains “right half” of autocorrelation; linear

Transformed relaxed formulation

$$
\begin{aligned}
\min_{\tilde{\mathbf{r}}_h \in \mathbb{R}^L} \quad & \frac{1}{\pi}\int_{\omega_s}^{\pi} R_h(e^{j\omega})\, d\omega = \mathbf{p}^T \tilde{\mathbf{r}}_h \\
\text{subject to} \quad & \tilde{\mathbf{v}}(\omega)^T \tilde{\mathbf{r}}_h \leq U_p^2 \quad \forall \omega \in [0, \omega_s) \\
& \tilde{\mathbf{v}}(\omega)^T \tilde{\mathbf{r}}_h \geq L_p^2 \quad \forall \omega \in [0, \omega_p] \\
& \tilde{\mathbf{v}}(\omega)^T \tilde{\mathbf{r}}_h \leq U_s^2 \quad \forall \omega \in [\omega_s, \pi] \\
& \tilde{\mathbf{v}}(\omega)^T \tilde{\mathbf{r}}_h \geq 0 \quad \forall \omega \in [0, \pi]
\end{aligned}
$$

- Linear program! but
- How many constraints? ∞
- Options:
  - Discretize and tighten: e.g., $\tilde{\mathbf{v}}(\omega_i)\tilde{\mathbf{r}}_h \leq U_P^2 - \epsilon_N$ for relevant $\omega_i = \pi i/N$, plus band edges typically $N = KL$, $K \in [8, 16]$ allows ϵ_N to be small
  - Represent exactly using linear matrix inequalities

Comparisons

Original formulation

$$\begin{aligned}\min_{\mathbf{x}\in\mathcal{X}} \ & \mathbf{x}^T\mathbf{Q}\mathbf{x} \\ \text{s.t.} \ & |\mathbf{v}(\omega)^H\mathbf{x}| \leq U_p \ \forall\omega\in[0,\omega_s) \\ & |\mathbf{v}(\omega)^H\mathbf{x}| \geq L_p \ \forall\omega\in[0,\omega_p] \\ & |\mathbf{v}(\omega)^H\mathbf{x}| \leq U_s \ \forall\omega\in[\omega_s,\pi]\end{aligned}$$

Non-convex

Transformed relaxed

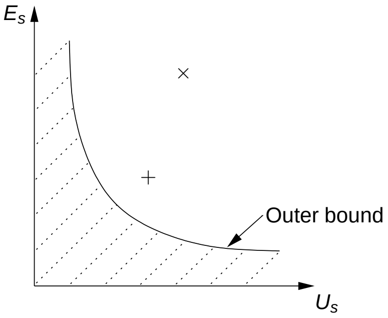
$$\begin{aligned}\min_{\tilde{\mathbf{r}}_h\in\mathbb{R}^L} \ & \mathbf{p}^T\tilde{\mathbf{r}}_h \\ \text{s.t.} \ & \tilde{\mathbf{v}}(\omega)^T\tilde{\mathbf{r}}_h \leq U_p^2 \ \forall\omega\in[0,\omega_s) \\ & \tilde{\mathbf{v}}(\omega)^T\tilde{\mathbf{r}}_h \geq L_p^2 \ \forall\omega\in[0,\omega_p] \\ & \tilde{\mathbf{v}}(\omega)^T\tilde{\mathbf{r}}_h \leq U_s^2 \ \forall\omega\in[\omega_s,\pi] \\ & \tilde{\mathbf{v}}(\omega)^T\tilde{\mathbf{r}}_h \geq 0 \ \forall\omega\in[0,\pi]\end{aligned}$$

Convex

Using transformed relaxed problem:

- Efficiently gen. outer bound on achievable (U_s, E_s) region by solving problem for different values of U_s
- Gen. an optimal $\mathbf{h}\in\mathbb{R}^L$ by spectral factorization
- Gen. good $\mathbf{x}\in\mathcal{X}$ by (randomized) rounding and/or local search
- When should we stop?

A conceptual figure



- Outer bound: solve transformed relaxed problem for different values of U_s; convex, global optimum reliably obtained
- If your current best discrete coeff. filter achieves $+$, you might be satisfied; you might stop your search
- If your current best discrete coeff. filter achieves $\times$, if you are not yet exhausted, probably keep looking

Applics in SPAWC areas

Recommended entry points to literature:

- Luo, *Mathematical Programming*, ser. B, 97:177–207, 2003
- *IEEE J. Select. Areas Communications*, Aug. 2006, especially tutorial by Luo and Yu
- *IEEE J. Select. Topics Signal Processing*, Dec. 2007
- Palomar and Eldar (Eds), *Convex Optimization in Signal Processing and Communications*, Cambridge, 2010
- *IEEE Signal Processing Magazine*, May 2010

Outline of rest of the session

- A sampling of the family of convex functions
- Quasi-convexity
- Beyond reliable algorithms, what does convexity offer?
- Using convexity in problems that remain non-convex
- Other tools for certain non-convex problems
- What about problems where only an oracle is available?
- What about functions that are uncertain?

Generic formulation

- Generic parameter design problem

$$\begin{aligned} \min_{\mathbf{x}\in\mathcal{X}} \quad & f_0(\mathbf{x}) \\ \text{subject to} \quad & f_m(\mathbf{x}) \leq \xi_m \\ & g_q(\mathbf{x}) = \zeta_q \end{aligned}$$

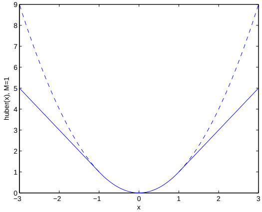
  or show that there is no feasible $\mathbf{x}$
- Convexity (almost always) yields reliable algo for a global opt
- For convexity:
  - $\mathcal{X}$ must be convex
  - $g_q(\mathbf{x})$ must be linear (affine)
  - for $f(\mathbf{x})$'s convexity suffices
- Quite a rich family of sets and functions
- These are the “target” functions when you try to find a convex problem related to the original
- Too many to list; see Boyd & Vandenberghe, CVX docs
- Some “art” in how to use the list

Some simple convex sets $\mathcal{X}$

- Polyhedron: $\{\mathbf{x} \mid \mathbf{a}_i^T \mathbf{x} \leq b_i\}$
- Second order cone: $\{(\mathbf{x}, t) \mid \|\mathbf{x}\|_2 \leq t\}$
  ice cream cone
- Semidefinite cone: $\{\mathbf{X} \mid \mathbf{X} = \mathbf{X}^T,\ \lambda_i(\mathbf{X}) \geq 0\}$
- Intersection preserves convexity

Some simple convex functions

- Linear (affine): $\mathbf{a}^T\mathbf{x} + b$
- Convex quadratic: $\mathbf{x}^T\mathbf{Q}\mathbf{x} + 2\mathbf{p}^T\mathbf{x} + r$, with $\mathbf{Q} \succcurlyeq \mathbf{0}$
- Abs. value: $|x|$; exp: e^{ax}; neg log: $-\log(x)$
- Sizes:
  - Norm: $\|\mathbf{x}\|_p$, $p \geq 1$; $1, \infty$: linear; 2: convex quad.
  - $\text{Huber}_M(x) = \begin{cases} x^2 & \text{for } |x| \leq M \\ 2M|x| - M^2 & \text{for } |x| > M \end{cases}$



Simple relationships

- Simple relationships that preserve convexity:
  - $\mathbf{x} \leftarrow \mathbf{A}\mathbf{x} + \mathbf{b}$
  - $\sum_i f_i(\mathbf{x})$
  - $\max_i f_i(\mathbf{x})$
  - Also composition of certain classes of functions

Application: Sparsity

- Even with these simple cases, we can take this on
- Given a set of m linear equations in n variables, $m < n$, find the most sparse ϵ solution
- “Problem first” approach:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{x}\|_0 \quad \text{subject to } \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2 \leq \epsilon$$

  where $\|\mathbf{x}\|_0$ is number of non-zero elements; not convex
- The 0-quasi-norm penalizes all non-zero elements equally; Norms: penalty increases with mag. of element
- Challenge: find convex f'n that behaves somewhat like $\|\cdot\|_0$
- p-norms, $p \geq 1$, are convex; which imposes smallest penalty on large elements? $p = 1$
- Hence, approximate original problem by following convex one

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{x}\|_1 \quad \text{subject to } \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2 \leq \epsilon$$

  Much can be said about probability that solutions coincide.

Geometric programs

- $\log \sum_i \exp(\mathbf{a}_i^T \mathbf{x} + b_i)$: convex
- Consider the problem
$$\min_{\mathbf{x} \in \mathbb{R}^n_{++}} \; f_0(\mathbf{x}) \quad \text{subject to } f_m(\mathbf{x}) \leq 1$$
where, with $a_{mki} \in \mathbb{R}$, functions take the form
$$f_m(\mathbf{x}) = \sum_{k=1}^{K} c_{mk}\, x_1^{a_{mk1}}\, x_2^{a_{mk2}} \, \ldots \, x_n^{a_{mkn}}$$

- This is called a geometric program
- Also arises in power allocation in wireless
- Not convex. However, let $y_i = \log(x_i)$.
- GP can be precisely transformed to
$$\begin{aligned} \min_{\mathbf{y} \in \mathbb{R}^n} \quad & \log\Big(\sum_{k=1}^{K} \exp(\mathbf{a}_{0k}^T \mathbf{y} + b_{0k})\Big) \\ \text{subject to} \quad & \log\Big(\sum_{k=1}^{K} \exp(\mathbf{a}_{mk}^T \mathbf{y} + b_{mk})\Big) \leq 0 \end{aligned}$$
where $b_{mk} = \log(c_{mk})$. Convex

Some matrix functions

- $\mathrm{trace}(\mathbf{A}^T\mathbf{X})$: convex
- Schur complement: If $\mathbf{A} \succ \mathbf{0}$, then

$$\mathbf{C} - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B} \succcurlyeq \mathbf{0} \iff \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix} \succcurlyeq \mathbf{0}$$

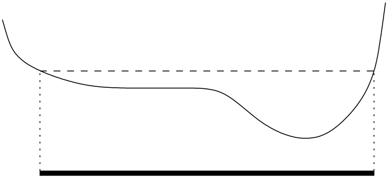
  If $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are linear in the variables, then LHS is not convex, in general, but RHS is convex

- A consequence

$$\begin{aligned} \sigma_{\max}(\mathbf{X}) \leq \tau &\iff \tau^2\mathbf{I} - \mathbf{X}^T\mathbf{X} \succcurlyeq \mathbf{0} \\ &\iff \tau\mathbf{I} - \mathbf{X}^T\mathbf{X}/\tau \succcurlyeq \mathbf{0} \\ &\iff \begin{bmatrix} \tau\mathbf{I} & \mathbf{X} \\ \mathbf{X}^T & \tau\mathbf{I} \end{bmatrix} \succcurlyeq \mathbf{0} \end{aligned}$$

Quasi-convexity

A function is quasi-convex if its sublevel sets are convex



- For convex constrs and quasi-convex obj., given γ consider

$$\begin{array}{rl} \text{find} & \mathbf{x} \\ \text{subject to} & f_0(\mathbf{x}) \leq \gamma; \quad f_m(\mathbf{x}) \leq \xi_m; \quad g_q(\mathbf{x}) \leq \zeta_q \end{array}$$

  A convex feasibility problem
- There is a single threshold for feasibility
- Use bisection on γ to find that thresh.; hence efficient algo
- Unfort. sum of quasi convex is not necess. quasi-convex
- A direct application: joint power and resource allocation in half-duplex cooperative communications

Quasi convexity: Design spec's

- If T_α is a nested family of convex sets, with $T_{\alpha_1} \subseteq T_{\alpha_2}$ for $\alpha_1 \leq \alpha_2$, then

$$\inf_{\mathbf{x},\alpha} \alpha \quad \text{such that } \exists\, \mathbf{x} \in T_\alpha$$

  can be handled in the same way

- Eng. interp: T_α represents design spec's; tighter for smaller α

- Applic's in filter design (when all other constr's convex):
  - minimum length filter that satisfies specifications
  - with previous mask and $\tilde{\mathbf{r}}_h$: min. stop-band edge

Beyond reliable algorithms

- Reasonably widely known that convexity (almost always) yields reliable algorithm for a global optimum
- What else does convexity offer?
  - Efficiently computable inherent trade-offs between competing criteria (first half)
  - Can assess the size of suboptimal set
  - Can gain considerable insight using duality and optimality conditions
    - Duality: lower bound on optimal value; often tight
    - Insight into structure of opt. sol'n (more efficient algos)
    - Some insight into how to modify configuration

Size of ϵ-subopt. set

- Set of feasible points with objective within ϵ of optimal
- For convex problems, this set can be approximated using straightforward convex opt. problems
- Impact on design problems
  - if ϵ-suboptimal set is large
    - lots of nearly optimal solutions
      might exploit this by optimizing a secondary obj.
- Impact on estimation problems
  - if ϵ-suboptimal set is large
    - many plausible solutions
      suggests low confidence in estimate

Duality

- For simplicity, consider ineq. constrs only, $\xi_m = 0$.
  Primal prob: $p^\star = \min_{\mathbf{x}} f_0(\mathbf{x})$ subject to $f_m(\mathbf{x}) \leq 0$
- Define Lagrangian: $L(\mathbf{x}, \boldsymbol{\lambda}) = f_0(\mathbf{x}) + \sum_m \lambda_m f_m(\mathbf{x})$
- Define Lagrangian dual function: $g(\boldsymbol{\lambda}) = \inf_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda})$
  Concave, even if $f_i(\mathbf{x})$ not convex
- For any $\boldsymbol{\lambda} \succcurlyeq \mathbf{0}$ and any feasible $\mathbf{x}$: $g(\boldsymbol{\lambda}) \leq f_0(\mathbf{x})$
  Hence, $g(\boldsymbol{\lambda}) \leq p^\star$
- What is best lower bound? $d^\star = \max_{\boldsymbol{\lambda} \succcurlyeq \mathbf{0}} g(\boldsymbol{\lambda})$
- In general, $d^\star \leq p^\star$.
  For convex problems with a strictly feasible point, equality! (strong duality)
- Some consequences:
  - can develop algo's with rigorous stopping criteria
  - can verify infeasibility

Optimality conditions

- Again for simplified problem: $\min_{\mathbf{x}} f_0(\mathbf{x})$ s.t. $f_m(\mathbf{x}) \leq 0$. Consider case of differentiable $f_i(\mathbf{x})$
- For a "regular" point, necessary conditions for optimality:

$$\begin{aligned} f_m(\mathbf{x}^\star) &\leq 0 \\ \lambda_m^\star &\geq 0 \\ \nabla f_0(\mathbf{x}^\star) + \sum_m \lambda_m^\star \nabla f_m(\mathbf{x}^\star) &= 0 \\ \lambda_m^\star f_m(\mathbf{x}^\star) &= 0 \end{aligned}$$

- For convex problems, under certain constraint qualifications (including strong duality), these are also sufficient
- Analysis of this set of non-linear equations can yield insight into optimal solution; e.g., structure

Sensitivities of config.

- Perturb the simplified prob: $\min_{\mathbf{x}} f_0(\mathbf{x})$ s.t. $f_m(\mathbf{x}) \leq \delta_m$
- Do we have to re-solve the problem?
- Under strong duality, some insight is already available:
  - Tightening:
    if $\lambda_m^\star$ is large, $\delta_m < 0$ greatly increases $p^\star$
  - Loosening:
    if $\lambda_m^\star$ is small, $\delta_m > 0$ does not greatly decrease $p^\star$
- In design setting, tells us what not to do to the configuration to reduce $p^\star$
- If, in addition, objective changes smoothly with δ_m's
  - $\lambda_m^\star$ is local sensitivity
  - so for small changes we get symmetric insight

Convexity offers

Beyond reliable algorithms for a globally optimal solution

We have highlighted the fact that you can

- Efficiently compute inherent trade-offs between competing criteria
- Assess the size of suboptimal set
- Gain considerable insight using duality and optimality conditions

What about non-convex case?

- In the general case, what you see is all you've got
- How can convexity help?
- We will investigate a few ways
  - Restriction and relaxation
    - bounds on inherent trade-offs
    - generating (provably) 'good' solutions
    - generating 'good' starting points for further search
  - Global optimization:
    - using (convex) relaxation in branch-and-bound algorithm
  - Local optimization:
    - using sequential convex approximation
- We will also look at one other approach that is useful in some non-convex problems

Assess trade-offs in non-convex

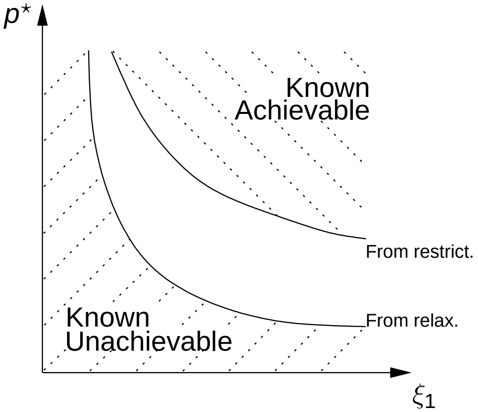
- Consider a simple problem: $\min_{\mathbf{x}} f_0(\mathbf{x})$ s.t. $f_1(\mathbf{x}) \leq \xi_1$
- Consider the trade-off between opt. value and ξ_1; i.e., $p^\star(\xi_1)$
- If $f_0(\mathbf{x})$ and $f_1(\mathbf{x})$ convex, can efficiently find trade-off by solving problem for different values of ξ_1.
- General non-convex case:
  - Even for one value of ξ_1, problem is hard to solve
  - Very hard to obtain inherent trade-off
  - Typically, all you have is best trade-off that has been found so far
  - What to do? How can convexity help?

Relaxation and restriction

- Relaxation:
  - loosen, or remove, constraints
  - feasible set expands;
  - generates lower bound on solution of original prob
- Restriction:
  - tighten, or add, constraints
  - feasible set shrinks;
  - generates upper bound on solution of original prob

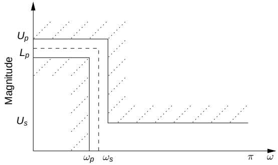
- If you can find a convex relaxation;
  get outer bound on trade-off region
- If you can find a convex restriction;
  get inner bound on trade-off region

A conceptual figure



Example of restriction

Recall filter design problem, relaxed for real coefficients:



$$
\begin{aligned}
\min_{\mathbf{h}\in\mathbb{R}^L} \quad & \mathbf{h}^T\mathbf{Q}\mathbf{h} \\
\text{subject to} \quad & |\mathbf{v}(\omega)^H\mathbf{h}|^2 \leq U_p^2 \quad \forall\omega \in [0, \omega_s) \\
& |\mathbf{v}(\omega)^H\mathbf{h}|^2 \geq L_p^2 \quad \forall\omega \in [0, \omega_p] \\
& |\mathbf{v}(\omega)^H\mathbf{h}|^2 \leq U_s^2 \quad \forall\omega \in [\omega_s, \pi]
\end{aligned}
$$

- Second constraint non-convex
- If you restrict to linear phase filter and constrain the sign, this constraint becomes linear, and hence convex
- Other constr's also become linear; obj. remains conv. quad.
- Hence, in this case, phase lin. generates convex restriction

Another ex. of relaxation

This time, focus is on generating ‘good’ soln,
although lower bounds are generated along the way

ML MIMO/MU detection for binary inputs, known channel, AWGN

$$\min_{\mathbf{x}\in\{-1,1\}^n} \|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2$$

Convex quadratic objective; non-convex constraints

- “Full” relaxation: $\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2$
  - Least-squares; closed-form solution
  - Once solved, (randomly) round to binary vector
- Box relaxation: $\min_{\mathbf{x}\in[-1,1]^n} \|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2$
  - Convex problem, of the same dimension
  - Clearly tighter relaxation
  - Once solved, (randomly) round to binary vector
- Semidefinite relaxation
  - A different relaxation; generates a matrix variable
  - Bounds on accuracy (worst-case)
  - Tends to be significantly tighter in practice

Semidefinite relaxation

- Rewrite $\min_{\mathbf{x}\in[-1,1]^n} \|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2$ as $\max_{\tilde{\mathbf{x}}\in[-1,1]^{n+1}} \tilde{\mathbf{x}}^T\mathbf{Q}\tilde{\mathbf{x}}$ s.t. $\tilde{x}_{n+1}=1$
- Using $\tilde{\mathbf{x}}^T\mathbf{Q}\tilde{\mathbf{x}} = \text{trace}(\mathbf{Q}\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T)$, rewrite as $\max_{\tilde{\mathbf{x}}\in[-1,1]^{n+1},\mathbf{X}\in\mathbb{S}^{n+1}} \text{trace}(\mathbf{Q}\mathbf{X})$ s.t. $\tilde{x}_{n+1}=1, \;\; \mathbf{X}=\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T$
- Rewrite again

$$\begin{aligned} \max_{\mathbf{X}\in\mathbb{S}^{n+1}} \quad & \text{trace}(\mathbf{Q}\mathbf{X}) \\ \text{subject to} \quad & \big[\mathbf{X}\big]_{ii} = 1 \\ & \mathbf{X} \succcurlyeq \mathbf{0} \\ & \text{rank}(\mathbf{X}) = 1 \end{aligned}$$

  Now “hardness” manifests as rank constraint
  Drop rank constraint to get semidefinite relaxation

- Generate candidate $\tilde{\mathbf{x}}$ using $\mathbf{L}\mathbf{u}$, where $\mathbf{L}$ is Cholesky factor of $\mathbf{X}_{\text{opt}}$, $\mathbf{u}$ is a normalized Gaussian rv

Convexity and global opt.

- In prev. ex's we obtained global lower bounds by relax.
- Can (local) relaxation help us find globally optimal solutions?
- Branch-and-bound: Basic principles
    - Partition the feasible set, and on each partition
        - Determine a lower bound on min. value of $f_0(\mathbf{x})$ on the partition, possibly by solving a convex relaxation
        - Determine an upper bound on the min. value of $f_0(\mathbf{x})$ on the partition, possibly by (coarse) local optimization or by solving a convex restriction
    - Compare lowest of lower bounds with lowest of upper
    - If not within desired accuracy, partition (one of) the existing partitions, and repeat
- Note recursive partitioning gives rise to a tree structure
- If the lower bound at one node of tree exceeds the upper bound at another; subtree below can be pruned
- Wide variety of tree searches available, incl. "best first" based on lower bounds
- SPAWC application: sphere decoder

Convexity and local opt.

- Although it will find an optimal solution, branch-and-bound is typically rather time consuming
- Alternative: accept suboptimality; run a local optimization algorithm from a number of starting points and pick the best
- Convexity plays a role in a large number of local optimization algorithms
- Emphasis here is on local approximation, rather than relaxation or restriction
- However, (global) relaxations and restrictions, as well as (global) convex approximations, may provide useful starting points.

Convexity and local opt.

- Let's look at a naive algorithm. At each iterate
  - Construct a quadratic approximation of objective and linearize the constraints
  - Take a step in direction that minimizes this approx.
  - Repeat until a measure of convergence satisfied
- Observations:
  - When the Hessian is positive definite,
    the approximate problem is convex:
    convex quadratic obj. with linear constr's
  - However, curvature info. of constraints is lost
  - Can be recovered by replacing obj. by Lagrangian and jointly optimizing over variables and multipliers
  - This is the basic principle that underlies sequential quadratic programming
- Other approximations can be used at each iterate.
  Convexity often plays a guiding role in the choice of approx.

Opt. on manifolds

- Although we have talked a lot about the use of convexity in non-convex problems, it is important to be aware of other potentially useful techniques
- As an example, we consider optimization on manifolds

- In some non-convex problems, feasible set has a perceptible structure
- In some cases feasible set forms a manifold
- In some cases, can construct optimization algorithms such that iterates remain on the manifold
- Some examples in SPAWC areas:
  - $\min f_0(\mathbf{X})$ over tall $\mathbf{X}$ s.t. $\mathbf{X}^T\mathbf{X} = \mathbf{I}$
    Stiefel manifold
  - if $f_0(\mathbf{XQ}) = f_0(\mathbf{X})$ for orthogonal $\mathbf{Q}$, it is the subspaces that matter; Grassmannian manifold

Opt. with oracles

- Generic problem

$$\begin{aligned} \min_{\mathbf{x}\in\mathcal{X}} \quad & f_0(\mathbf{x}) \\ \text{subject to} \quad & f_m(\mathbf{x}) \leq \xi_m \\ & g_q(\mathbf{x}) = \zeta_q \end{aligned}$$

- What if we don't have a formula for $f_0(\mathbf{x})$?
  perhaps just a numerical code (outcome of a PDE solver); might take several days to evaluate one point
- Could try pattern search, but we would like to try to use some of our insight into the problem
- Try to construct a surrogate optimization problem to guide where to evaluate the objective
- Key current applic's in aerospace (wing tips, rotor blades), microwave filter design, etc

Robust optimization

$$\begin{aligned} \min_{\mathbf{x}\in\mathcal{X}} \quad & f_0(\mathbf{x}) \\ \text{subject to} \quad & f_m(\mathbf{x}) \leq \xi_m \\ & g_q(\mathbf{x}) = \zeta_q \end{aligned}$$

What if we don't know these functions precisely?
e.g., imperfect CSI

Let's just look at a linear constraint, $\mathbf{a}^T\mathbf{x} \leq b$, with $\mathbf{a}$ uncertain

Possible models

- Distribution for $\mathbf{a}$, ask for $E_{\mathbf{a}}\{\mathbf{a}^T\mathbf{x}\} \leq b$
  Constraint satisfied on average
- $\mathbf{a}$ in a convex bounded set $\mathcal{A}$, ask for $\mathbf{a}^T\mathbf{x} \leq b$ for all $\mathbf{a} \in \mathcal{A}$
  Constraint always satisfied
- Distribution for $\mathbf{a}$, ask for $\Pr\big(\mathbf{a}^T\mathbf{x} \leq b\big) \geq 1 - \epsilon$
  Chance constrained; reminiscent of outage

Robust optimization

- In some cases, the requirement specified via an uncertain convex constraints can be precisely characterized using deterministic convex constraints (of possibly different type)
- In some other cases, one can obtain a set of deterministic convex constraints that are conservative, in sense that they guarantee that the requirement is satisfied
- In the case in which the underlying constraints are not convex, things are much more difficult

Summary

- This has been a whirlwind tour!
- Take home messages
  - “Problem first” methodology
  - Convexity buys you more than just a nice algorithm
  - Convex opt. still has much to offer when problem is not convex
- Additional information:
  - List of recommended entry points to the literature
  - Some references on the topics discussed
  - Further reading on some other aspects of (convex) optimization that have been applied SPAWC areas

Recommended entry points

- Convex optimization
  - Boyd & Vandenberghe, *Convex Optimization*, Cambridge, 2004
  - Bertsekas, *Convex Optimization Theory*, Athena Scientific, 2009
  - Grant, *CVX software*, http://cvxr.com
- Continuous optimization
  - Nocedal & Wright, *Numerical Optimization*, 2nd ed, Springer, 2006
  - Bertsekas, *Nonlinear Programming*, 2nd ed, Athena Scientific, 1999
  - Antoniou & Lu, *Practical Optimization: Algorithms and Engineering Applications*, Springer, 2007
- Global optimization
  - Lawler & Wood, “Branch-and-bound methods: A survey”, *Operations Research*, Jul.–Aug. 1966
  - Horst *et al*, *Intro. to Global Optimization*, 2nd ed, Springer, 2001
  - Neumaier, “Complete search in continuous global optimization . . . ”, *Acta Numerica*, May 2004
- Robust optimization
  - Ben-Tal *et al*, *Robust Optimization*, Princeton, 2009

Recommended entry points

- Applications in SPAWC areas
  - Luo, *Mathematical Programming*, ser. B, 97:177–207, 2003
  - *IEEE J. Select. Areas Communications*, Aug. 2006 especially tutorial of Luo and Yu
  - *IEEE J. Select. Topics Signal Processing*, Dec. 2007
  - Palomar and Eldar (Eds), *Convex Optimization in Signal Processing and Communications*, Cambridge, 2010
  - *IEEE Signal Processing Magazine*, May 2010

References

A somewhat biased list

- FIR filter design
  - Davidson, *IEEE Signal Processing Mag.*, May 2010, and references therein
- Sparsity
  - *IEEE Signal Processing Mag.*, Mar. 2008
  - *Proceedings of the IEEE*, June 2010
- Geometric programs and applic's in SPAWC areas
  - Boyd *et al*, *Optimization and Engineering*, 2007
  - Chiang, *Found. & Trends Commun. Info. Theory*, Aug. 2005
  - Gohary & Davidson, *EURASIP JWCN*, 2009
- Quasi-convexity in cooperative communications
  - Mesbah and Davidson, in *Proc. ICASSP*, 2010, and refs therein
- ϵ-suboptimal set
  - Skaf & Boyd, *Optimization and Engineering*, June 2010
- Semidefinite relaxation
  - Luo *et al*, *IEEE Signal Processing Mag.*, May 2010

References

- Branch-and-bound interpretation of sphere decoder
  - Murugan *et al*, *IEEE Trans. Info. Theory*, Mar. 2006
- Optimization on manifolds and applic's
  - Edelman *et al*, *SIAM J. Matrix Anal. Applic*, 1998
  - Manton, *IEEE Trans. Signal Processing*, Mar. 2002
  - Gohary & Davidson, *IEEE Trans. Info. Theory*, Mar. 2009
- Optimization with oracles
  - Booker *et al*, *Structural and Multidisciplinary Optimization*, Feb. 1999
  - Koziel *et al*, *IEEE Microwave Mag.*, Dec. 2008
- Some applic's of robust optimization in SPAWC areas
  - Vorobyov *et al*, *IEEE Trans. Signal Processing*, Feb. 2003
  - Gershman *et al*, *IEEE Signal Processing Mag.*, May 2010
  - Vucic & Boche, and Shenouda & Davidson,
    IEEE Trans. Signal Processing, Feb./May 2009
  - Rong *et al*, *IEEE J. Select. Areas Commun.*, Aug. 2006
  - Shenouda and Davidson, in *Proc. Asilomar Conf.*, 2008

Further reading

Here are some starting points for information on a few related topics that have applications in SPAWC areas

- Majorization and Schur convexity
  - Palomar & Jiang, *Found. & Trends Commun. Info. Theory*, 2006
  - Shenouda & Davidson, *IEEE J. Select. Areas Commun.*, Feb. 2008
- Decomposition
  - Palomar & Chiang, Lin *et al*, and Johansson *et al*, *IEEE J. Select. Areas Commun.*, Aug. 2006
  - Chiang *et al*, *Proceedings of the IEEE*, Jan. 2007
- Gossip algorithms and consensus
  - Olfati-Saber *et al*, *Proceedings of the IEEE*, Jan. 2007
  - Dimakis *et al*, http://arxiv.org/abs/1003.5309, Mar. 2010
- Game theory
  - *IEEE Signal Processing Mag.*, Sep. 2009
- Dynamic programming
  - Bertsekas, *Dynamic Programming and Optimal Control*, 3rd ed, Athena Scientific, 2005/2007