Convex-Concave Procedure

Stephen Boyd Max Schaller Daniel Cederberg

February 18, 2025

Overview

the **convex-concave procedure** (CCP)

- is a heuristic method for solving a specific type of nonconvex problem
- involves solving a (typically small) sequence of convex problems
- leverages expressiveness and reliability of convex optimization
- is a good street-fighting trick to know about

Outline

Difference of convex functions

Convex-concave procedure

Examples

Difference of convex functions

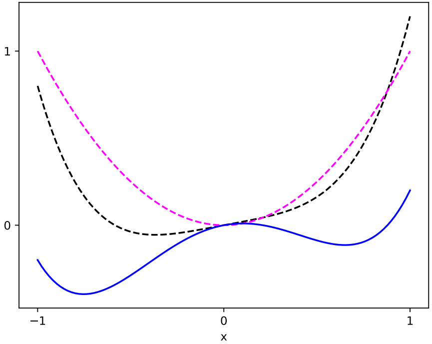
- a **difference of convex** (DC) function $h : \mathbf{R}^n \to \mathbf{R}$ has the form

$$h(x) = f(x) - g(x)$$

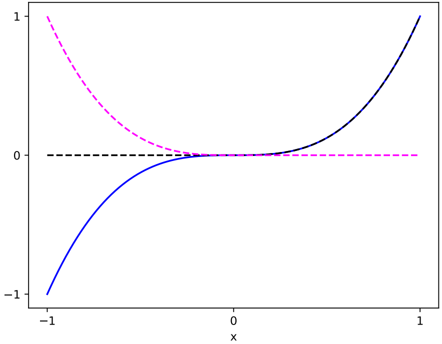
  with f and g convex
- any function with continuous second derivative has this form, but most useful when we have explicit expressions for f and g

Examples

$$f(x) = x^4 + 0.2x, \quad g(x) = x^2$$



$$f(x) = (x^3)_+, \quad g(x) = (x^3)_-$$



$f(x)$ in black, $g(x)$ in magenta, $h(x) = f(x) - g(x)$ in blue

Quadratic function as DC

- (nonconvex) quadratic function $h(x) = (1/2)x^T Px + q^T x + r$, $P \in \mathbf{S}^n$
- decompose P into its PSD and NSD parts $P = P_{\text{psd}} - P_{\text{nsd}}$, $P_{\text{psd}}, P_{\text{nsd}} \succeq 0$
  - $P = Q\Lambda Q^T$ is eigenvalue decomposition
  - $P_{\text{psd}} = Q\Lambda_+ Q^T$, with $\Lambda_+ = \max\{0, \Lambda\}$ (elementwise)
  - $P_{\text{nsd}} = Q\Lambda_- Q^T$, with $\Lambda_- = \max\{0, -\Lambda\}$ (elementwise)
- express h in DC form as $h = f - g$ with

$$f(x) = (1/2)x^T P_{\text{psd}} x + q^T x + r, \qquad g(x) = (1/2)x^T P_{\text{nsd}} x$$

A simple majorizer for a DC function

- if (convex) g is differentiable, then for all x

$$\hat{g}(x; z) = g(z) + \nabla g(z)^T (x - z) \leq g(x)$$

(for nondifferentible g, replace $\nabla g(x)$ with any subgradient)

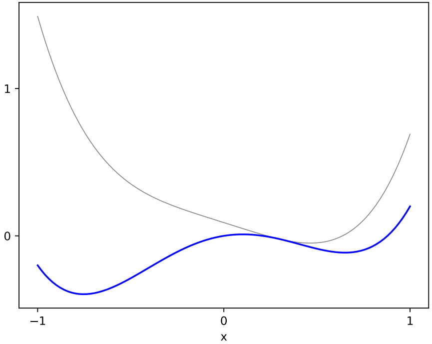
- for DC function $h = f - g$, define

$$\hat{h}(x; z) = f(x) - \hat{g}(x; z)$$

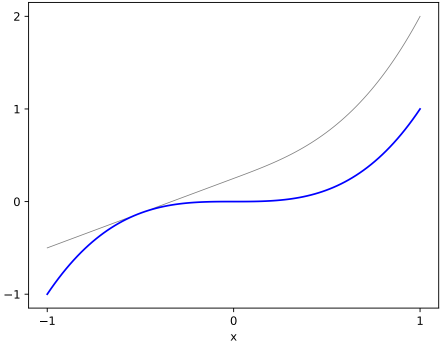
- $\hat{h}$ is convex (in x) and satisfies $\hat{h}(x; z) \geq h(x)$ for all x, $\hat{h}(z; z) = h(z)$
- *i.e.*, $\hat{h}(x; z)$ is a **majorizer** of g, tight at z

Examples

$$h(x) = x^4 + 0.2x - x^2,$$
majorized at $z = 0.3$



$$h(x) = (x^3)_+ - (x^3)_- = x^3,$$
majorized at $z = -0.5$



$h(x)$ in blue, $\hat{h}(x; z)$ in gray

Outline

Minimizing a DC function

- (unconstrained) **DC problem**: minimize DC function $h(x) = f(x) - g(x)$
- convex constraints can be handled as indicator functions added to f
- **convex-concave procedure** (CCP): iterate

$$x^{k+1} = \operatorname*{argmin}_x \hat{h}(x; x^k) = \operatorname*{argmin}_x \left(f(x) - \hat{g}(x; x^k) \right)$$

- x^{k+1} can be found via convex optimization
- CCP has no parameters, no line search / trust penalty, ...

Properties of CCP

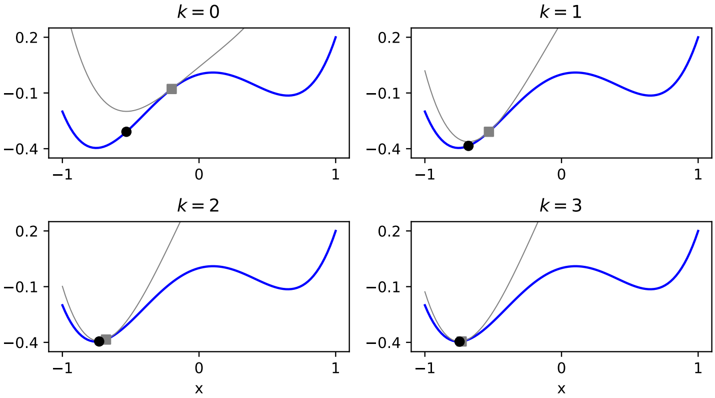
- CCP is a **descent method**:

$$h(x^{k+1}) \le \hat{h}(x^{k+1}; x^k) \le \hat{h}(x^k; x^k) = h(x^k)$$

- $h(x^k)$ converges, but not necessarily to $h^\star = \inf_x h(x)$
- ultimate value $\lim_{k\to\infty} h(x^k)$ depends on initial point x^0
- standard trick: run CCP for multiple initial points; take best ultimate point found

Example

$$f(x) = x^4 + 0.2x,\ g(x) = x^2, \text{ initialized at } x^0 = -0.2$$

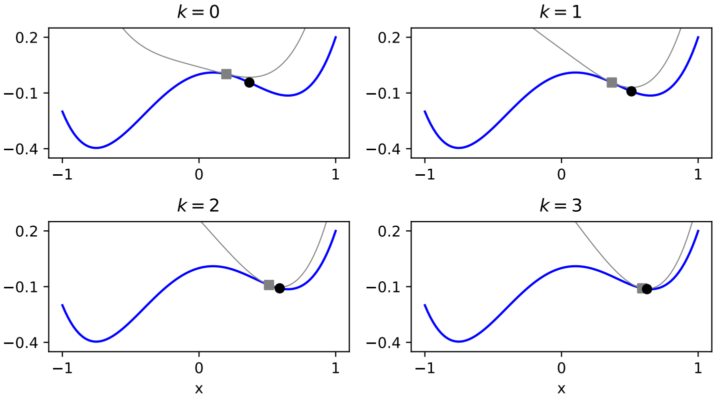


$h(x)$ in blue, $\hat{h}(x; x^k)$ in gray,

$(x^k, h(x^k))$ as squares, $(x^{k+1}, h(x^{k+1}))$ as circles

Example

$f(x) = x^4 + 0.2x$, $g(x) = x^2$, initialized at $x^0 = 0.2$



$h(x)$ in blue, $\hat{h}(x; x^k)$ in gray,
$(x^k, h(x^k))$ as squares, $(x^{k+1}, h(x^{k+1}))$ as circles

SAT problem

- SAT problem: find $x_i \in \{0, 1\}$ with $Ax \preceq b$
- includes 3-SAT; NP-hard
- encode Boolean constraint using $x_i \in \{0, 1\} \iff x_i^2 - x_i = 0$
- SAT problem equivalent to minimizing DC function $h = f - g$ with

$$f(x) = \begin{cases} 0 & Ax \preceq b, \; 0 \preceq x \preceq \mathbf{1} \\ \infty & \text{otherwise} \end{cases} \qquad g(x) = \sum_{i=1}^{n} (x_i^2 - x_i)$$

since $h(x) \geq 0$ for all x and

$$h(x) = 0 \quad \iff \quad Ax \preceq b, \quad x_i \in \{0, 1\}, \; i = 1, \ldots, n$$

SAT problem via CCP

- CCP: x^{k+1} is a solution of LP

$$\begin{array}{ll} \text{minimize} & (\mathbf{1} - 2x^k)^T x \\ \text{subject to} & Ax \preceq b, \quad 0 \preceq x \preceq \mathbf{1} \end{array}$$

- **example:** find $x \in \{0,1\}^4$ that satisfies predicate

$$(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_2 \vee x_3 \vee \neg x_4)$$

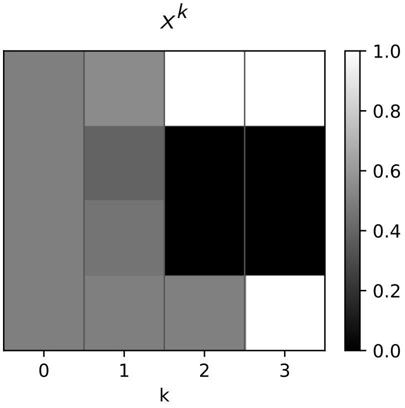
(a feasible 3-SAT instance with $n = 4$ variables and 3 clauses)

- can be represented as

$$A = \begin{bmatrix} -1 & -1 & 1 & 0 \\ 1 & 1 & 0 & -1 \\ 0 & 1 & -1 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

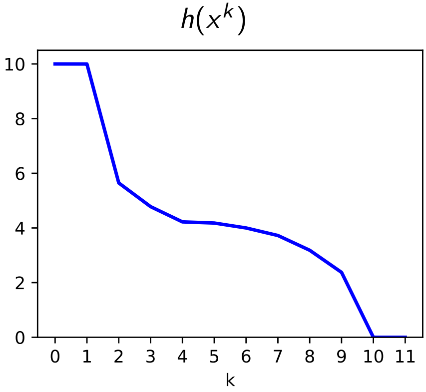
SAT problem via CCP

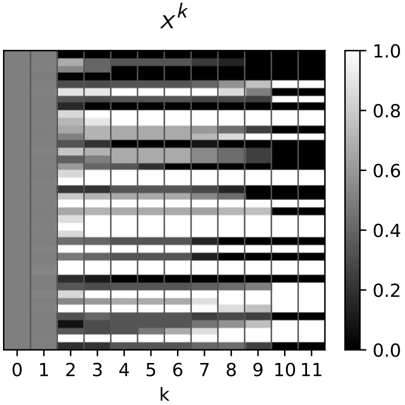
```
n = 4
x = Variable(n)
xk = Parameter(n)

obj = Minimize((ones(n) - 2 * xk) @ x)
constr = [A @ x <= b, 0 <= x, x <= 1]
prob = Problem(obj, constr)

xk.value = ones(n) / 2
for _ in range(3):
    prob.solve()
    xk.value = x.value
```



SAT problem via CCP

larger example, (feasible) 3-SAT instance with $n = 40$, 120 clauses





finds feasible point for 19% of random initializations

Constrained DC problem

- **DC problem** with DC inequality constraints has form

$$\begin{array}{ll} \text{minimize} & f_0(x) - g_0(x) \\ \text{subject to} & f_i(x) - g_i(x) \leq 0, \quad i = 1, \dots, m \end{array}$$

  f_i and g_i are convex

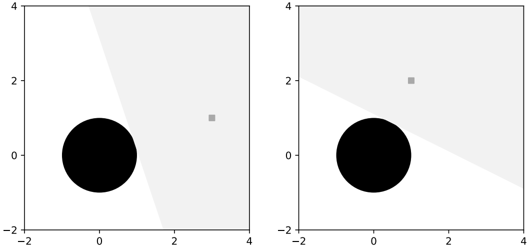
- CCP algorithm: x^{k+1} is solution of convex problem

$$\begin{array}{ll} \text{minimize} & f_0(x) - \hat{g}_0(x; x^k) \\ \text{subject to} & f_i(x) - \hat{g}_i(x; x^k) \leq 0, \quad i = 1, \dots, m \end{array}$$

- in words: linearize concave parts; solve; repeat
- if x^k is feasible, then so is x^{k+1}, and $h(x^{k+1}) \leq h(x^k)$
- CCP is a feasible descent method

Convex restrictions

- if $\hat{h}_i(x; x^k) = f_i(x) - \hat{g}_i(x; x^k) \leq 0$, then $h_i(x) \leq \hat{h}_i(x; x^k) \leq 0$
- so convexified constraint is a **convex restriction** of original constraint
- **example**: $f(x) = 0$, $g(x) = \|x\|_2 - 1$, $x^k = (3, 1)$ (left) and $x^k = (1, 2)$ (right)



$h(x) \leq 0$ in white, $\hat{h}(x; x^k) \leq 0$ in gray (x^k as gray dot)

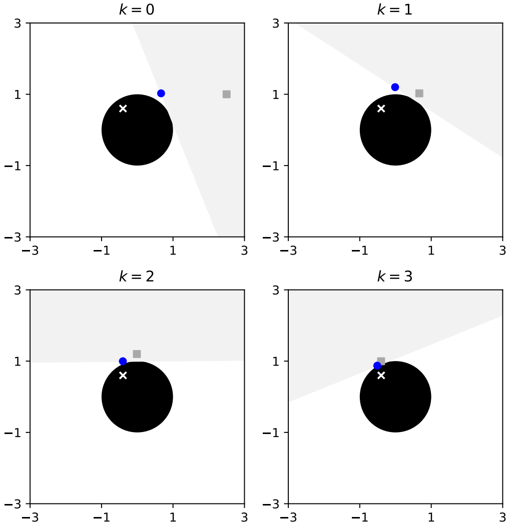
Convex restrictions

- **example**: CCP for simple problem

$$\begin{array}{ll} \text{minimize} & \|x - c\|_2 \\ \text{subject to} & \|x\|_2 \geq 1 \end{array}$$

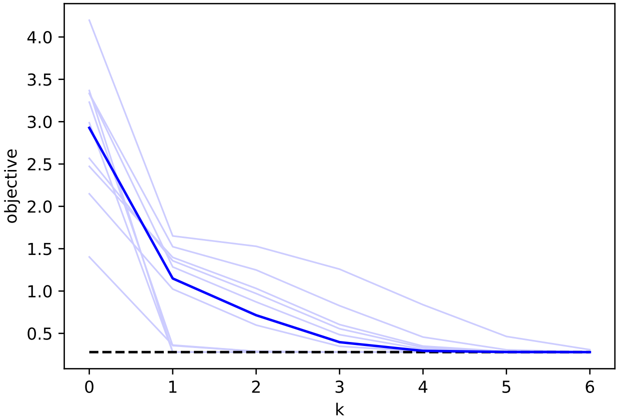
  with variable $x \in \mathbf{R}^2$, data c with $\|c\|_2 < 1$, $c \neq 0$
- *i.e.*, find closest point to c that is **outside** unit ball
- solution is $x^\star = c/\|c\|_2$
- consider $c = (-0.4, 0.6)$ and $x^0 = (2.5, 1)$

CCP iterations



x^k as a square, x^{k+1} as a circle, c as a cross

CCP iterations



$\|x^k - c\|_2$ in blue, $\|x^\star - c\|_2 = 1 - \|c\|_2$ as dashed black line

DC equality constraints

- linear equality constraints can just be added as indicator function to f
- DC equality constraints of the form $p_i(x) = q_i(x)$, with p_i and q_i convex, can be expressed as the pair of DC inequalities

$$p_i(x) - q_i(x) \leq 0, \qquad q_i(x) - p_i(x) \leq 0$$

Ensuring feasibility of convex subproblems

- convex subproblems can be infeasible, even if original problem is feasible
- introduce **slack variable** $s_i \geq 0$ for constraint i
- **penalty CCP** algorithm: x^{k+1} is solution of convex problem

$$
\begin{array}{ll}
\text{minimize} & f_0(x) - \hat{g}_0(x; x^k) + \tau_k \sum_{i=1}^m s_i \\
\text{subject to} & f_i(x) - \hat{g}_i(x; x^k) \leq s_i, \quad i = 1, \ldots, m \\
& s_i \geq 0, \quad i = 1, \ldots, m
\end{array}
$$

where $\tau_k > 0$ is increased between iterations

Disciplined convex-concave programming (DCCP)

- **disciplined convex-concave program** (DCCP) has form

$$\begin{array}{ll} \text{minimize/maximize} & o(x) \\ \text{subject to} & l_i(x) \sim r_i(x), \quad i = 1, \ldots, m, \end{array}$$

- o, l_i, r_i are DCP convex or concave expressions
- $\sim$ can be $\leq$, $\geq$, or $=$
- to minimize DC objective $f_0 - g_0$
  - introduce epigraph variable t and DCCP constraint $f_0(x) - g_0(x) \leq t$
  - minimize t
- implemented in DCCP package

Disciplined convex-concave programming (DCCP)

- **example (revisited)**:

$$\begin{array}{ll} \text{minimize} & \|x - c\|_2 \\ \text{subject to} & \|x\|_2 \geq 1 \end{array}$$

with variable $x \in \mathbf{R}^2$, data c

```
x = Variable(2)
obj = Minimize(norm(x - c))
constr = [norm(x) >= 1]
prob = Problem(obj, constr)

x.value = [2.5, 1]
prob.solve(method='dccp')
print(x.value)
```

Outline

Difference of convex functions

Convex-concave procedure

Examples

Path planning with obstacles

- find shortest path connecting points a and b in $\mathbf{R}^d$, avoiding m disks
- disk j has center c_j, radius r_j
- discretize path as points $x_0, \ldots, x_n$ and solve

$$\begin{array}{ll} \text{minimize} & L \\ \text{subject to} & x_0 = a, \quad x_n = b \\ & \|x_i - x_{i-1}\|_2 \leq L/n, \quad i = 1, \ldots, n \\ & \|x_i - c_j\|_2 \geq r_j, \quad i = 1, \ldots, n, \quad j = 1, \ldots, m \end{array}$$

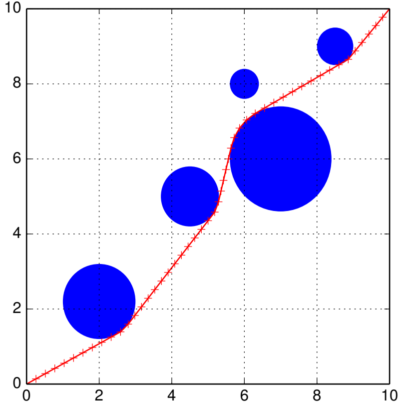
  with variables L and x
- useful initialization: straight line connecting a and b

Path planning with obstacles

```
L = Variable()
x = Variable((n+1, d))
L.value, x.value = ...  # initialize to straight line

constr = [x[0] == a, x[n] == b]
for i in range(1, n+1):
    constr += [norm(x[i] - x[i-1]) <= L/n]
for j in range(m):
    constr += [norm(x[i] - c[j]) >= r[j]]

prob = Problem(Minimize(L), constr)
prob.solve(method = 'dccp')
```

Path planning with obstacles

$d = 2, \ n = 50, \ m = 5$



Floor planning

- n rectangles with lower left corner (x_i, y_i), $i = 1, \ldots, n$ (variables)
- rectangle i has width w_i, height h_i (given)
- centers are $c_i = (x_i + w_i/2, y_i + h_i/2)$
- edge set $\mathcal{E} \subset \{1, \ldots, n\}^2$ contains pairs of rectangles we want to be close
- minimize $\sum_{(i,j)\in\mathcal{E}} \|c_i - c_j\|_2^2$
- without loss of generality, we can require $(1/n)\sum_{i=1}^n c_i = 0$
- these constraints and objective are convex; non-overlapping constraint is not

Non-overlapping constraint

- rectangles i and j don't overlap if

$$x_i + w_i \leq x_j \quad \vee \quad x_j + w_j \leq x_i \quad \vee \quad y_i + h_i \leq y_j \quad \vee \quad y_j + h_j \leq y_i, \quad i < j$$

  i.e., rectangle i is left of, right of, below, or above rectangle j
- express as inequality

$$\min\left(x_i + w_i - x_j, x_j + w_j - x_i, y_i + h_i - y_j, y_j + h_j - y_i\right) \leq 0$$

- left-hand side is concave function of (x, y)
- so non-overlapping can be expressed as a DC inequality constraint

Floor planning as DC problem

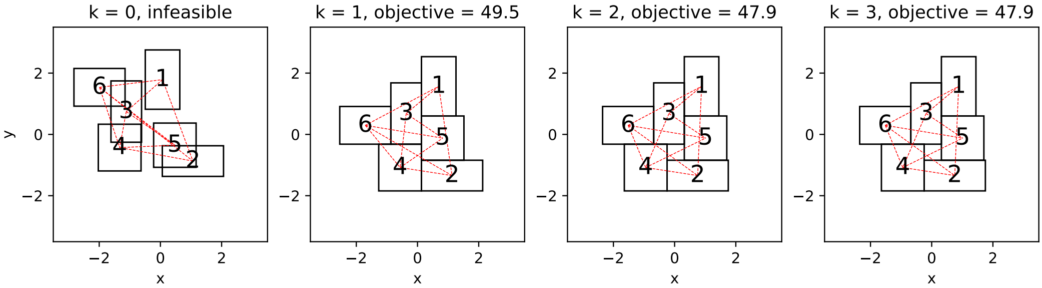
- DC formulation of floor planning problem:

$$
\begin{array}{ll}
\text{minimize} & \sum_{(i,j)\in\mathcal{E}} \|c_i - c_j\|_2^2 \\
\text{subject to} & \sum_{i=1}^n c_i = 0, \quad c_i = (x_i + w_i/2, y_i + h_i/2), \quad i = 1, \ldots, n, \\
& \min\left(x_i + w_i - x_j, x_j + w_j - x_i, y_i + h_i - y_j, y_j + h_j - y_i\right) \leq 0, \quad i < j
\end{array}
$$

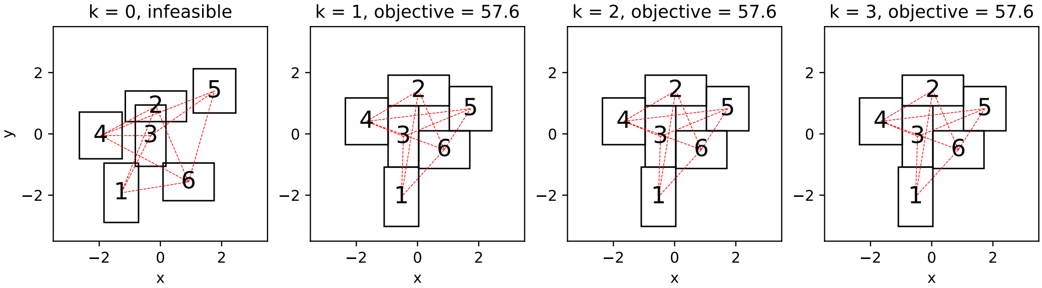
- x_i, y_i, and c_i are variables; w_i, h_i, and $\mathcal{E}$ are given

Floor planning example

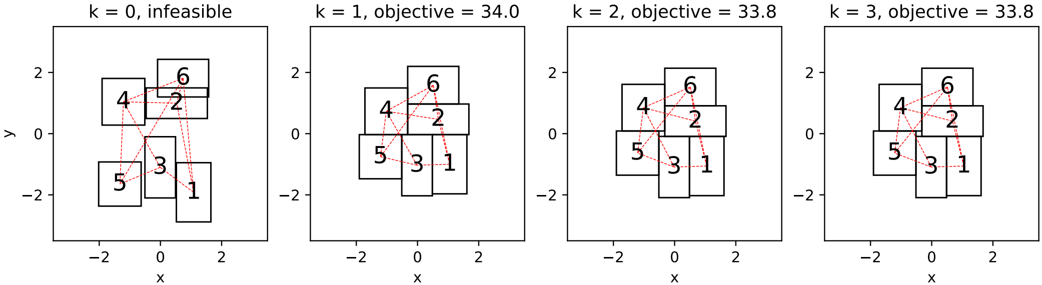
$n = 6$ rectangles; initial centers drawn uniformly from $[-2, 2]^2$



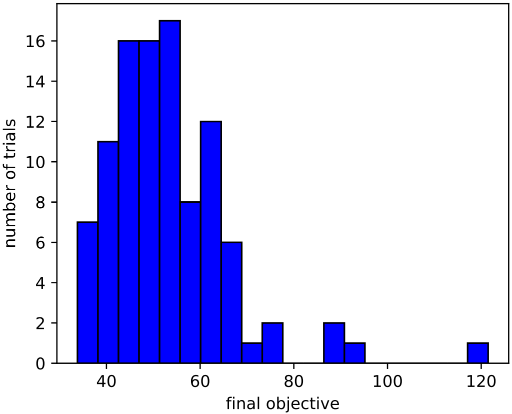
Floor planning example



Floor planning example



Floor planning example



Collision avoidance

- N vehicles moving simultaneously along (discretized) time $t = 0, \ldots, T$
- vehicle i aims to travel from initial position x_i^{init} to final position x_i^{final}
- trajectories described as

$$x_{i,t}, \quad i = 1, \ldots, N, \quad t = 0, \ldots, T$$

- vehicles must keep minimum distance d from each other at all times

Collision avoidance

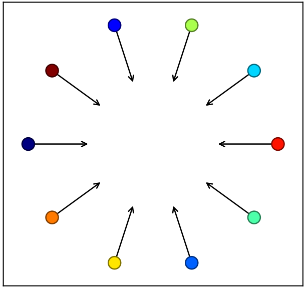
- minimize total length of travel while avoiding collisions as

$$
\begin{array}{ll}
\text{minimize} & \sum_{i=1}^{N} L_i \\
\text{subject to} & x_{i,0} = x_i^{\text{init}}, \quad x_{i,T} = x_i^{\text{final}}, \quad i = 1, \ldots, N, \\
& \|x_{i,t} - x_{i,t-1}\|_2 \leq L_i/T, \quad i = 1, \ldots, N, \quad t = 1, \ldots, T, \\
& \|x_{i,t} - x_{j,t}\|_2 \geq d, \quad i < j, \quad t = 0, \ldots, T
\end{array}
$$

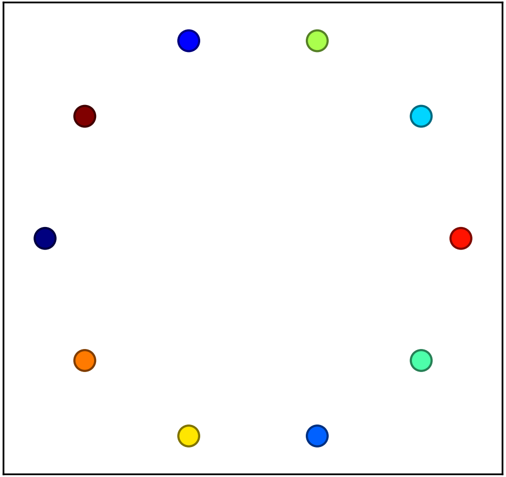
- $x_{i,j}$ and L_i are variables
- x_i^{init}, x_i^{final}, and d are given

Collision avoidance example

- $N = 10$ vehicles with starting points x_i^{init} uniformly on a circle around zero
- each vehicle aims to travel to opposite side of circle, $x_i^{\text{final}} = -x_i^{\text{init}}$
- minimum distance is $d = 1$



Collision avoidance example



$\ell_{1/2}$ regularized regression

- minimizing $\|Ax - b\|_2^2 + \lambda\|x\|_1$ where $A \in \mathbf{R}^{m\times n}$ tends to yield **sparse** x
- (approximately) minimizing $\|Ax - b\|_2^2 + \lambda \sum_i |x_i|^{1/2}$ can yield **sparser** x
- sometimes called $\ell_{1/2}$ regularization (but it's not a norm)
- not a convex problem

DC formulation

- DC problem

$$\begin{array}{ll} \text{minimize} & \|Ax - b\|_2^2 + \lambda \mathbf{1}^T t \\ \text{subject to} & t \succeq 0, \quad |x_i| \leq t_i^2, \quad i = 1, \ldots, n \end{array}$$

- CCP: x^{k+1} is solution of

$$\begin{array}{ll} \text{minimize} & \|Ax - b\|_2^2 + \lambda \mathbf{1}^T t \\ \text{subject to} & t \succeq 0, \quad |x_i| \leq (t_i^k)^2 + 2t_i^k(t_i - t_i^k), \quad i = 1, \ldots, n \end{array}$$

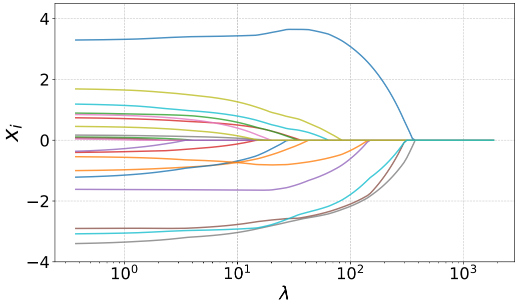
- so x^{k+1} is solution of weighted ℓ_1 regularized problem

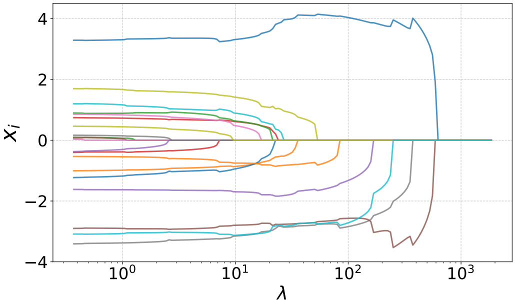
$$\text{minimize} \quad \|Ax - b\|_2^2 + \lambda \textstyle\sum_{i=1}^n \frac{1}{2t_i^k} |x_i|$$

with $t_i^{k+1} = \frac{1}{2t_i^k}|x_i^{k+1}| + \frac{1}{2} t_i^k$ (we can impose a small minimum value for t_i^k)

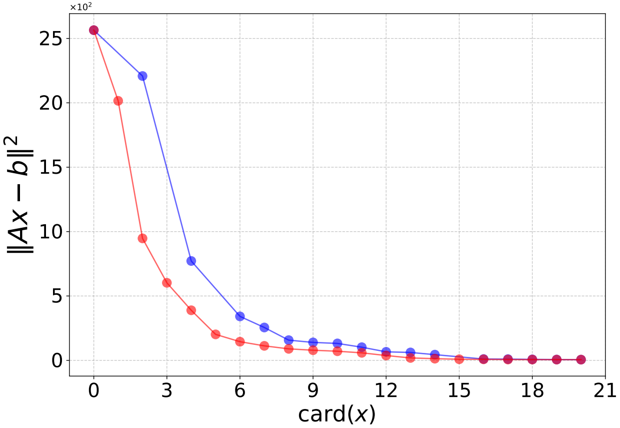
Example

- example with $n = 20$, $m = 40$
- regularization paths for ℓ_1 (left) and $\ell_{1/2}$ (right) regularization





Example



optimal trade off curves for ℓ_1 (blue) and $\ell_{1/2}$ (red) regularization