ALG 076

ALGORITHMS FOR CONSTRAINED NONLINEAR PROGRAMS.

PENALTY FUNCTION METHODS.

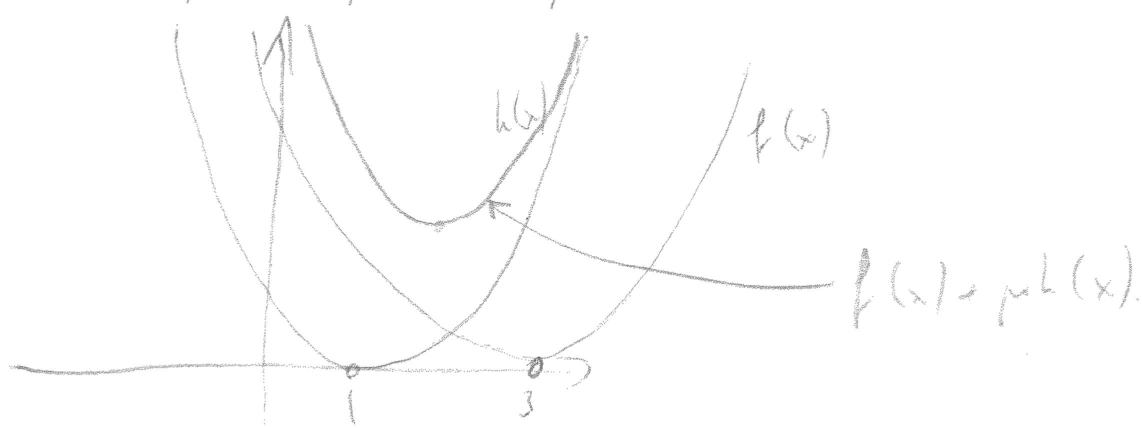
Eg: $\min \frac{1}{2}(x_1-3)^2 =: f(x)$

s.t. $h(x) := x_1 - 1 = 0$

Instead, try solving unconstrained problem

$$\min \quad f(x) + \mu h(x)^2 = \tfrac{1}{2}(x_1-3)^2 + \tfrac{\mu}{2}(x_1-1)^2$$

for some positive μ. As $\mu \to \infty$, will force constraint to hold at equality



E.g. $\min \ \frac{1}{2}(x_1-3)^2 =: f(x)$

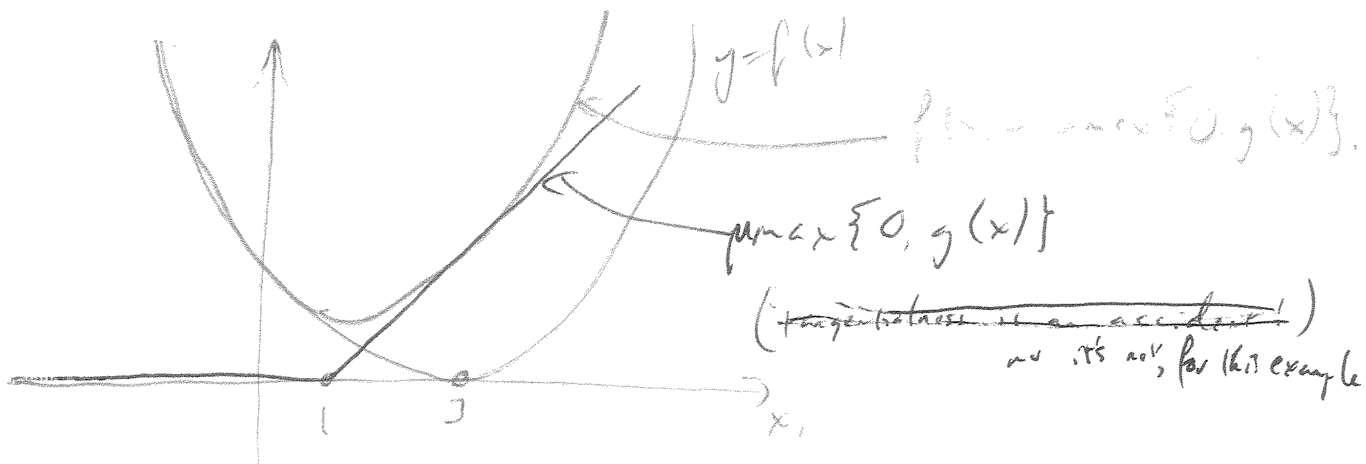
s.t. $g(x) = x_1 - 1 \leq 0$

No longer appropriate to use $f(x) + \mu g(x)^2$, since penalizes feasible points with $g(x) < 0$.

So use $f(x) + \mu \max\{0, g(x)\}$

or $f(x) + \mu \left(\max\{0, g(x)\}\right)^2$.

Thus only infeasible points are penalized.



In general:

$$\begin{aligned} \min \ & f(x) \\ \text{s.t.} \ & g_i(x) \leq 0 \quad i = 1, \ldots, m \\ & h_j(x) = 0 \quad j = 1, \ldots, p \end{aligned}$$

Use PENALTY FUNCTION

$$\alpha(x) = \sum_{i=1}^{m} \left(\max\{0, g_i(x)\}\right)^r + \sum_{j=1}^{p} \left(h_j(x)\right)^r$$

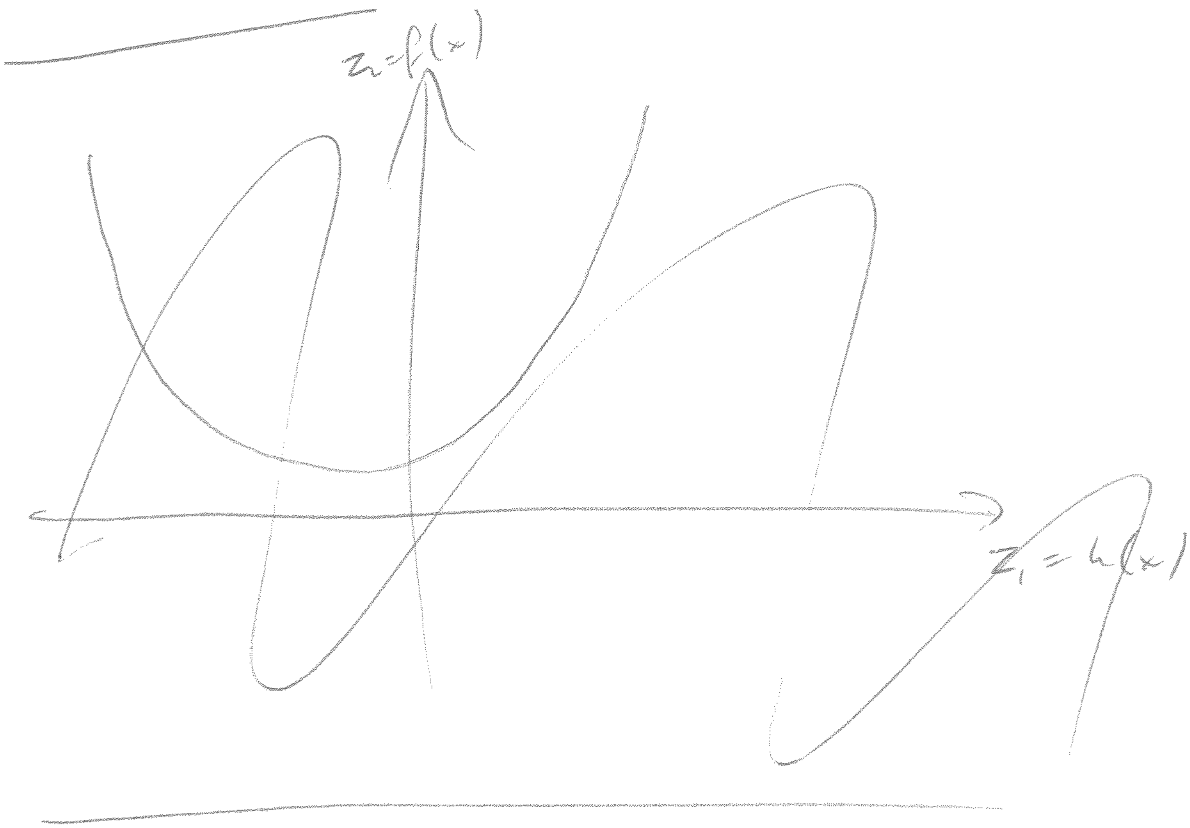
for some power r, e.g. $r = 1$ or 2.

Get AUXILIARY FUNCTION: $f(x) + \mu\alpha(x)$

Want to solve the unconstrained problem

$$\min_{x \in \mathbb{R}^n} f(x) + \mu\,\alpha(x)$$

Try different values of μ, $\mu \to +\infty$.



Require:

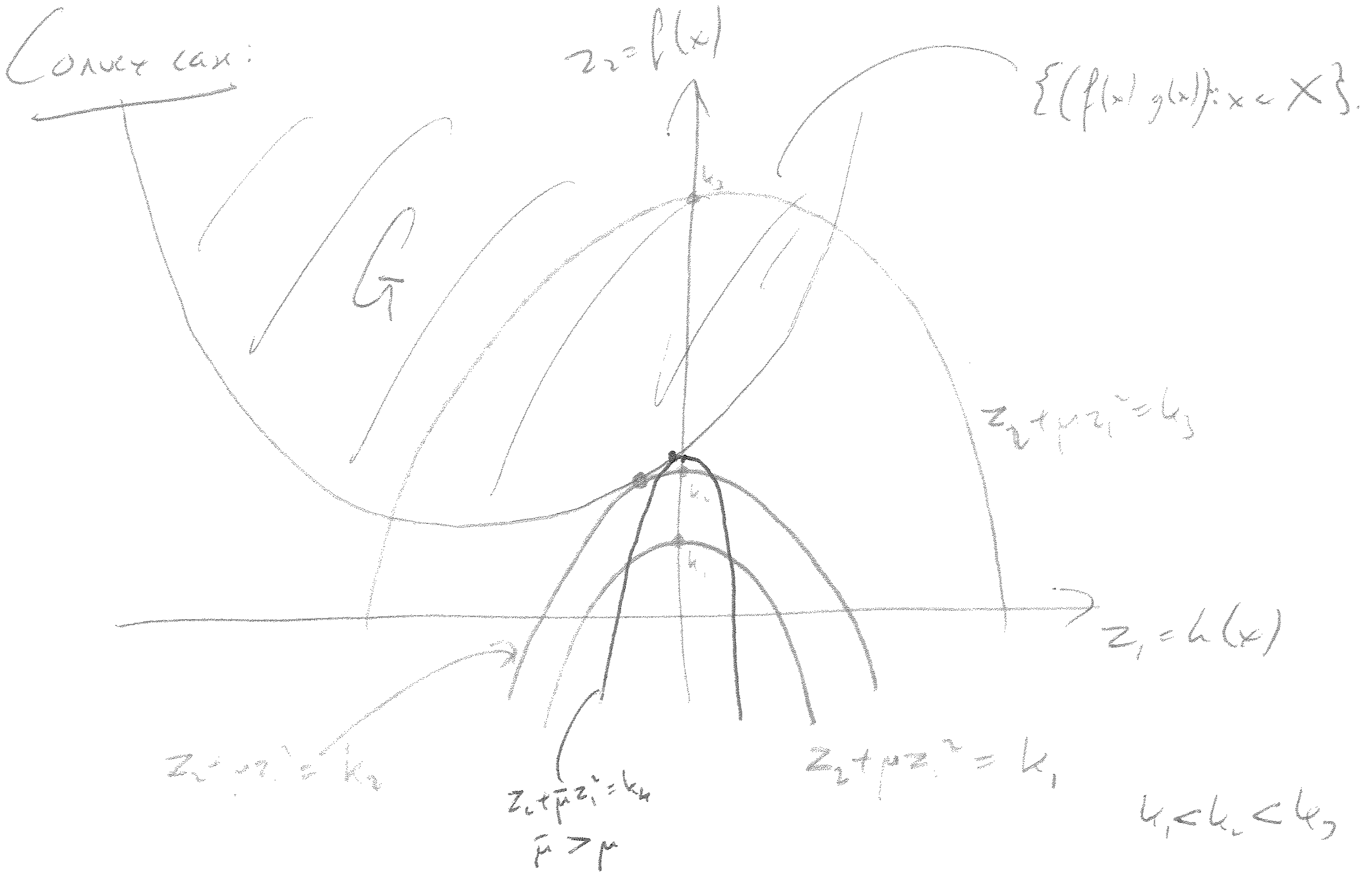
$$\alpha(x) = \sum_{i=1}^{m} \phi_i(g_i(x)) + \sum_{j=1}^{l} \psi_j(h_j(x)),$$

when

$$\phi_i(y) \begin{cases} = 0 & \text{if } y \le 0 \\ > 0 & \text{if } y > 0 \end{cases}$$

$$\psi_j(y) \begin{cases} > 0 & \text{if } y \ne 0 \\ = 0 & \text{if } y = 0. \end{cases}$$

Convex case:



$k_1 < k_2 < k_3$

Note that many points $(f(x), h(x))$ in G achieve $f(x) + \mu h(x)^2 = k_3$,

only one point achieves $f(x) + \mu h(x)^2 = k_2$

no points achieve $f(x) + \mu h(x)^2 = k_1$.

Optimal value of $\min f(x) + \mu h(x)^2$ s.t. $x \in X$ is k_2.

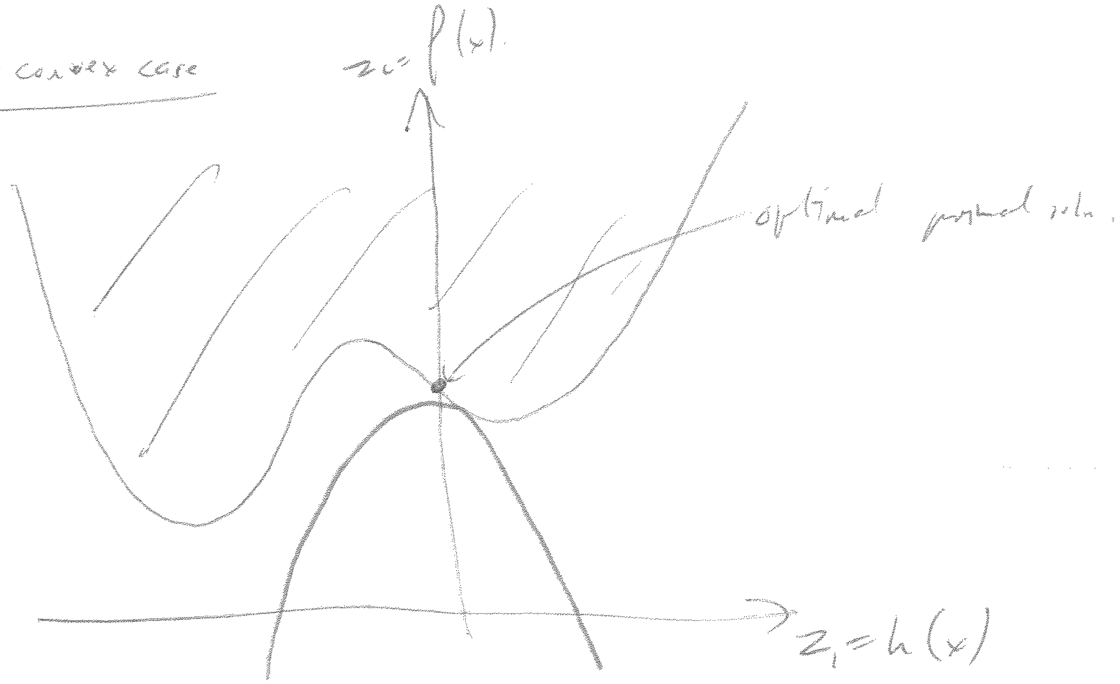
If choose $\bar{\mu} > \mu$, get value $> k_2$ and get closer to feasibility

~~If choose $\bar{\mu}$ big enough, will get optimal soln to original problem.~~

Note that contour $z_2 + \mu z_1^2 = k$ is always horizontal when $z_1 = 0$, so will never actually get to optimal primal solution.

Aug 80.

Non-convex case



Can still get ~~close to~~ arbitrarily close to optimal soln.

Theorem. Consider the following problem:

$$\begin{array}{lll} \min & f(x) & \\ \text{s.t.} & g_i(x) \le 0 & i=1,\dots,m \\ & h_j(x) = 0 & j=1,\dots,p \\ & x \in X \subseteq \mathbb{R}^n & \end{array} \quad (NLP)$$

where f, g_i, h_j continuous function on $\mathbb{R}^n$ and $X \neq \phi$. (Note: no convexity assumptions.)

Suppose (NLP) has a feasible solution and let $\alpha(x)$ be a continuous penalty function.

Suppose for each $\mu > 0$ there exists x_μ which solves $\min_{x \in X} f(x) + \mu\alpha(x)$, and that $\{x_\mu\}$ is contained in a compact subset of X.

Then,

$$\inf\{f(x) : g(x) \le 0, h(x) = 0, x \in X\} = \sup_{\mu \ge 0} \theta(\mu) = \lim_{\mu \to \infty} \theta(\mu)$$

where $\theta(\mu) = \inf_{x \in X}\{f(x) + \mu\alpha(x) : x \in X\} = f(x_\mu) + \mu\alpha(x_\mu)$.

Furthermore, the limit $\bar{x}$ of any convergent subsequence of $\{x_\mu\}$ is an optimal solution to the original problem, and $\mu\alpha(x_\mu) \to 0$ as $\mu \to \infty$.

(No proof — proof in book.)

This theorem justifies the penalty function approach.

Numerical considerations

Eg: $\min f(x)$ s.t. $h_j(x) = 0 \quad j = 1, \dots, p$, with opt soln $\bar{x}$.

Have $F(x) = f(x) + \mu \sum_{j=1}^{p} \psi_j(h_j(x))$ for penalty functions ψ_j.

At minimum, $DF(x) = 0$,

$$\text{ie, } \quad 0 = Df(x) + \sum_{j=1}^{p} \left(\mu \psi'(h_j(x))\right) Dh_j(x).$$

Since $x_\mu \to \bar{x}$, get $\mu \psi'(h_j(x)) \longrightarrow$ KKT multiplier.

Hessian is: $D^2F(x) = \left[D^2 f(x) + \sum_{j=1}^{p} \left(\mu \psi'(h_j(x))\right) D^2 h_j(x) \right]$

$$\cancel{+\sum_{j=1}^{p}\mu} \; + \underbrace{\mu \sum_{j=1}^{p} \psi''(h_j(x)) Dh_j(x) Dh_j(x)^T}.$$

$\longrightarrow D^2 L(\bar{x})$

If use quadratic penalty, get $\psi'' \longrightarrow$ constant, so this tends to a matrix of rank $\leq p$, with eigenvalues $\to \infty$

So get some eigenvalues $\to +\infty$, some finite eigenvalues.

Thus, steepest descent may well be disastrous, although Newton-type methods should work OK.

Thus, often use SUMT (Successive Unconstrained Minimization Technique):

~~Find~~ Choose initial x^0, set $k=0$, choose initial μ_0.

Solve the problem $\min_{\text{s.t. } x \in X} f(x) + \mu_k \alpha(x)$, starting from x_k.

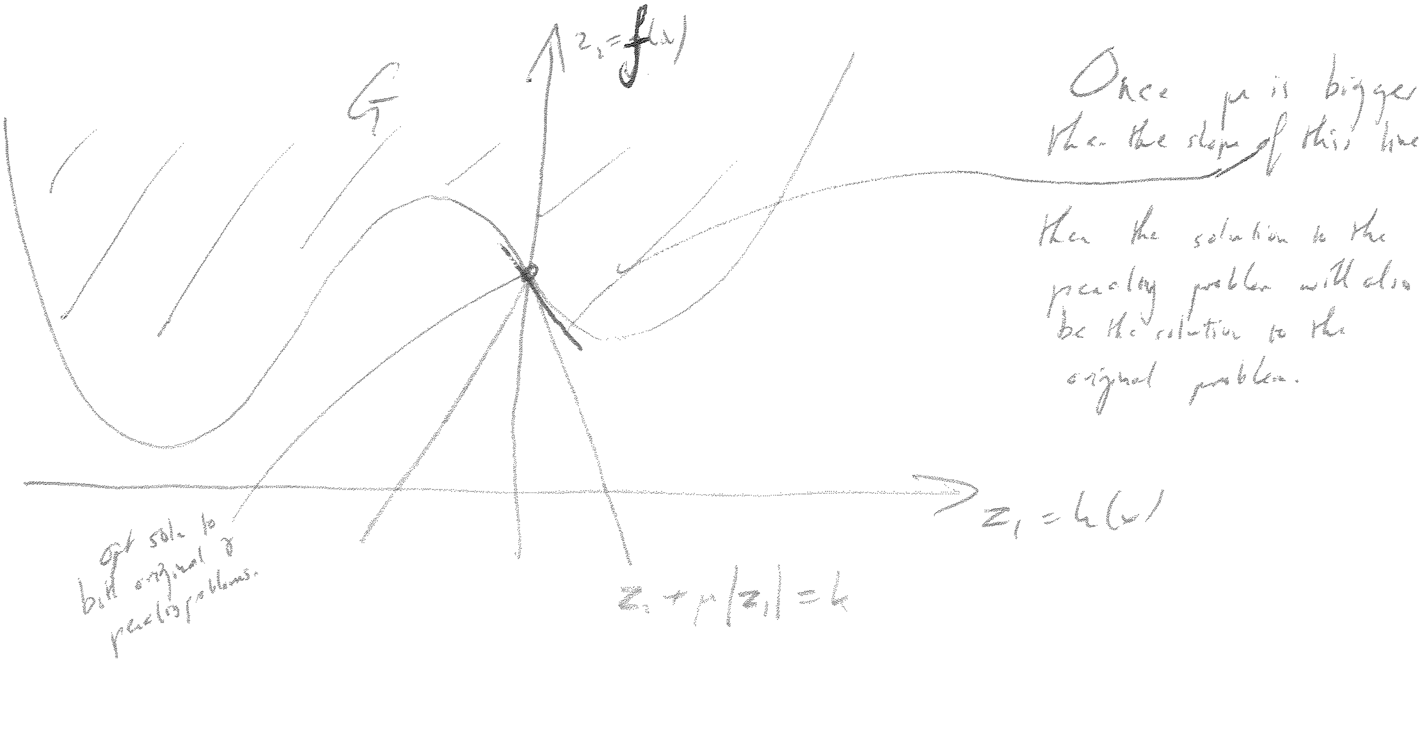
If satisfied, stop,

Else increase μ to give μ_{k+1}, set $k=k+1$, loop

~~EXACT~~

ABSOLUTE VALUE PENALTY FUNCTION:

Take $\psi_j(h(x)) = |h_j(x)|$. This is also called l_1-penalty function.



ALGO 84

__Theorem__ Consider

$$\begin{aligned} \min\ & f(x) \\ \text{s.t.}\ & g_i(x) \le 0 \quad i=1,\dots,m \qquad (NLP) \\ & h_j(x) = 0 \quad j=1,\dots,p \end{aligned}$$

Let $\bar{x}$ be KKT point w/ multipliers $\bar{u}_i,\ i \in I$, $\bar{v}_j, j=1,\dots,p$, where I = set of active ineq

Suppose f and g_i are convex, for $i \in I$, and h_j are affine.

Then, for $\mu \ge \max\{\bar{u}_i, i \in I, |\bar{v}_j|, j=1,\dots,p\}$, $\bar{x}$ also minimizes ...

the exact l_1-penalized objective function F_E:

$$F_E(x) = f(x) + \mu \sum_{i=1}^{m} \max\{0, g_i(x)\} + \mu \sum_{j=1}^{p} |h_j(x)|.$$

(No proof - see text for proof.)

This is called an EXACT penalty function because it is capable of recovering the optimal solution for the original problem EXACTLY for a finite value of μ.

Drawback to this penalty function:

It is not differentiable when $g_i(x) = 0$ or $h_j(x) = 0$.

AUGMENTED LAGRANGIAN PENALTY FUNCTIONS.

- Exact penalty function
- Differentiable.

Consider problems with just equality constraints: (for problems with ineq constraints, see text)

$$\min \; f(x)$$
$$\text{s.t.} \quad h_j(x) = 0 \qquad j = 1, \ldots, \ell.$$

Use penalty function:

$$F_{ALAG}(x) := \underbrace{f(x) + \sum_{j=1}^{\ell} v_j h_j(x)}_{\text{Lagrangian}} + \underbrace{\mu \sum_{j=1}^{\ell} h_j^2(x)}_{\text{augmented by quadratic penalty term.}}, \quad \text{for some constants } v_j.$$

Want $v_j \to$ optimal Lagrange multipliers.

Then

$$DF_{ALAG}(x) = \underbrace{Df(x) + \sum_{j=1}^{\ell} v_j Dh_j(x)}_{\text{tends to } DL(\bar{x}) = 0} + 2\mu \sum_{j=1}^{\ell} \underbrace{h_j(x)}_{\text{tends to } h_j(\bar{x}) = 0,\ \text{since } \bar{x} \text{ feasible}} Dh_j(x)$$

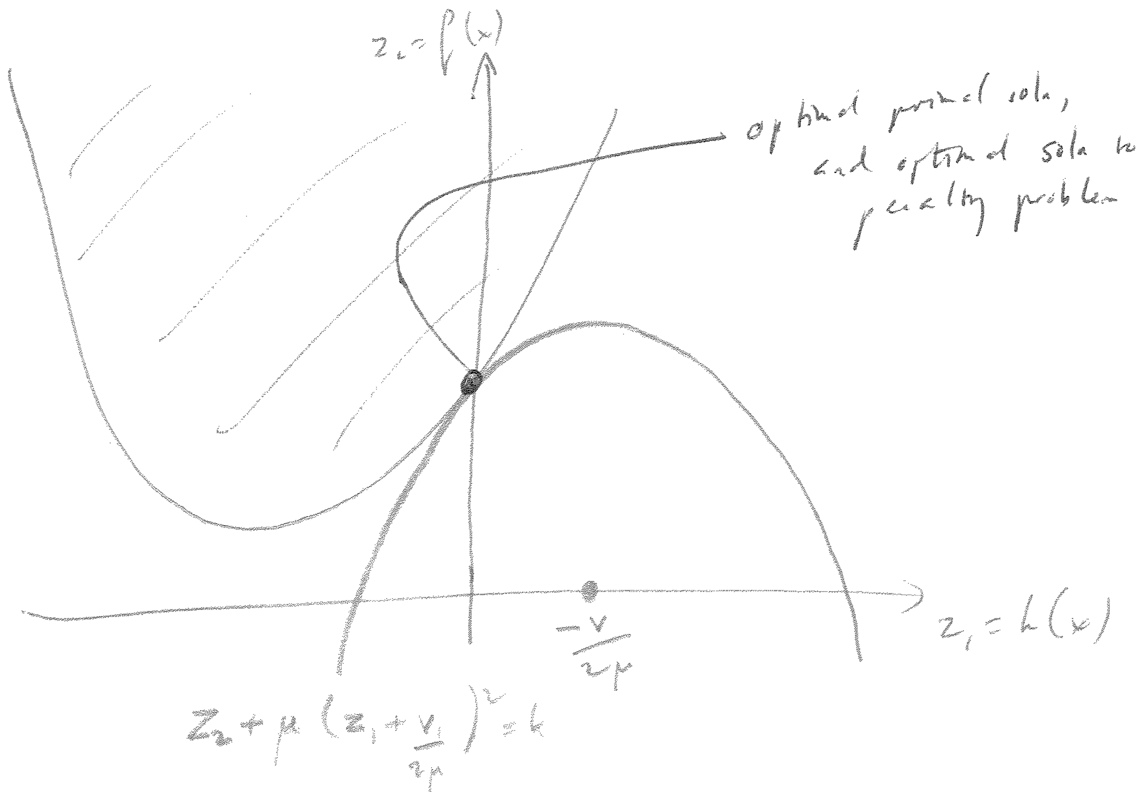
Thus, for some finite μ, appropriate v_j, could get optimal solution $\bar{x}$ as solution to

$$\min \; F_{ALAG}(x).$$

Geometrically:

Can write

$$F_{ALAG}(x) = \underbrace{f(x) + \mu \sum_{j=1}^{p} \left(h_j(x) + \frac{v_j}{2\mu} \right)^2}_{\substack{\text{Shift from } h_j(x) = 0 \\ \text{to } h_j(x) + \frac{v_j}{2\mu} = 0.}} - \frac{1}{\mu} \sum_{j=1}^{p} v_j^2$$



<u>Theorem</u> Consider the problem $\min f(x)$ s.t. $h_j(x) = 0$, $j = 1, \ldots, p$. (P)

Let the KKT solution $\bar{x}, \bar{v}$ satisfy the 2nd order sufficiency conditions for a local minimum. Then there exists a $\bar{\mu}$ such that for $\mu \geq \bar{\mu}$, the penalty function F_{ALAG} defined with $v = \bar{v}$ also achieves a strict local minimum at $\bar{x}$. In particular, if f is convex and h_j, $j = 1, \ldots, p$ are affine, then any minimizing solution $\bar{x}$ for (P) also minimizes F_{ALAG} for all $\mu \geq 0$.

NOTE. True because usually $\Sigma h_j^2(x)$ serves to convexify; here, don't need it to do so.

(No proof – see text.)

Note: If $\bar{x}, \bar{v}$ does not satisfy second order sufficiency conditions then may need $\mu \to \infty$ to recover $\bar{x}$.

METHOD OF MULTIPLIERS

Alternately update x and update Lagrange multipliers.

Take $F_{ALAG}(x) := f(x) + \Sigma v_j h_j(x) + \Sigma \mu_j h_j^2(x)$ (so μ_j can vary for different j.)

Alg:

<u>Initialize</u>: Choose x^0, $v_1, \ldots, v_p$, $\mu_1, \ldots, \mu_p$. Let $\text{viol}(x) = \max_{j=1,\ldots,p} \{|h_j(x)|\}$.
Set $\text{viol}(x^{-1}) = +\infty$

<u>Inner Loop</u>: ~~Re~~ find x^k to $\min F_{ALAG}(x)$.

If $\text{viol}(x^k)$ acceptably small, STOP with x^k declared (approx) ~~optimal~~ KKT point.

If $\text{viol}(x^k) > \frac{1}{4}\,\text{viol}(x^{k-1})$ then for j with $|h_j(x^k)| > \frac{1}{4}\,\text{viol}(h_j(x^k))$, set $\mu_j = 10\mu_j$,
Repeat inner loop

Else go to outer loop.

update Lagrange multipliers

<u>Outer Loop</u>: Replace v by $\bar{v}_{new}$: $(\bar{v}_{new})_j = v_j + 2\mu_j h_j(x^k)$
Set $k = k+1$.
Return to inner loop.

This is ∇F_{ALAG} s.t. ∇L

Barrier Function Methods

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \le 0 \qquad i=1,\dots,m \\ & x \in X. \end{aligned}$$

Assume $\exists\, x \in X$ with $g_i(x) < 0 \quad i=1,\dots,m$.

So, to handle equality constraints, need to eliminate some variables to get a lower dimensional feasible space.

Don't need to do this, especially for affine constraints. Just assume only have ineqs in what follows.

Eg: $\begin{aligned} \min \quad & x_1 - 2x_2 \\ \text{s.t.} \quad & x_1 + x_2 = 1 \\ & x_1, x_2 \ge 0 \end{aligned}$

is equivalent to: $\begin{aligned} \min \quad & (1-x_2) - 2x_2 \\ \text{s.t.} \quad & x_2 \ge 0 \\ & 1-x_2 \ge 0 \end{aligned}$

by setting $x_1 = 1 - x_2$.

But packages do handle them.

Nonlinear equality constraints are harder to handle.

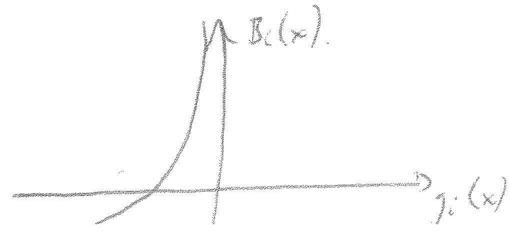
Use a barrier $B_i(x)$ to stop $g_i(x) > 0$.

true at least in a nbhd of $g_i(x)=0$. Need as written for cvgce theorem.

So solve unconstrained problem

$$\min_{x \in X} f(x) + \mu \sum B_i(x) \quad \text{where } B_i(x) \begin{cases} \ge 0 & \text{if } g_i(x) < 0 \\ \to +\infty & \text{as } g_i(x) \to 0_- . \end{cases}$$

Eg: $B_i(x) = \dfrac{-1}{g_i(x)}$, or $B_i(x) = -\ln(-g_i(x))$.



Let $B(x) = \sum_{i=1}^{m} B_i(x)$

The function $f(x) + \mu B(x)$ is then an auxiliary function, and we can solve the problem

$$\min_{x \in X} f(x) + \mu B(x)$$

Note that we need to start with an x strictly satisfying the constraints, and that at each iteration we have to ensure that we still strictly satisfy the constraints.

Can again use a SUMT approach:

Initialize: pick strictly feasible x^0, choose μ_0, set $k = 0$.

Main Step: Starting with x^k, solve the barrier subproblem

$$\begin{aligned} \min \quad & f(x) + \mu B(x) \\ \text{s.t.} \quad & g(x) < 0, \; x \in X \end{aligned}$$

Let x^{k+1} be optimal soln.

Loop/Terminate: If $\mu_k B(x^{k+1}) < \varepsilon$, STOP.

Else, set $\mu^{k+1} = \beta \mu^k$ for some $\beta \in (0,1)$.

Set $k = k+1$, Loop

<u>Theorem</u> Let ~~f~~ f and g be continuous functions, let $X \subseteq \mathbb{R}^n$ be nonempty, closed.

Assume ~~th~~ $\{x : g(x) < 0\} \neq \phi$.

Assume the problem $\min f(x)$ s.t. $g_i(x) \leq 0,\ i=1,\dots,m$, $x \in X$ has an optimal solution $\bar{x}$, and

that, ~~....~~ given any neighbourhood N of $\bar{x}$, there exists $x \in X \cap N$

with $g(x) < 0$.

Then $\min\{f(x) : g(x) \leq 0, x \in X\} = \lim_{\mu \to 0+} \theta(\mu) = \inf_{\mu > 0} \theta(\mu)$

where $\theta(\mu) = \inf\{f(x) + \mu B(x) : g(x) < 0, x \in X\}$

If $\theta(\mu)$ is attained by x_μ, then the limit of any convergent

subsequence of $\{x_\mu\}$ is an optimal solution to the primal problem ~~and~~,

and $\mu B(x_\mu) \to 0$ as $\mu \to 0^+$.

(No proof - see text.)

<u>Computational Difficulties.</u>

As for penalty function methods, the Hessian of the auxiliary function approaches the Hessian of $L(\bar{x}, \bar{u})$, but with an additional term that has eigenvalues $\to \infty$ as $\mu \to 0$.

So steepest descent not good, but Newton-type methods work quite well.

ALGO 9H.

Primal-Dual Path Following Method

~~An int~~ A nonlinear programming algorithm for linear programming.

$$\begin{array}{llll} \min & c^T x & & \\ \text{s.t.} & Ax = b & (P) \\ & x \geq 0 & \end{array} \qquad \begin{array}{lll} \max & b^T y & \\ \text{s.t.} & A^T y + z = c & (D) \\ & z \geq 0. & \end{array}$$

Barrier subproblem:

$$\min \{ \underbrace{c^T x - \mu \sum \ln(x_i)}_{\text{convex}} : Ax = b, (x > 0) \}.$$

KKT conditions:

$$\begin{aligned} c - \mu X^{-1} e - A^T y &= 0 \\ \cancel{Ax = b} \quad Ax &= b \end{aligned} \qquad \text{where } X = \begin{bmatrix} x_1 & & \\ & \ddots & \\ & & x_n \end{bmatrix}, X^{-1} = \begin{bmatrix} \frac{1}{x_1} & & \\ & \ddots & \\ & & \frac{1}{x_n} \end{bmatrix}.$$

Let $z = \mu X^{-1} e$

Get equivalent conditions:

$$\begin{aligned} Ax &= b \\ A^T y + z &= c \\ Xz &= \mu e, \text{ or } x_i s_i = \mu \quad \forall i. \end{aligned}$$

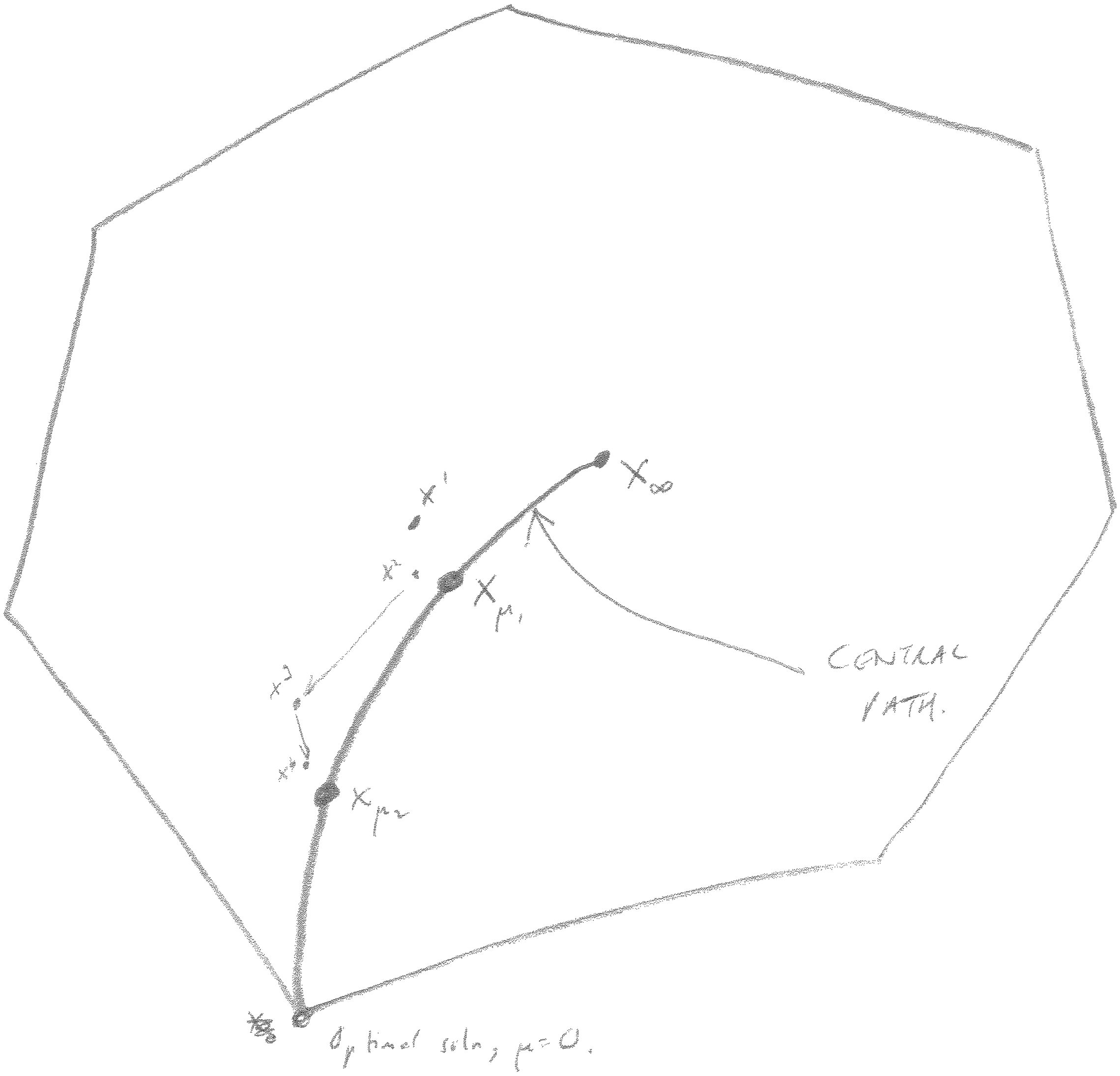
Thus, if $\mu = 0$, get usual primal feas, dual feas, comp slackness.

Let (x_μ, y_μ, z_μ) denote solution to KKT conditions.

As $\mu \to 0$, get $(x_\mu, y_\mu, z_\mu) \to$ optimal soln to (P) and (D).

Duality gap: $c^T x - b^T y = (A^T y + z)^T x - (Ax)^T y$

$$= z^T x = \sum_{i=1}^{n} z_i x_i = n\mu.$$



Suffices to solve barrier subproblem approximately — "carrot and donkey".

"Approximately" means: $Ax = b,\ A^T y + z = c,\ \|Xz - \mu e\| \le \theta\mu$ for some $\theta \in (0,1)$

Can solve with, eg, $\theta = 0.35$

Then replace μ by $\hat{\mu} = \beta\mu$, with $\beta = 1 - \frac{\delta}{\sqrt{n}}$ (with $\delta = 0.35$, for example).

Then $\|Xz - \hat{\mu} e\| \le \hat{\mu}$, and get convergence to ~~soln to subproblem~~ a point with $\|Xz - \hat{\mu} e\| \le \theta\hat{\mu}$ in one Newton step.

Choosing β smaller, eg, $\beta = 0.1$, may require more Newton steps to get ~~back~~ to an approximate solution for next μ.

Can show get convergence in $O(\sqrt{n})$ iterations if $\beta = 1 - \frac{\delta}{\sqrt{n}}$,

or $O(n)$ iterations if $\beta = 0.9$, say.

Can modify algorithm to get superlinear or quadratic convergence.

Calculation of the Newton step requires solving:

$$(*) \quad \begin{bmatrix} A & & \\ & A^T & I \\ \bar{Z} & & \bar{X} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \mu e - \bar{X}\bar{z} \end{bmatrix} \begin{array}{l} \leftarrow \text{stay primal feasible} \\ \leftarrow \text{stay dual feasible} \\ \leftarrow \text{linearization of } XZe = \mu e \end{array}$$

where $\bar{x}, \bar{y}, \bar{z}$ is current iterate.

Last line reads: $\bar{Z}\Delta x + \bar{X}\Delta z = \mu e - \bar{X}\bar{z}$

This is a linearization of

$$(\bar{X} + \Delta X)(\bar{Z} + \Delta Z)e = \mu e$$

$$\text{ie,} \quad \bar{X}\Delta z + \bar{Z}\Delta x + \underbrace{\cancel{\Delta X \Delta Z e}}_{\text{drop since quadratic.}} = \mu e - \bar{X}\bar{z}$$

$(*)$ is a full rank ~~system of linear~~ square system of linear equalities,

so can be solved by, eg, Gaussian elimination, ...

Interior point methods for NLP:

LOQO, KNITRO.

Can work with equality constraints.

To ease presentation, consider

$$\min \; f(x) \qquad f: \mathbb{R}^n \to \mathbb{R}$$

$$\text{s.t.} \; g(x) \le 0 \qquad g: \mathbb{R}^n \to \mathbb{R}^m \qquad f, g \text{ smooth.}$$

Barrier problem: $\min \; f(x) - \mu \sum \ln s_i$

$$\text{s.t.} \; g_i(x) + s_i = 0 \qquad i = 1, \dots, m$$
$$s \ge 0$$

Optimality conditions:

$$\nabla f + A(x)^T y = 0 \quad (1)$$
$$-\mu S^{-1} e + Ye = 0 \quad (2)$$
$$g(x) + s = 0 \quad (3) \quad (\text{and } s \ge 0)$$

where $A(x)$ is Jacobian of $g(x)$,

and y are KKT multipliers

Replace (2) by $-\mu e + SYe = 0$

Applying Newton's Method to the system of equations:

Choose direction $(\Delta x, \Delta s, \Delta y)$ satisfying:

$$ -\nabla f - A(x)^T y$$

$$\begin{bmatrix} \nabla^2_{xx} L(x,s,y) & 0 & A(x)^T \\ 0 & Y & S \\ A(x) & I & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta s \\ \Delta y \end{bmatrix} = \begin{bmatrix} \cancel{-\nabla_x L(x,s,y)} \\ \mu e - SYe \\ -g(x) - s \end{bmatrix}$$

Can eliminate Δs from this equation,

so $\cancel{\Delta s = -Y^{-1} S(\Delta y}$

$$\Delta s = -Y^{-1} S \Delta y + Y^{-1}(\mu e - SYe)$$

and then solve a system of the form:

$$\begin{bmatrix} \nabla^2_{xx} L(x,s,y) & A(x)^T \\ A(x) & -Y^{-1} S \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} / \\ \mathcal{D} \end{bmatrix}$$

for appropriate vectors $/, \mathcal{D}$.

ALGO 93c

Need a line search to pick step length.

Approach must be modified to cope with non-convexity

and to prevent convergence to non-stationary points.

Can be done either through a trust region approach,

or by modifying the primal-dual matrix.

For more details, see papers by Vanderbei et al,
and by Nocedal et al.

Successive Quadratic Programming (SQP) Methods.

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & h_j(x) = 0 \quad j = 1, \ldots, p \end{aligned}$$

All functions twice continuously differentiable.

KKT conditions:

$$\left.\begin{aligned} \nabla f(x) + \sum_{j=1}^{p} v_j \nabla h_j(x) &= 0 \\ h_j(x) &= 0 \quad j = 1, \ldots, p. \end{aligned}\right\} \quad \text{Write: } W(x, v) = 0$$

Use Newton's method to find a root of this system of equations:

Given an iterate (x^k, v^k), choose new iterate by solving

$$W(x^k, v^k) + \nabla W(x^k, v^k) \begin{bmatrix} x - x_k \\ v - v_k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Now,

$$\nabla W(x, v) = \begin{bmatrix} \nabla^2 f(x) + \sum v_j \nabla^2 h_j(x) & \nabla h_1(x) \ldots \nabla h_p(x) \\ \nabla h_1^T(x) & \\ \vdots & 0 \\ \nabla h_p^T(x) & \end{bmatrix}$$

So we have, letting $d = x - x^k$:

$$\left(\nabla^2 f(x) + \sum v_j \nabla^2 h_j(x)\right) d + \sum (v_j - \cancel{v_{k_j}}) \nabla h_j(x) = - \nabla f(x^k) - \cancel{\sum_{j=1}^{p} v_{k_j} \nabla h_j(x)}$$

$$\nabla h_j^T(x^k)\, d = \cancel{0}\, -h_j(x^k) \quad j = 1, \dots, p$$

or:

$$\left.\begin{array}{ll} n \text{ equations} & \nabla^2 L(x^k, v^k)\, d + \sum v_j \nabla h_j(x) = -\nabla f(x^k) \\ p \text{ equations} & \underbrace{\nabla h_j^T(x^k)\, d}_{\substack{n \text{ variables} \\ d_1, \dots, d_n}} \qquad \underbrace{\qquad\qquad}_{\substack{p \text{ variables} \\ v_1, \dots, v_p}} = \cancel{0}\, -h_j(x^k),\ j = 1, \dots, p \end{array}\right.$$

Square system of equations. Near optimum, will have $d^T \nabla^2 L(x^k, v^k) d \geq 0$ for directions d satisfying $\nabla h_j^T(x) d = 0$, by second order conditions.

Can solve the system to find the solution.

Can also think of the system as the optimality conditions for the QP:

$$\min \quad f(x^k) + \nabla f(x^k)^T d + \tfrac{1}{2} d^T \nabla^2 L(x^k, v^k)\, d \qquad QP(x^k, v^k)$$

$$\text{s.t.} \quad h_j(x^k) + \underbrace{\nabla h_j^T(x)\, d}_{\text{linearization of constraints}} = 0$$

ALGORITHM

Initialization: Set $k=1$, Choose (x^1, v^1).

Main Step: Solve $QP(x^k, v^k)$, giving d^k and multipliers v^{k+1}.

If $d^k = 0$, then STOP: x^k is a KKT point with multipliers v^{k+1}.

Else, set $x^{k+1} = x^k + d^k$, set $k \leftarrow k+1$, repeat main step.

Algorithm converges quadratically once it is close to optimum.

If have ineq constraints:

$$\begin{aligned} \min\ & f(x) \\ \text{s.t. } & g_i(x) \le 0 \quad i=1,\dots,m \\ & h_j(x) = 0 \quad j=1,\dots,p \end{aligned}$$

use $QP(x^k, u^k, v^k)$:

$$\min\ f(x^k) + \nabla f(x^k)^T d + \tfrac{1}{2} d^T \nabla^2 L(x_k, u_k, v_k)\, d$$

$$\text{s.t.}\quad g_i(x^k) + \nabla g_i(x^k)^T d \le 0 \quad i=1,\dots,m$$

$$h_j(x^k) + \nabla h_j(x^k)^T d = 0 \quad j=1,\dots,m$$

It's harder to solve a QP with ineq constraints, can't just factor a matrix. Can use, eg, interior pt methods, or active set methods (see later)

For iterates far from optimality:

Use <u>penalty function</u> or trust region.

Common to use either l_1 or augmented Lagrangian penalty function.

Can then get global convergence.

Use linesearch to pick step length, with QP subproblem providing the direction.

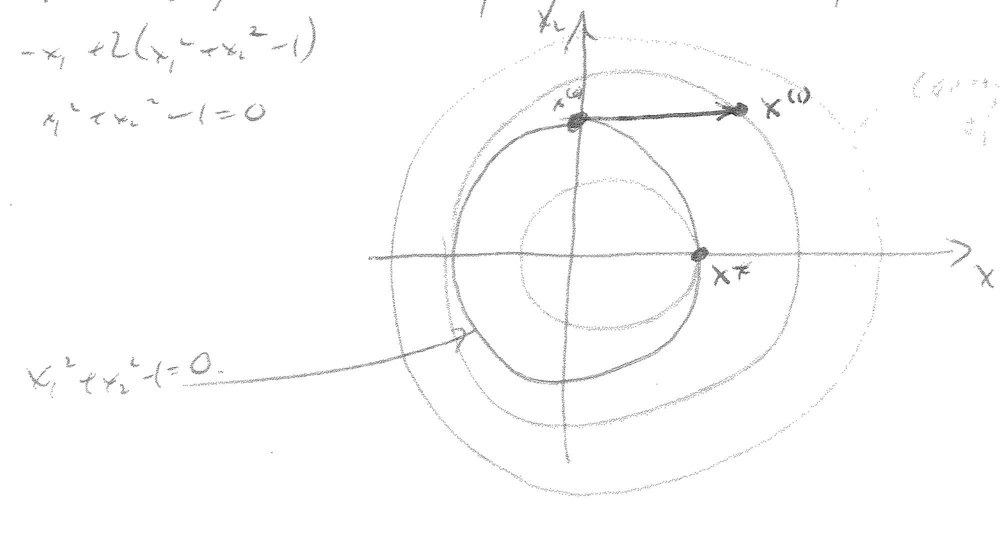
In practice, a very successful method.

Need to be careful of the Maratos effect:

The full step d^k may lead to an increase in both $f(x)$ and the merit function, so it will be rejected by a line search. This can stop the algorithm achieving quadratic convergence.

e.g.: $\min -x_1 + 2(x_1^2 + x_2^2 - 1)$

s.t. $x_1^2 + x_2^2 - 1 = 0$



Reduced Gradient Method

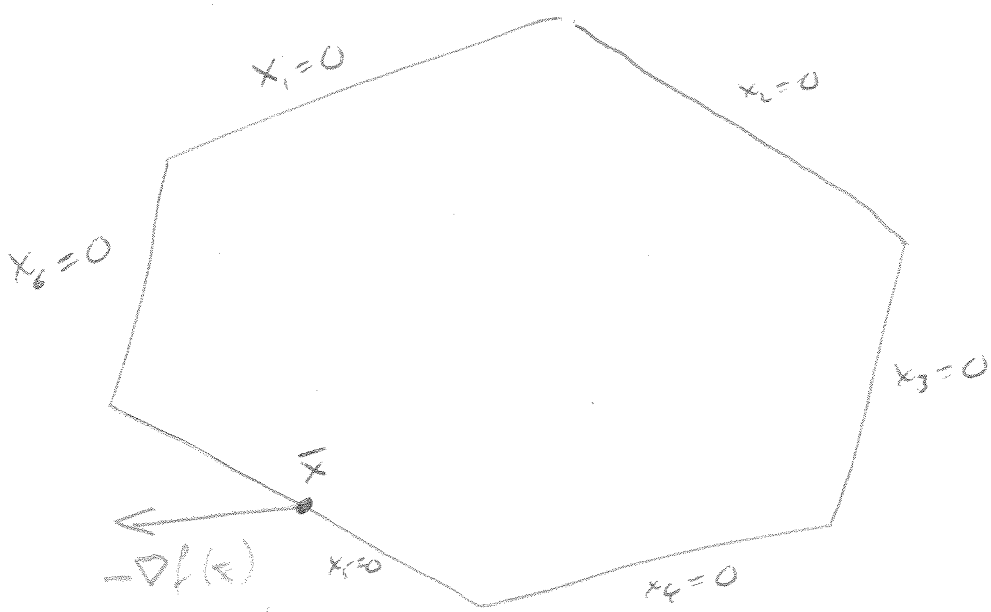
$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

A is $m \times n$. (discuss nonlinear constraints later)

Nondegeneracy assumption:

- Any m columns of A are linearly independent.
- Every extreme point has exactly m positive variables.

Choosing a direction: Modify ∇f.



$n - m = 2$, by nondegeneracy assumption.

Divide $\bar{x}$ into a set of m basic variables, all > 0
and a set of $n-m$ nonbasic variables, either $= 0$ or > 0.

NOTE.

Usually take largest m components to be basic.

Split up $A = [B, N]$, split ∇f up $(\nabla f_B(\bar{x}), \nabla f_N(\bar{x}))$.

Eg: $\underbrace{x_1, x_2, x_3, x_4}_{>0}$ basic, $\underset{>0}{x_5}$, $\underset{=0}{x_6}$ nonbasic

Soln

As with simplex, consider ~~direction~~ "reduced cost",

$$r_N := \nabla f_N^T(\bar{x}) - \nabla f_B^T(\bar{x}) B^{-1} N$$

$$c_N^T - c_B^T B^{-1} N$$

Choosing a direction d_N and then setting $d_B = -B^{-1} N d_N$

results in ~~this change~~ the change $\left(\nabla f_N^T(\bar{x}) - \nabla f_B^T(\bar{x}) B^{-1} N\right) d_N$

in the linearization of the objective.

To get decrease: want $d_{N_\xi}^T r_{N_\xi} < 0$. ~~for each i~~

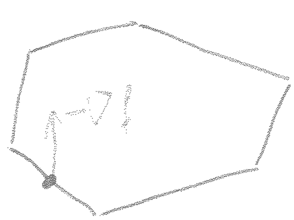
So choose:

If $r_i \leq 0$: set $d_i = -r_i$

If $r_i > 0$: set $d_i = -x_i r_i$

In example, get r_5 and r_6 both positive,
so set $d_5 = 0$, $d_6 = -x_6 r_6$

Can move off a constraint:



Line search to find best step length, ensuring that stay feasible.

Have:

- direction is ~~ze~~ nonzero and improving, unless $\bar{x}$ is a KKT point.
- Algorithm converges to a KKT point.

GENERALIZED REDUCED GRADIENT METHOD (GRG)

Generalize to general equality constraints:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & h(x) = 0 \qquad m \text{ equality constraints.} \\ & x \geq 0 \end{aligned}$$

Linearize constraints as $h(\bar{x}) + Dh(\bar{x})(x - \bar{x}) = 0.$

$h(\bar{x})$: m vector

$Dh(\bar{x})$: $m \times n$ Jacobian.

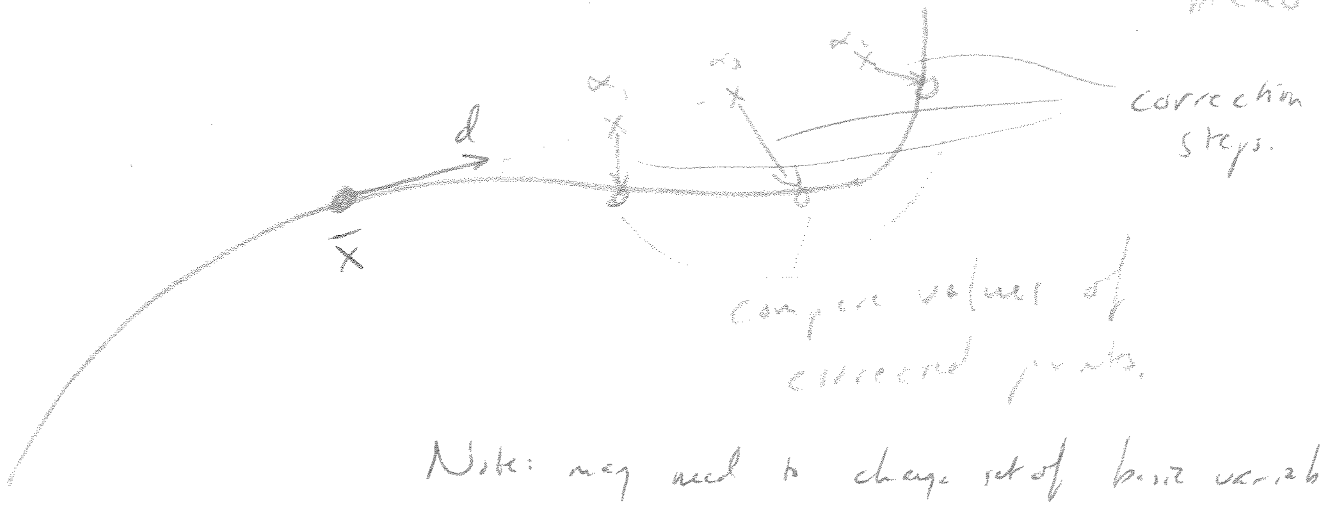
$$(Dh(\bar{x}))_{ij} = \frac{\partial h_i}{\partial x_j}$$

Assume $\bar{x}$ is feasible so $h(\bar{x}) = 0$.

Set up d as before with now $A \equiv Dh(\bar{x})$, $b \equiv Dh(\bar{x})\bar{x}$

Get a new feasible point by linesearching in direction d, and for each choice of α trying to find a feasible point using only the basic variables. ~~This may require~~

See picture on next page:



Note: may need to change set of basic variables in correction step because some basic variable becomes negative.

Provides a pretty robust & efficient algo for NLPs.

<u>Further refinements:</u>

- Superbasic variables:

x_B, x_S, x_N

x_B: as before

x_S: usually larger nonbasic variables

x_N: division of all x_N.

Only allow superbasic (and hence basic) variables to vary. x_N constant. Variables can move from one set to another each iteration.

Purpose is to stop small steps and "jamming", because too many small variables changing.

- Second order terms:

Choose d to minimize

$$f(\bar{x}) + \nabla f(\bar{x})^T d + \tfrac{1}{2} d^T \nabla^2 f(\bar{x}) d$$

$$\text{s.t.} \quad A d = 0$$

with $d = (d_B, d_S, d_{N'})$ and $d_{N'} = 0$.

Have $d_B = -B^{-1} S d_S$, where $A = [B \quad S \quad N']$.

So $d = \begin{bmatrix} -B^{-1}S \\ I \\ 0 \end{bmatrix} d_S$, so choose d_S to solve:

$$\min \; f(\bar{x}) + \nabla f(\bar{x})^T \begin{bmatrix} -B^{-1}S \\ I \\ 0 \end{bmatrix} d_S$$

$$\cancel{+ \tfrac{1}{2} d_S^T [-B^{-1}S}$$

$$+ \tfrac{1}{2} d_S^T \underbrace{\begin{bmatrix} -S^T B^{-T} & I & 0 \end{bmatrix} \nabla^2 f(\bar{x}) \begin{bmatrix} -B^{-1}S \\ I \\ 0 \end{bmatrix}}_{\text{PROJECTED HESSIAN matrix.}} d_S$$

Usually use Q-N approximation to projected Hessian.

Can get superlinear convergence.

Active set methods for QP:

$$\begin{aligned} \min \quad & \tfrac{1}{2} x^T Q x + c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0. \end{aligned}$$

These are reduced gradient methods, where we divide the variables into three sets again $[x_B, x_S, x_{N'}]$

where now $x_{N'} = 0$, and we keep $x_{N'} = 0$.

Simplex is in this framework: x_S consists of just one variable, namely the entering variable.

The choice of x_S determines x_B.

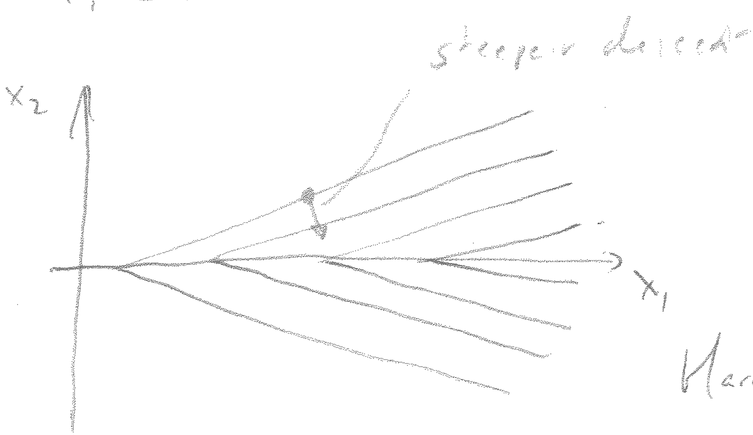
Nonsmooth Optimization

Gradient functions have discontinuities. Eg. l_1-penalty function.

Eg: $\min \; -x_1 + 100|x_2|$
s.t. $0 \le x_1 \le 10.$

Optimal soln: $x_1 = 10$, $x_2 = 0$.

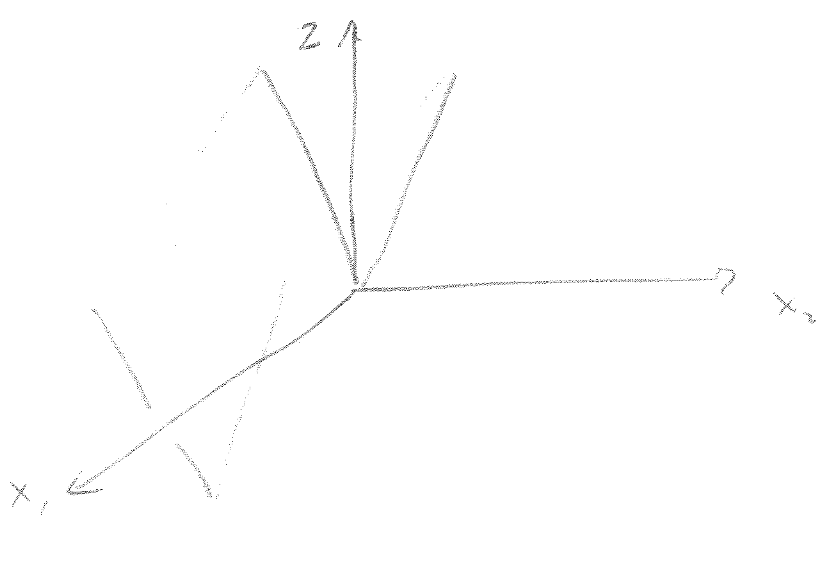
Contours:



Hard to choose good direction.
Hard to ~~cho~~ find good step length algebraically.

$$\nabla f = \begin{cases} \begin{bmatrix} -1 \\ +100 \end{bmatrix} & \text{if } x_2 > 0 \\ \begin{bmatrix} -1 \\ -100 \end{bmatrix} & \text{if } x_2 < 0 \end{cases}$$

Graph of function:



More involved example:

$$\max_{y,S} \left(b^T y + \lambda_{min}(S)\right)$$
s.t. $\sum A_i y_i + S = C$

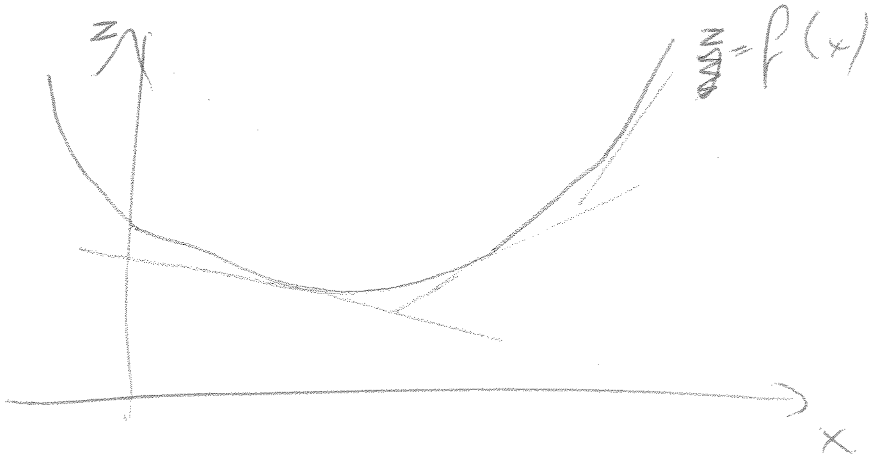
A_i, C, S square symmetric matrices.

y vector

$$\min \; f(x) \qquad \text{(NSP)}$$
$$\text{s.t.} \; g_i(x) \le 0 \quad i=1,\dots,m.$$

Assume functions are convex, ~~both~~ continuous, but may not be differentiable

Can approximate using LP:



~~At a point $\bar{x}$:~~

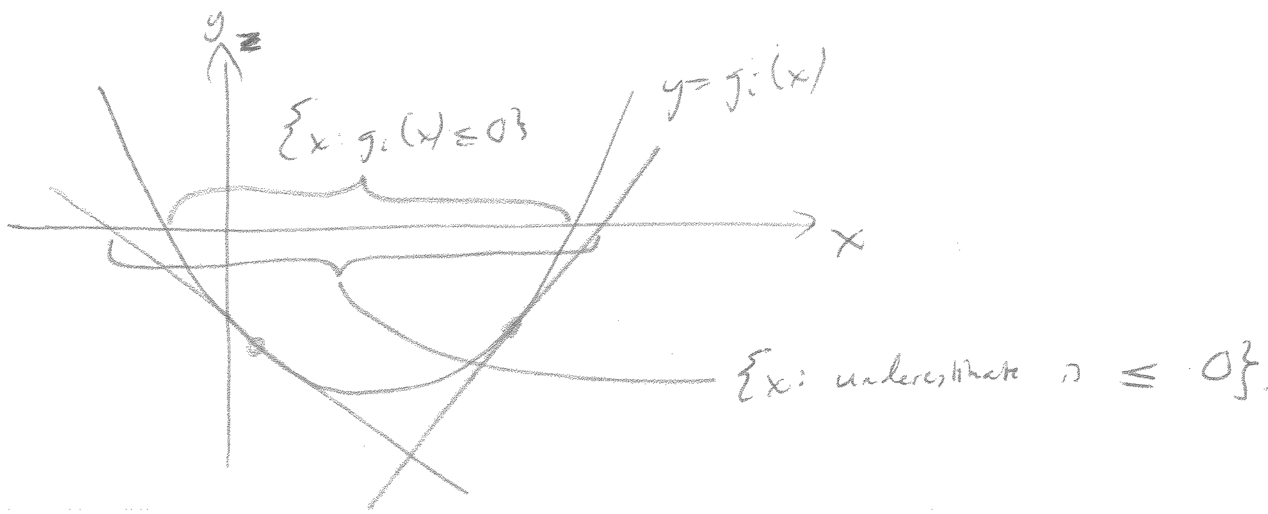
Rewrite problem as

$$\min \; z$$
$$\text{s.t.} \; f(x) - z \le 0$$
$$g_i(x) \le 0 \quad i=1,\dots,m$$

Approximate constraints from underneath: (as with Kelley's cutting plane algorithm)

$$\cancel{f(x)} \; g_i(x) \ge g_i(\bar{x}) + \xi^T(x-\bar{x})$$

ξ is a subgradient of g_i at $\bar{x}$.



Can underestimate $f(x) - z$ similarly.

So get LP:

$$\text{(LP)} \quad \min \ z \quad \text{s.t. } Ax + hz \le b$$

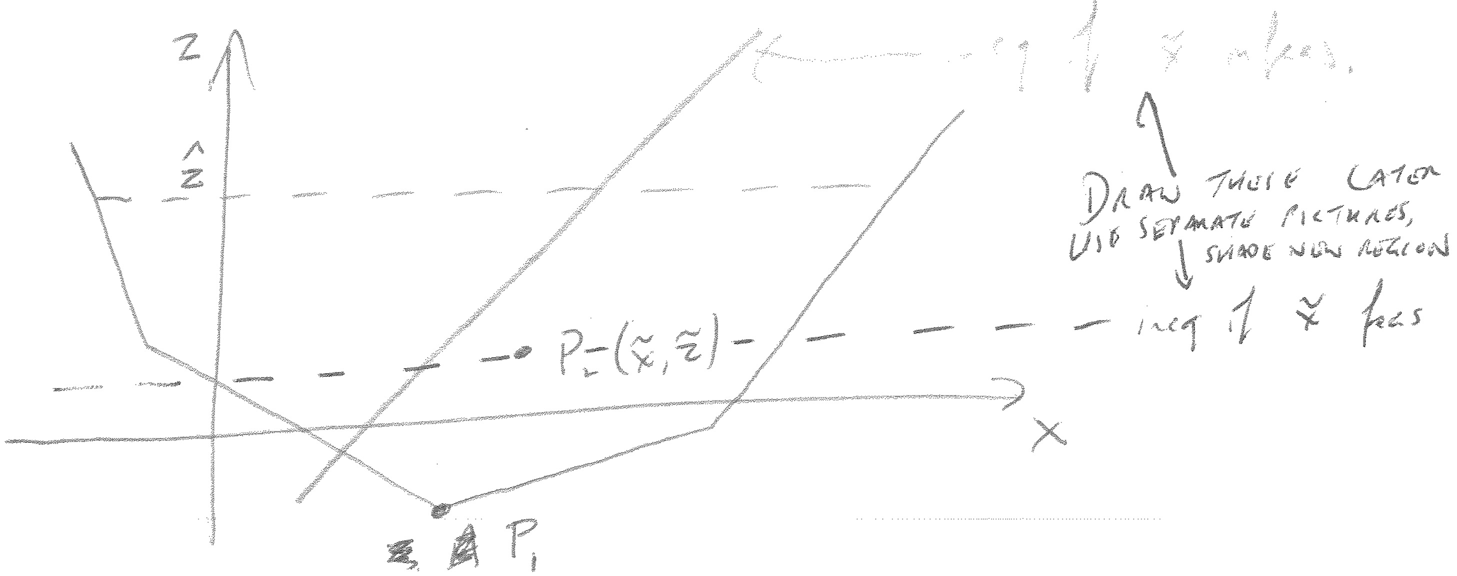
relaxation of (NSP)

for appropriate A, h, b, generated from supporting inequalities.

Solve (LP) using an algorithm for linear programming. Can use simplex, or an interior point method.

Note that any feasible point $\hat{x}$ gives an upper bound $\hat{z} = f(\hat{x})$ on optimal value of (NSP).

If use an interior point method, best not to solve problem exactly:



If solve ~~exactly~~ (LP) exactly, get (P_1). But hard to ~~solve~~ restart an interior point method from an extreme point.
So find analytic center (or a good point in the middle), P_2

Let P_2 be $(\tilde{x}, \tilde{z})$.

~~If $\tilde{x}$ infeasible, add~~
If $\tilde{x}$ violates $g_i(x) \leq 0$, add supporting ineq for this constraint and resolve.

If $\tilde{x}$ feasible, add constraint $z \leq \tilde{z}$ and find new center

Algo finds a solution within ε of optimality in $O(n/\varepsilon)$ iterations.

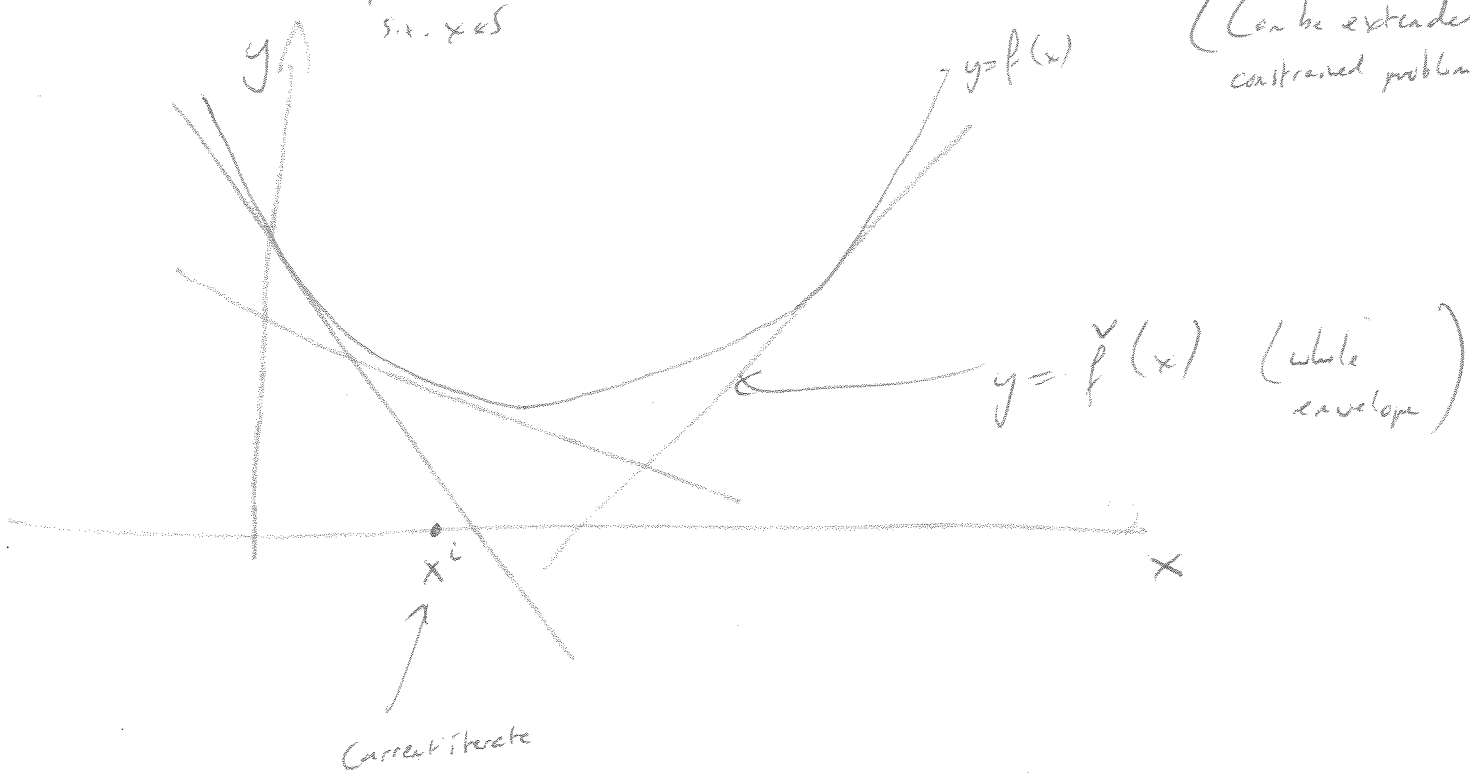
If also allow dropping of constraints, can get within ε of optimality in $O(n \ln(\frac{1}{\varepsilon}))$ iterations.

Only need a handful of iterations to find new center after adding a cut
(Can add multiple cuts at once.

BUNDLE METHODS

$$\min_{\text{s.t. } x \in S} f(x)$$ where f is convex, continuous, nonsmooth.

(Can be extended to constrained problems.)



Have upper bound: $\min f(x^j)$, $j = 1, \ldots, i$.
ie, value of best point seen to date.

Have lower bound: min value of piecewise $\underbrace{\text{linear underestimator}}$ of f.

~~It can be very~~

This is a "bundle of information" provided by earlier iterates.

It can be very slow to move directly to the ~~solu~~ minimizer of the underestimator.
So use "proximal" term, $\|x - x^i\|^2$. ~~where x^i~~

So: $\min \quad z + \frac{1}{2}\mu \|x - x^i\|^2$ (QP)

s.t. $\quad z \geq$ underestimator function, $\check{f}(x)$.

Then we don't move too far.

Let $\bar{x}$ be the point that solves this quadratic problem (QP).

If $f(\bar{x})$ is too large compared to z (so $\check{f}$ is not a good approximation to f),

REJECT the step, set $x^{i+1} = x^i$, update $\check{f}$ with a constraint from $\bar{x}$ and repeat

Else,

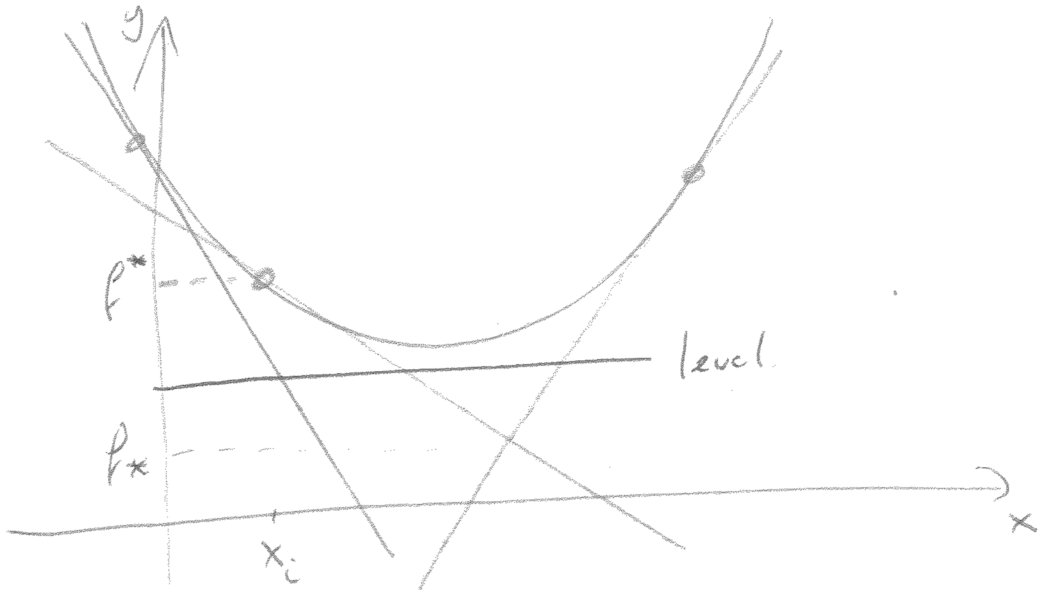
Set x^{i+1} ~~[illegible]~~ $= \bar{x}$, update $\check{f}$ with a constraint from $\bar{x}$, and repeat.

Repeat until upper and lower bounds are close enough.

Difficulty: hard to choose μ. Should probably be dynamically updated. Nonetheless, algorithm works well.

References:

Proximal level bundle methods:



Optimal value is between f_* and f^*.

Look for proximal point with z-value $\leq \lambda f_* + (1-\lambda) f^*$ in the model, $\check{f}$.

So: $\min \frac{1}{2} \|x - x^i\|^2$

s.t. $z \leq \lambda f_* + (1-\lambda) f^*$

$z \geq \check{f}$ underestimator of f, as a function in x.

i.e. $z \geq f(x^j) + \xi^T (x - x^j)$

Typically, get convergence in $O(\frac{K}{\varepsilon})$ iterations, where ε = desired accuracy, K = diameter of space = $\max\{\|x-y\| : x, y \in S\}$
Work well in practice.