- Equality constrained minimization:

$$\min_x f(x) \quad \text{s.t.} \quad Ax = b$$

$$\Rightarrow \boxed{Ax^* = b \quad \text{and} \quad \nabla f(x^*) + A^T v^* = 0}$$

- special case:

$$\min \ \frac{1}{2} x^T P x + q^T x + r \quad \text{s.t.} \quad Ax = b \quad \Rightarrow \underbrace{\begin{bmatrix} P & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x^* \\ v^* \end{bmatrix} = \begin{bmatrix} -q \\ b \end{bmatrix}}_{\text{KKT}}$$

- Elimination Technique:

$$\{x \mid Ax = b\} = \{x \mid x = Fz + \underbrace{\hat{x}}_{\text{particular solution}} \text{ for some } z\}$$

$$\Rightarrow \underbrace{\begin{array}{l} \min_x f(x) \\ \text{s.t. } Ax = b \end{array}}_{\text{Constrained}} \iff \underbrace{\min_z f(Fz + \hat{x})}_{\text{Unconstrained}}$$

- Newton's method with equality constraints:

Question: Given $x^{(k)}$ satisfying $Ax^{(k)} = b$, how to find $\underbrace{\Delta x}_{\text{Newton step}}$?

KKT: $A(x^{(k)} + \Delta x) = b$ and $\underbrace{\nabla f(x^{(k)} + \Delta x)}_{\simeq \nabla f(x^{(k)}) + \nabla^2 f(x^{(k)}) \Delta x} + A^T v = 0$

$$\Rightarrow \begin{bmatrix} \nabla^2 f(x^{(k)}) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ v \end{bmatrix} = \begin{bmatrix} -\nabla f(x^{(k)}) \\ 0 \end{bmatrix}$$

- Note: If $x^{(k)}$ is feasible and $A\Delta x = 0$, then $x^{(k+1)} = x^{(k)} + \alpha \Delta x$ is feasible for all α's. $\Rightarrow$ At each iteration, we find a feasible point. $\Rightarrow$ $\boxed{\text{Feasible descent direction.}}$

- Infeasible start Newton method:

$x^{(k)}$ may not satisfy $Ax^{(k)} = b$.

$$\Rightarrow \begin{bmatrix} \nabla^2 f(x^{(k)}) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ v \end{bmatrix} = - \begin{bmatrix} \nabla f(x^{(k)}) \\ Ax^{(k)} - b \end{bmatrix}$$

$\Rightarrow$ $(x^{(k)} + \Delta x)$ is feasible but $x^{(k)} + t\Delta x$ may be infeasible.

(previous idea: choose t such that $f(x^{(k+1)}) < f(x^{(k)})$)

- Primal-dual Newton step: iterate on both primal and dual parameters: $(x^{(k)}, v^{(k)}) \longrightarrow (x^{(k+1)}, v^{(k+1)})$

$$A(x^{(k)} + \Delta x) = b \quad , \quad \nabla f(x^{(k)} + \Delta x) + A^T(v^{(k)} + \Delta v) = 0$$

$$\Rightarrow \begin{bmatrix} \nabla^2 f(x^{(k)}) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta v \end{bmatrix} = - \begin{bmatrix} \nabla f(x^{(k)}) + A^T v^{(k)} \\ Ax^{(k)} - b \end{bmatrix}$$

- Measure of optimality: $\underset{\text{residual}}{r}(x, v) = [Ax - b, \nabla f(x) + A^T v]$

- Step size: choose t such that $\| r(x^{(k)} + t\Delta x, v^{(k)} + t\Delta v) \|$ is minimum.

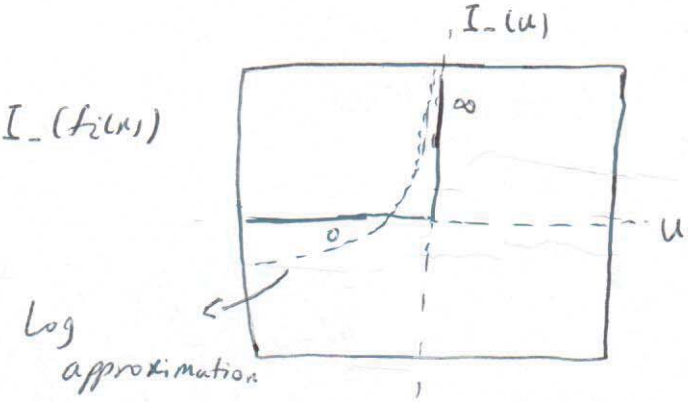
- step size is based on residual, instead on $f(x)$.
  $(f(x^{(k+1)}) < f(x^{(k)}))$

- Interior point method:

$$\begin{aligned} \min \quad & f_0(x) \\ \text{s.t.} \quad & f_i(x) \le 0 \\ & Ax = b \end{aligned} \quad \Longrightarrow \quad \begin{aligned} & - Ax^* = b, \quad f_i(x^*) \le 0 \\ & - \lambda^* \ge 0 \\ & - \nabla f_0(x^*) + \sum_i \lambda_i^* \nabla f_i(x^*) + A^T \nu^* = 0 \\ & - \lambda_i^* f_i(x^*) = 0 \end{aligned}$$

$$\Updownarrow$$

$$\begin{aligned} \min_x \quad & f_0(x) + \sum_i I_-(f_i(x)) \\ \text{s.t.} \quad & Ax = b \end{aligned}$$



$\longrightarrow \quad I_-(u) \simeq \frac{-1}{t} \log(-t)$

Approximation $\longrightarrow$

$$\begin{aligned} \min_x \quad & f_0(x) \boxed{- \sum_{i=1}^{m} \frac{1}{t} \log(-f_i(x))} \\ \text{s.t.} \quad & Ax = b \end{aligned}$$

$\longrightarrow$ log barrier

- Quality of approximation improves as $\underline{t}$ grows.

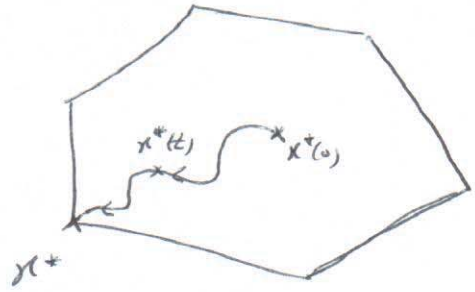
- Define: $\phi(x) = - \sum_{i=1}^{m} \log(-f_i(x))$

$$\Rightarrow \quad \begin{aligned} \min_x \quad & t f_0(x) + \phi(x) \\ \text{s.t.} \quad & Ax = b \end{aligned} \quad \longrightarrow \quad \text{solution } x^*(t)$$

- Thm: $x^*(t)$ satisfies modified KKT where $\lambda_i f_i(x)$ is replaced by $\lambda_i f_i(x) = \frac{1}{t}$.

- Two observations:
  1- for every t, $x^*(t)$ is feasible for original opt.
  2 As $t \to \infty$, $x^*(t)$ goes to x^*.

$\Rightarrow$ we have central path:



<u>Thm:</u> $\boxed{f_0(x^*(t)) - p^* \leq \frac{m}{t}}$ (m : # of constraints)

(because in the dual: $\sum_{i=1}^{m} \lambda_i^*(t) f_i(x^*(t)) = \frac{m}{t}$

$\Rightarrow$ Given t, we know the quality of approximation.

Algorithm: - Given strictly feasible x, $t^{(0)} > 0$ and tolerance $\varepsilon > 0$.

- Repeat:
  - Centering step:
    Compute $x^*(t)$ by $\min_{\substack{x \\ Ax=b}} t f_0 + \phi$ by starting at $\underline{x}$.
  - Update: $x = x^*(t)$
  - stopping criterion: quit if $\frac{m}{t} < \varepsilon$
  - Increase t: $t = t\mu$ (for some $\mu > 1$)

Comment: - Finding $x^*(t)$ accurately is not important.
- Finding $x^*(t)$ needs a few iterations given its initialization.

- Question: How to find a strictly feasible x to start with?

- Phase I: solve an optimization to find x:

$$\min_{x,s} \; s$$
$$\text{s.t. } f_i(x) \le s$$
$$Ax = b$$

$\longrightarrow$ - Choose x and a big s to make to feasible

- Use barrier method to find an x corresponding to $s<0$.

- Summary: we have inner and outer loop.
  - inner loop finds a point on the central path.
  - outer loop updates t.

- Question: Can we merge these inner and outer iterations?

Modified KKT:

$$\boxed{\begin{aligned} &\nabla f_0(x) + J(f(x))^T \lambda + A^T v = 0 \\ &-\operatorname{diag}(\lambda) f(x) - \frac{1}{t} = 0 \\ &Ax - b = 0 \end{aligned}}$$

, $J(f(x)) =$ Jacobian

Define $y = \begin{bmatrix} x \\ \lambda \\ v \end{bmatrix}$. $\Rightarrow$ Iterate on y, ie on both primal and dual parameters.

- Direction update: find Δy such that $y + \Delta y$ satisfies Linearized modified KKT.

- step size update: find α such that $y + \alpha \Delta y$ gives the smallest residual for the left side of KKT.

- Determine t: increase it by a factor of $\mu > 1$.

- Note: step size should be chosen to gaurantee $\lambda > 0$, $f_i(x) < 0$.

"Distributed computation"

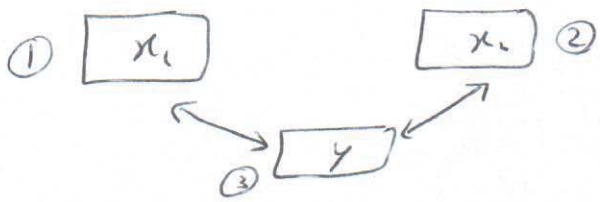
- $\min_{x_1, x_2} f_1(x_1) + f_2(x_2) \Rightarrow \nabla f_1(x_1) = 0$ and $\nabla f_2(x_2) = 0$

  "no coupling"

- $\min_{x_1, x_2, y} f_1(x_1, y) + f_2(x_2, y) \Rightarrow \nabla_{x_1} f_1(x_1, y) = 0,\ \nabla_{x_2} f_2(x_2, y) = 0$

  $\nabla_y f_1(x_1, y) + \nabla_y f_2(x, y) = 0$

- Consider three entities for x_1, x_2 and y.



- 3 propagates $y^{(k)}$.
- 1 solves $\min_{x_i} f_i(x_i, y^{(k)})$ for $i = 1, 2$.
- 1 returns $x_i^{(k)}$ to 3.
- 3 updates $y^{(k+1)}$ as $y^{(k)} - t \times (\nabla_y f_1(x_1^{(k)}, y^{(k)}) + \nabla_y f_2(x_2^{(k)}, y^{(k)}))$

This is called primal decomposition.

$\Rightarrow$ - The problem can be solved on multiple machines.

- Different entities may not know private information of each other (say 1 doesn't need to know $f_2(x, y)$).

- Another approach:

$$\min_{x_1, x_2, y} f_1(x_1, y) + f_2(x_2, y) \iff \min_{x_1, x_2, y_1, y_2} f_1(x_1, y_1) + f_2(x_2, y_2)$$

$$y_1 = y_2$$

Consistency constraint.

$\longrightarrow$ let λ be the dual param for $y_1 = y_2$.

$\Rightarrow$ Algorithm:
- ③ propagates $\lambda^{(k)}$.
- ① solves $\min_{x_1, y_1} f_1(x_1, y_1) + \lambda^{(k)} y_1$
- ② solves $\min_{x_2, y_2} f_2(x_2, y_2) - \lambda^{(k)} y_2$
- ⓘ returns $(x_i^{(k)}, y_i^{(k)})$ to ③ for $i = 1, 2$.
- ③ updates $\lambda^{(k+1)}$ as $\lambda^{(k)} + t(y_1 - y_2)$

This is called dual decomposition.

- More complicated example:

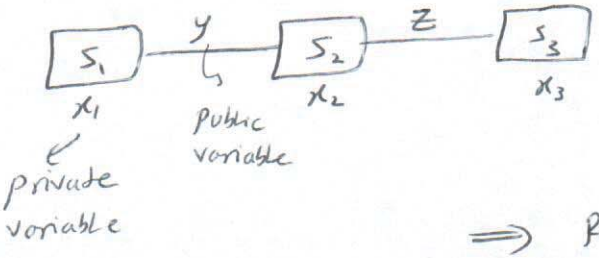
$$\min_{x_1, x_2} \quad f_1(x_1) + f_2(x_2)$$
$$h_1(x_1) + h_2(x_2) \leq 0$$

$\Rightarrow$ ③ propagates λ

ⓘ solves $\min_{x_i} f_i(x_i) + \lambda_i h_i(x_i)$ for $i = 1, 2$

③ updates λ based on $h_1(x_1) + h_2(x_2)$ at the new x_1 and x_2.

- General decomposition structure:
  - multiple subsystems
  - coupling constraints between subsystems
  - we represent the structure by a graph



$$\min f_1(x_1, y) + f_2(x_2, y, z) + f_3(x_3, z)$$
$$\text{s.t. } (x_1, y_1) \in C_1, \ (x_2, y, z) \in C_2, \ (x_3, z) \in C_3$$

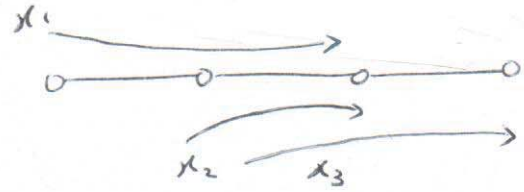
$\Rightarrow$ Replace y by y_1 and y_2 s.t. $y_1 = y_2$

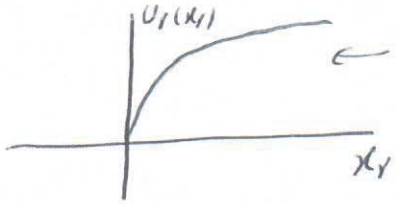
Replace z by z_2 and z_3 s.t. $z_2 = z_3$

Congestion control for Comm networks:

- There is a network of links with fixed capacities.
- Each user wants to transmit dat from a source to a destination over a fixed route.
- x_r: rate of user $\underline{r}$, y_l: capacity of link l,

  $U_r(x_r)$: utility of user $\underline{r}$.





← Concave and increasing

problem of interest:
$$\max_{x} \sum_{r \in S} U_r(x_r)$$
$$\sum_{r \in S(l)} x_r \le c_l \quad \forall l \in E$$

(S: set of users, E: set of links, $S(l)$: users using link l)

- This convex problem can be solved easily.
- challenge: there is no coordinator to collect the information and solve a centralized problem.
- Solution: propose a distributed algorithm.
- Define: $y_l = \sum_{r \in S(l)} x_r$, p_l: dual variable for $y_l \le c_l$

  $q_r = \sum_{l \in E(r)} p_l$ → sum of dual params over a route

  ($E(r)$: links of route)

$$\Rightarrow \text{KKT:} \begin{cases} U'(x_r) = q_r \\ p_l(y_l - c_l) = 0 \\ y_l - c_l \le 0 \\ p_l \ge 0 \end{cases}$$

strategy: - Each user updates x_r based on q_r
- Each link updates p_l based on y_l

p_l : price for using link l

$\Rightarrow$ - x_r is updated based on a route price $\sum_{l \in E(r)} p_l$

- p_l is updated based on the incoming rate $\sum_{r \in S(l)} x_r$

- Dual Algorithm:

$$\begin{cases} - \; x_r(t) = (U_r')^{-1}(q_r(t)) \\ - \; \dot{p}_l(t) = \alpha \left(y_l(t) - c_l\right)^+_{p(t)} \end{cases}$$

This is a fluid model as updates happen in the continuous domain t (time).

$$\left((U)^+_p = \begin{cases} U & p > 0 \\ \max(U, 0) & p = 0 \end{cases} \right)$$

Interpretation: If $y_l(t) > c_l \rightarrow p_l(t)$ increases

$\rightarrow q_r(t)$ increase

$\rightarrow x_r(t)$ decreases $\rightarrow y_l(t)$ decreases

If $y_l(t) < c_l \rightarrow x_r(t)$ tries to increase.

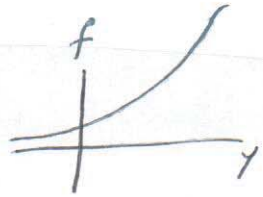
$\Rightarrow$ with global coordination, each user and link make decisions based on local information.

- Primal Algorithm:

$$\begin{cases} - \; \dot{x}_r(t) = \alpha \left(U_r'(x_r(t)) - q_r(t) \right) \\ - \; p_l(t) = f_l(y_l(t)) \end{cases}$$

dynamic

static



$y_l \uparrow \Rightarrow p_l \uparrow \Rightarrow q_r \uparrow \Rightarrow x_r \downarrow \Rightarrow y_l \downarrow$