CS711008Z Algorithm Design and Analysis

Lecture 8. Linear programming: interior point method

内点法

Dongbo Bu

Institute of Computing Technology
Chinese Academy of Sciences, Beijing, China

Outline

- Brief history of interior point method
- Basic idea of interior point method

A brief history of linear program

- In 1949, G. B. Dantzig proposed the simplex algorithm;
- In 1971, Klee and Minty gave a counter-example to show that simplex is not a polynomial-time algorithm.
- In 1975, L. V. Kantorovich and T. C. Koopmans, Nobel prize, application of linear programming in resource distribution;
- In 1979, L. G. Khanchian proposed a polynomial-time ellipsoid method;
- In 1984, N. Karmarkar proposed another polynomial-time interior-point method;
- In 2001, D. Spielman and S. Teng proposed smoothed complexity to prove the efficiency of simplex algorithm.

In 1979, L. G. Khanchian proposed a polynomial-time ellipsoid method for LP

Figure: Leonid G. Khanchian

In 1984, N. Karmarkar proposed a new polynomial-time algorithm for LP

Karmarkar at Bell Labs: an equation to find a new way through the maze

Folding the Perfect Corner

A young Bell scientist makes a major math breakthrough

Every day 1,200 American Airlines jets crisscross the U.S., Mexico, Canada and the Caribbean, stopping in 110 cities and bearing over 80,000 passengers. More than 4,000 pilots, copilots, flight personnel, maintenance workers and baggage carriers are shuffled among the flights; a total of 3.6 million gal. of high-octane fuel is burned. Nuts, bolts, altimeters, landing gears and the like must be checked at each destination. And while performing these scheduling gymnastics, the company must keep a close eye on costs, pro-

Indian-born mathematician at Bell Laboratories in Murray Hill, N.J., after only a years' work has cracked the puzzle of linear programming by devising a new algorithm, a step-by-step mathematical formula. He has translated the procedure into a program that should allow computers to track a greater combination of tasks than ever before and in a fraction of the time.

Unlike most advances in theoretical mathematics, Karmarkar's work will have an immediate and major impact on the real world

Basic idea of interior point method

KKT conditions for LP

- Let's consider a linear program in slack form and its dual problem, i.e.
  - Primal problem:

$$\begin{array}{rlcl} \min & c^T x & & \\ s.t. & Ax & = & b \\ & x & \geq & 0 \end{array}$$

拉格朗日乘子.

  - Dual problem:

$$\begin{array}{rlcl} \max & b^T y & & \\ s.t. & A^T y & \leq & c \end{array}$$

$(a_i^T y - c_i) x_i = 0$

- KKT condition:
  - Primal feasibility: $Ax = b$, $x \geq 0$.
  - Dual feasibility: $A^T y \leq c$
  - Complementary slackness: $x_i = 0$, or $a_i^T y = c_i$ for any $i = 1, ..., m$.

Rewriting KKT condition

- Let's consider a linear program in slack form and its dual problem, i.e.
  - Primal problem:

$$
\begin{array}{lrcl}
\min & c^T x & & \\
s.t. & Ax & = & b \\
 & x & \geq & 0
\end{array}
$$

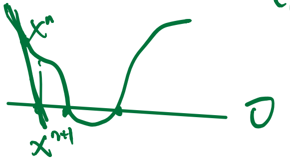
  - Dual problem:

$$
\begin{array}{lrcl}
\max & b^T y & & \\
s.t. & A^T y & \leq & c
\end{array}
$$

- KKT condition:
  - Primal feasibility: $Ax = b$, $x \geq 0$;
  - Dual feasibility: $A^T y + \sigma = c$, $\sigma \geq 0$;
  - Complementary slackness: $x_i \sigma_i = 0$.
- These conditions consists of $m + 2n$ equations over $m + 2n$ variables plus non-negative constraints.

$f(x)=0$

Newton method { ①求根 ②最优化.



$x^{n+1} = x^n - \frac{f(x_n)}{f'(x^n)}$

① 原始 $\min f_0(x)$ s.t. $f_i(x) \leq 0$

② $\min s$ s.t. $f_i(x) \leq s \Rightarrow$ { $s<0$: 找到了; $s=0$: 检验是否最优; $s>0$ 不可行

初始解 $x=(0,0,\dots,0)$, $s = \min f_i(0)$

找 x, y, σ 同时满足这些条件.

Rewrite KKT conditions further

- Let's define diagonal matrix:

$$X = \begin{pmatrix} x_1 & 0 & 0 \\ 0 & x_2 & 0 \\ 0 & 0 & x_3 \end{pmatrix}, \Sigma = \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{pmatrix}$$

and rewrite the complementary slackness as:

$$X\Sigma e = \begin{pmatrix} x_1\sigma_1 & 0 & 0 \\ 0 & x_2\sigma_2 & 0 \\ 0 & 0 & x_3\sigma_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} x_1\sigma_1 \\ x_2\sigma_2 \\ x_3\sigma_3 \end{pmatrix}$$

- KKT condition:
  - Primal feasibility: $Ax = b,\ x \geq 0$;
  - Dual feasibility: $A^T y + \sigma = c,\ \sigma \geq 0$;
  - Complementary slackness: $X\Sigma e = 0$. 二次约束 ⇒ 改进 $X\Sigma e = t$
- We call (x, y, σ) **interior point** if $x > 0$ and $\sigma > 0$.
- Question: How to find (x, y, σ) satisfy the KKT conditions?

A simple interior-point method

- Basic idea: Assume we have already known $(\bar{x}, \bar{y}, \bar{\sigma})$ that are both primal and dual feasible, i.e., $A\bar{x} = b, \bar{x} > 0$
  $A^T\bar{y} + \bar{\sigma} = c, \bar{\sigma} > 0$
  and try to improve to another point (x^*, y^*, σ^*) to make complementary slackness hold.
- Improvement strategy: Starting from $(\bar{x}, \bar{y}, \bar{\sigma})$, we follow a direction $(\Delta x, \Delta y, \Delta \sigma)$ such that $(\bar{x} + \Delta x, \bar{y} + \Delta y, \bar{\sigma} + \Delta \sigma)$ is a better solution, i.e., it comes closer to satisfying the complementary slackness. 更满足互补松弛性.
- To find such a direction, we substitute $(\bar{x} + \Delta x, \bar{y} + \Delta y, \bar{\sigma} + \Delta \sigma)$ into the KKT conditions:
  - Primal feasibility: $A(\bar{x} + \Delta x) = b$, $\bar{x} + \Delta x \geq 0$; $\rightarrow A\Delta x = 0$
  - Dual feasibility: $A^T(\bar{y} + \Delta y) + (\bar{\sigma} + \Delta \sigma) = c$, $\bar{\sigma} + \Delta \sigma \geq 0$; $\rightarrow A\Delta y + \Delta\sigma = 0$
  - Complementary slackness: $(\bar{X} + \Delta X)(\bar{\Sigma} + \Delta \Sigma)e = 0$. $\Rightarrow$ 代进去

$\Sigma = \begin{pmatrix} \sigma_1 & & \\ & \sigma_2 \ddots & \\ & & \sigma_n \end{pmatrix}$

- Since we start from $(\bar{x}, \bar{y}, \bar{\sigma})$ such that $A\bar{x} = b, \bar{x} > 0$ $A^T\bar{y} + \bar{\sigma} = c, \bar{\sigma} > 0$, the above conditions change into

线性方程组 $\Leftarrow$

  - $A\Delta x = 0,\ \bar{x} + \Delta x \geq 0;$
  - $A^T\Delta y + \Delta\sigma = 0,\ \bar{\sigma} + \Delta\sigma \geq 0;$ 很小，可忽略.
  - $\bar{X}\Delta\Sigma + \bar{\Sigma}\Delta X = -\bar{X}\bar{\Sigma}e - \underline{\Delta X \Delta \Sigma e}.$ 非线性项. $\to \frac{1}{t} - \bar{X}\bar{\Sigma}e - \Delta X \Delta\Sigma e.$

- Note that when Δx and $\Delta\sigma$ are small, the final non-linear term $-\Delta X \Delta \Sigma e$ is small relative to $-\bar{X}\bar{\Sigma}e$ and can thus be dropped out, generating a linear system.

$\bar{X}\Delta\Sigma = -\bar{X}\bar{\Sigma}e - \bar{\Sigma}\Delta X$

$\Delta\Sigma = -\bar{X}^{-1}\bar{X}\bar{\Sigma}e - \bar{X}^{-1}\bar{\Sigma}\Delta X$

$= -\bar{X}^{-1}\bar{X}\bar{\sigma} - \bar{X}^{-1}\bar{\Sigma}\Delta X$

$= -\bar{\sigma} - \bar{X}^{-1}\bar{\Sigma}\Delta X$

  - $A\Delta x = 0,\ \bar{x} + \Delta x \geq 0;$
  - $A^T\Delta y + \Delta\sigma = 0,\ \bar{\sigma} + \Delta\sigma \geq 0;$
  - $\bar{X}\Delta\Sigma + \bar{\Sigma}\Delta X = \underline{-\bar{X}\bar{\Sigma}e}$

$\to \begin{cases} A\Delta x = 0 \\ A^T\Delta y - \bar{\sigma} - \bar{X}^{-1}\bar{\Sigma}\Delta X = 0 \end{cases}$

- The solution are: $\underline{\Delta\sigma = \bar{X}^{-1}(-\bar{X}\bar{\Sigma} - \bar{\Sigma}\Delta x)} = \underline{-\bar{\sigma} - \bar{X}^{-1}\bar{\Sigma}\Delta x}$

$$\begin{pmatrix} -\bar{X}^{-1}\bar{\Sigma} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} \bar{\sigma} \\ 0 \end{pmatrix}$$

满足1,2条件,
没进去满足c.

1: Set initial solution $\bar{x}$ such that $A\bar{x} = b$, $\bar{x} > 0$;
2: Set initial solution $\bar{y}, \bar{\sigma}$ such that $A^T\bar{y} + \bar{\sigma} = c$, $\bar{\sigma} > 0$;
3: **while** TRUE **do**
4: Solve

$$\begin{pmatrix} -\bar{X}^{-1}\bar{\Sigma} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} \bar{\sigma} \\ 0 \end{pmatrix}$$

5: Set $\Delta\sigma = \bar{X}^{-1}(-\bar{X}\bar{\Sigma} - \bar{\Sigma}\Delta x) = -\bar{\sigma} - \bar{X}^{-1}\bar{\Sigma}\Delta x$;
6: Calculate θ to make $\bar{x} + \theta\Delta x \geq 0$, and $\bar{\sigma} + \theta\Delta\sigma \geq 0$;
7: Update $\bar{x} = \bar{x} + \theta\Delta x$, $\bar{y} = \bar{y} + \theta\Delta y$, $\bar{\sigma} = \bar{\sigma} + \theta\Delta\sigma$;
8: **if** $x_i\sigma_i \leq \epsilon$ for all $i = 1, \ldots, m$ **then**
9: break;
10: **end if**
11: **end while**
12: **return** (x, y, σ);

An example

$$
\begin{array}{lrrcr}
\min & 2x_1 & +1.5x_2 & & \\
s.t. & 12x_1 & +24x_2 & \geq & 120 \\
 & 16x_1 & +16x_2 & \geq & 120 \\
 & 30x_1 & +12x_2 & \geq & 120 \\
 & x_1 & & \leq & 15 \\
 & & x_2 & \leq & 15 \\
 & x_1 & & \geq & 0 \\
 & & x_2 & \geq & 0
\end{array}
$$

$\Rightarrow$ $12x_1 + 24x_2 + x_3 = 120$

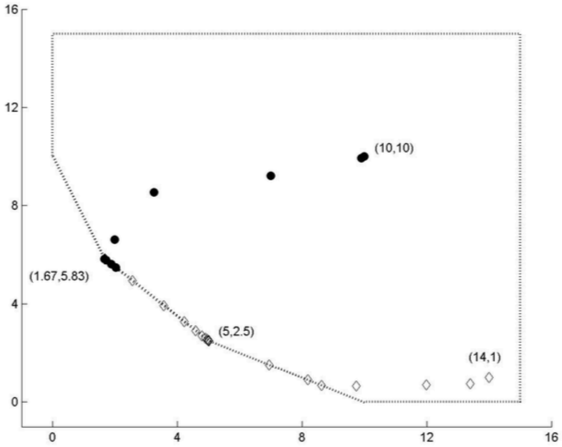
$\downarrow x_3 = 0$

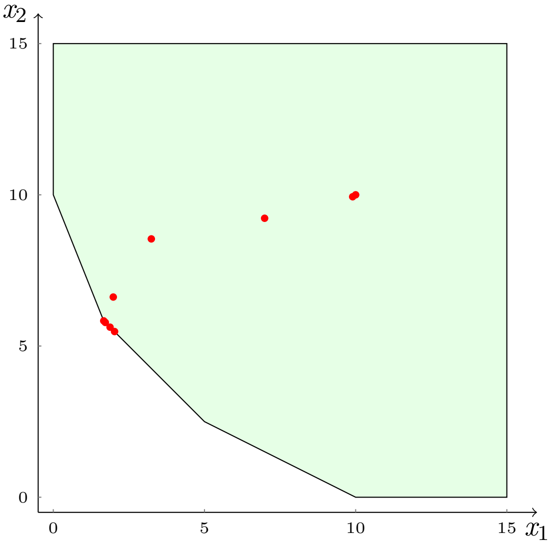
$12x_1 + 24x_x = 120$ → 落在线上

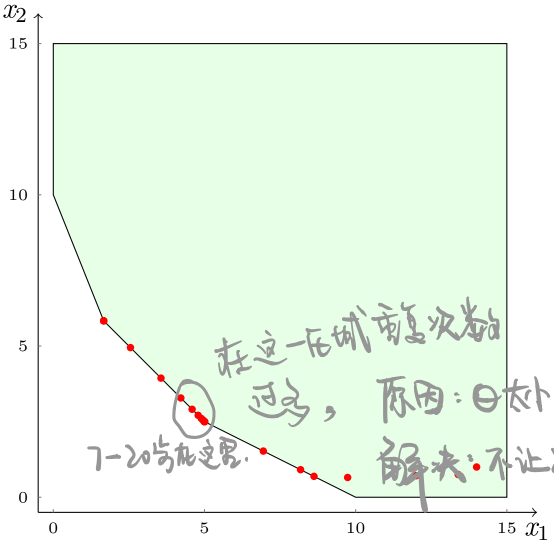
An execution

内点法：始终在内部走.

- Starting point 1: $x = (10, 10)$.
- Starting point 2: $x = (14, 1)$.



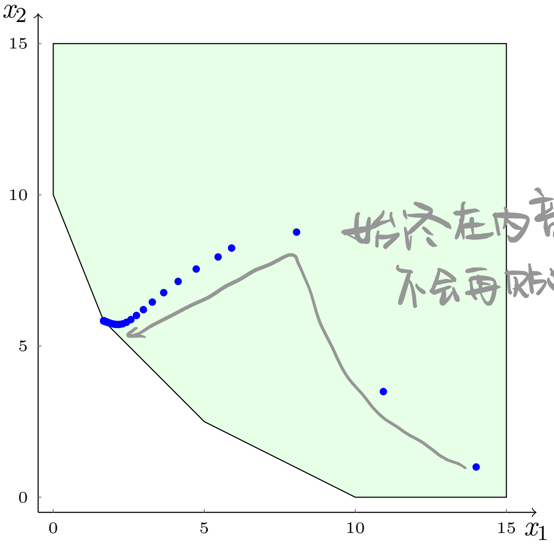

x_2
15
10
5
0
0
5
10
15
x_1



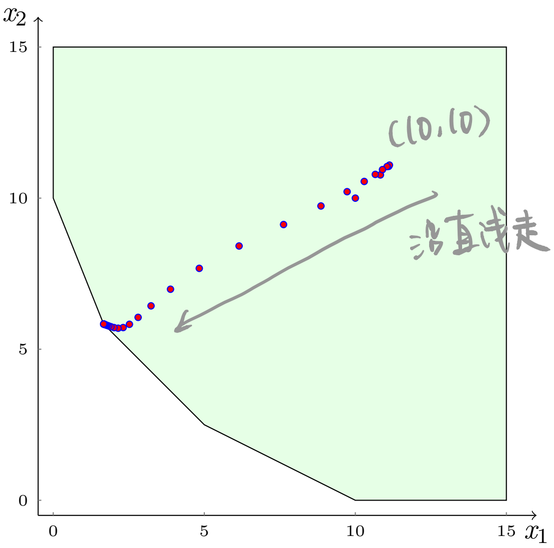
7-20步在这里.

在这一区域迭代次数过多，原因：θ太小 → $x_3=0$ → 太靠近边界

解决：不让 $x_i, \sigma_i = 0$.



始终在内部走.
不会再贴边



x_2
15
10
5
0
0
5
10
15
x_1
(10,10)
沿直线走

Centered interior-point method

- To avoid the poor performance, we need a way to keep the iterate away from the boundary until the solution approaches the optimum.
- An efficient way to achieve this goal is to relax complementary slackness:

$$X\Sigma e = 0$$

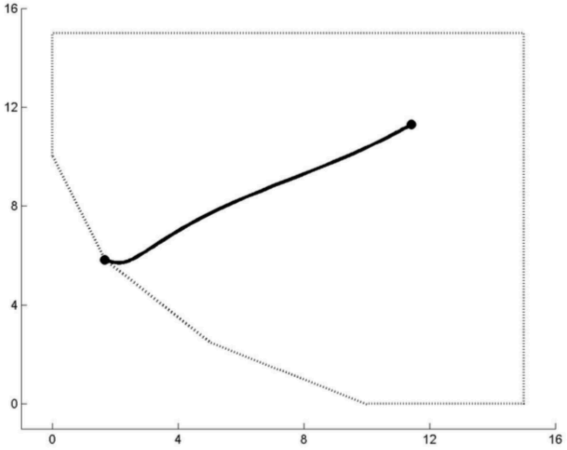
into

$$X\Sigma e = \frac{1}{t} \quad (t > 0)$$ 替代.

- We start from a small t, and gradually increase it as the algorithm proceeds. At each iteration, we execute the previous affine interior point method.

An execution

- Starting point 1: $x = (10, 10)$.
- Starting point 2: $x = (14, 1)$.



Another viewpoint

$L(x,y) = c^T x - \frac{1}{t}\sum_i \log x_i - \lambda^T(Ax-b)$

$\nabla L(x,y) = 0$

$c^T - \frac{1}{t}\sum_i \frac{1}{x_i} - \lambda^T A = 0$

$\Downarrow$

$c - \frac{1}{t}X^{-1}e - A^T\lambda = 0$

- Actually, the relaxed complementary slackness

$$X\Sigma e = \frac{1}{t} \quad (t > 0)$$

corresponds to the following linear program:

barrier function

$$\begin{array}{lrcl} \min & c^T x - \frac{1}{t}\sum_{i=1}^{n} \log x_i & & \\ s.t. & Ax & = & b \\ & x & \geq & 0 \end{array}$$

min $c^T x$
s.t. $Ax = b$
$x \geq 0$

$\Updownarrow$

min $c^T x + I_-(x)$ <不可解
s.t. $Ax = b$

近似

最优解

$\begin{cases} \nabla L(x,y) = 0 \\ Ax = b \end{cases}$

① $c - \frac{1}{t}X^{-1}e - A^T\lambda = 0$

② $A^T y + \sigma = c$

③ $\sigma = \frac{1}{t}X^{-1}e$, $\Sigma = \begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{pmatrix}$

$\Downarrow$

$X\Sigma e = \frac{1}{t}$

Consider a general optimization problem

- Suppose we are trying to solve a convex program with **inequality constraints**:

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

where $f(x)$ and $g(x)$ are convex, and twice differentiable.

Let's start from **equality constrained** quadratic programming

- Suppose we are trying to solve a QP with **equality constraints** :

$$\begin{array}{ll} \min & \frac{1}{2}x^TPx + Q^Tx + r \\ s.t. & Ax = b \end{array}$$

- Applying Lagrangian conditions, we have

$$Ax^* = b, \text{ and } Px + Q + A^T\lambda = 0$$

- Thus, the optimum point x^* can be solved as follows:

$$\begin{bmatrix} P & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x^* \\ \lambda \end{bmatrix} = \begin{bmatrix} -Q \\ b \end{bmatrix}$$

Then how to minimize $f(x)$ with **equality constraints**? Newton's method

- Suppose we are trying to minimize a convex function $f(x)$, which is not a QP.

$$\begin{array}{rc} \min & f(x) \\ s.t. & Ax = b \end{array}$$

- Basic idea: let's try to improve from a feasible solution x. At x, we write the Taylor extension, and use **quadratic approximation** to
$\tilde{f}(x + \Delta x) = f(x) + \bigtriangledown f(x) + \frac{1}{2}\Delta x^T \bigtriangledown^2 f(x)\Delta x$

$$\begin{array}{rc} \min & \tilde{f}(x) \\ s.t. & A(x + \Delta x) = b \end{array}$$

- Thus, the optimum point Δx^* can be solved as follows:

$$\begin{bmatrix} \bigtriangledown^2 f(x) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x^* \\ \lambda \end{bmatrix} = \begin{bmatrix} -\bigtriangledown f(x) \\ b \end{bmatrix}$$

- Since this is just an approximation to $f(x)$, we need to iteratively perform line search.

Then how to minimize $f(x)$ with **inequality constraints**?

- Suppose we are trying to solve a convex program with **inequality constraints**:

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

  where $f(x)$ and $g(x)$ are convex, and twice differentiable.
- Basic idea:
  1. Transform it into a series of **equality constrainted** optimisation problems;
  2. Solve each **equality constrainted** optimisation problem using Newton's method.

Log barrier function: transform into **equality constraints**

Log barrier function

- Suppose we are trying to solve a convex program with **inequality constraints**:

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

  where $f(x)$ and $g(x)$ are convex, and twice differentiable.
- This is equivalent to the following optimization problem.

$$
\begin{aligned}
\min \quad & f(x) + I\_(g(x)) \\
s.t. \quad & Ax = b
\end{aligned}
$$

  where $I\_(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ \infty & \text{otherwise} \end{cases}$
- But indicator function is not differentiable.

Log barrier function: **smooth approximation** to the indicator function

- Suppose we are trying to solve a convex program with **inequality constraints**:

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

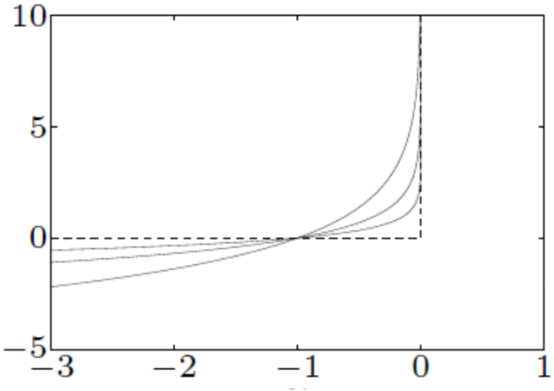
where $f(x)$ and $g(x)$ are convex, and twice differentiable.

- This is equivalent to the following optimization problem.

$$
\begin{aligned}
\min \quad & f(x) - \frac{1}{t}\log(-g(x)) \\
s.t. \quad & Ax = b
\end{aligned}
$$

- Basic idea: a smooth approximation to indicator function $-\frac{1}{t}\log(-x)$, and this approximation improves as $t \to \infty$.

Log barrier function approximates indicator function



- $-\frac{1}{t}\log(-x)$ approximates the indicator function, and the approximation improves as $t \to \infty$.
- For each setting of t, we obtain an approximation to the original optimisation problem.

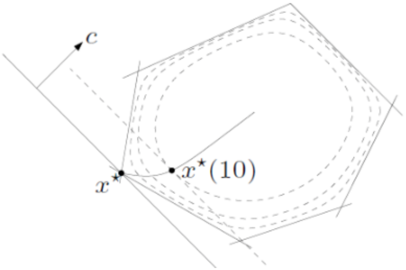
- Solve a sequence of optimization problem:

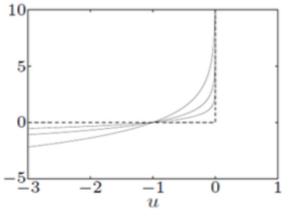
$$\begin{array}{rr} \min & f(x) - \frac{1}{t}\log(-g(x)) \\ s.t. & Ax = b \end{array}$$

- For any $t > 0$, we define $x^*(t)$ as the optimal solution.
- t increases step by step. t should not be too large initially, as it is not easy to solve it using Newton's method.
- Central path: $\{x^*(t) | t > 0\}$.

- Consider the following LP

$$
\begin{array}{rrcl}
\min & c^T x & & \\
s.t. & Ax & \leq & b \\
 & x & \geq & 0
\end{array}
$$





- For any $t > 0$, we define $x^*(t)$ as the optimal solution to:

$$\begin{array}{ll} \min & f(x) - \frac{1}{t}\log(-g(x)) \\ s.t. & Ax = b \end{array}$$

- $x^*(t)$ is not optimal solution to the original problem.
- The duality gap is bounded by $\frac{m}{t}$.

Interior point algorithm

given strictly feasible x, $t := t^{(0)} > 0$, $\mu > 1$, tolerance $\epsilon > 0$.

repeat

1. *Centering step.* Compute $x^\star(t)$ by minimizing $tf_0 + \phi$, subject to $Ax = b$.
2. *Update.* $x := x^\star(t)$.
3. *Stopping criterion.* **quit** if $m/t < \epsilon$.
4. *Increase* t. $t := \mu t$.