# CSC2411 - Linear Programming and Combinatorial Optimization*
# Lecture 8: Interior Point Method and Semi Definite Programming

Notes taken by Sara Mostafavi

March 22, 2007

**Summary:** In this lecture, we show how the Interior Point Algorithm approaches the optimum solution to the LP from the interior of the polytope. Then, we introduce semidefinite programming (SDP), we discuss certain properties of positive semidefinite matrices, and we show how an LP can be transformed into an SDP. We conclude by providing some example SDP problems.

## 1 Interior Point Method

In the last lecture, we were introduced to the Interior Point Method. The simplex method solves linear programming problems (LP) by visiting extreme points (vertices) on the boundary of the feasible set. In contrast, the interior point method is based on algorithms that find an optimal solution while moving in the interior of the feasible set. Intuitively, in each iteration of the interior point method we improve a combination of the objective function while trying to stay away from the boundary. Since we know that the solution to an LP lies on the boundary of the polytope, once the value of the objective function is close to its optimum we find the closest vertex and declare it as the optimum solution. Below we briefly review some definitions and then we will discuss how the Ye's Interior Point Method moves in the interior of the polytope to find the optimal vertex.

**Idea:** Interior points is an iterative method similar to simplex algorithm, where it always holds a feasible solution and it attempts to improve in two different phases. But, unlike the simplex it moves in the interior of the feasible set.

**Defination:** $P$ is a feasible region,
$x$ is 'almost optimal' if $\langle x, c \rangle \leq OPT(P) - 2^{-2L}$

* Lecture Notes for a course given by Avner Magen, Dept. of Computer Sciecne, University of Toronto.

**Lemma:** If $x$ is almost optimal and if $y$ is a $BFS$ so that $\langle y, c\rangle < \langle x, c\rangle$ then $y$ is optimal.

**Proof:** If $y'$ is a $BFS$ $\langle y', c\rangle < \langle y, c\rangle$ then $\langle y, c\rangle - \langle y', c\rangle > 2^{-2L}$

## 1.1 Ye's primal-dual Interior Point Method

The Interior Point Method (IPM) described here is a primal-dual method, that is, it solves both the primal and dual at the same time. This method makes use of the duality gap as described below. Consider the following LP in the primal form:

$$\begin{gathered}\min \langle c, x\rangle \\ \text{s.t } Ax=b \\ x \geq 0\end{gathered}$$

The dual is given by:

$$\begin{gathered}\max \langle b, y\rangle \\ \text{s.t } \ yA \leq c \\ y \gtreqless 0\end{gathered}$$

We can add a slack variable S to the dual and write it in standard form:

$$\begin{gathered}\max \langle b, y\rangle \\ \text{s.t } \ yA + S = c \\ y \gtreqless 0,\ s \geq 0\end{gathered}$$

The Interior Point Method holds (x,s) as a solution. Thus, IPM traverses through the solutions $(x^0, s^0) \longrightarrow (x^{(1)}, s^{(1)}) \longrightarrow \ldots$

**Definition 1.1.** For a pair of solutions to primal P and dual D, the quantity $\langle s, x\rangle$, called the duality gap, gives us a bound on how far we are from the optimum solution for the primal-dual pair.

To see this, let $x$ and $(y, s)$ be solutions to primal and dual program, then

$$\langle c, x\rangle - \langle s, x\rangle = \langle c - s, x\rangle = yAx = \langle y, b\rangle$$

so $\langle c, x\rangle - \langle y, b\rangle \leq \langle s, x\rangle$. Thus, $\langle s, x\rangle$ is an upper-bound on the distance of $\langle c, x\rangle$ from the optimum of the primal which by duality theorem is at least $\langle y, b\rangle$.

The *primal-dual potential function*, describe below, (i) measures the distance of a feasible pair (x,s) to the boundary and (ii) gives an indication about the duality gap.

The primal-dual potential function $G(x, s)$ associated with a primal-dual LP is defined as:

$$G(x, s) = (n + \sqrt{n}) \ln(\langle x, s\rangle) - \sum_i \ln(x_i, s_i)$$

The first term measures the duality gap $\langle x, s\rangle$, the more negative it is, the smaller the duality gap. The second term, $-\sum_i \ln(x_i, s_i)$, measures the distance to the boundary of the feasible set, the more positive it is the closer we are to the boundary of the feasible set. The potential function exhibits the trade off between closeness to the boundary and the size of the duality gap. The parameter $(n+\sqrt{n})$ can be thought of as the tuning parameter. In fact, we claim that if $G(x, s)$ is small enough then $x$ is almost optimal.

**Claim 1.2.** *if $G(x, s) \leq 2\sqrt{n}L$ then $x$ is "almost optimal"*

*proof* It is enough to show that if $G(x, s) \leq 2\sqrt{n}L \Rightarrow \langle x, s\rangle < 2^{-2L}$ since $\langle x, s\rangle$ is an upper bound on the distance of $\langle x, c\rangle$ to the optimum when $x$ is the optimal solution.

$$G(x, s) = \sqrt{n}\ln(\langle x, s\rangle) + [n\ln(\langle x, s\rangle) - \sum \ln(x_i s_i)]$$

The term in the brackets are $\geq 0$ by Jensen's inequality because of the concavity of the logarithm function.

$$\ln(\langle x, s\rangle) = \ln(\sum x_i s_i) = \ln(n\tfrac{1}{n}\sum x_i s_i) \geq \ln(n) + \tfrac{1}{n}\sum \ln(x_i s_i)$$

$$n\ln(\langle x, s\rangle) \geq n\ln(n) + \sum \ln(x_i s_i)$$

$$n\ln(\langle x, s\rangle) - \sum \ln(x_i s_i) \geq n\ln(n) \geq 0$$

therefore,

$$2\sqrt{n}L \geq G(x, s) \geq \sqrt{n}\ln(\langle x, s\rangle) \text{ and so } e^{-2L} \geq 2^{-2L} \geq \langle x, s\rangle\,.$$

## 1.2 An Overview of the Algorithm

The overall goal of the algorithm is to obtain, at each iteration $k$, a set of feasible solutions $(x^{(k)}, s^{(k)})$ that reduce the potential function $G(x, s)$ by a constant positive amount. We have shown how one can use the duality gap and the potential function to measure the "optimality" of a solution, the remaining goals are to show that:

- we can start with an initial solution $\overline{w} = (x, s)$ for which G is not very large
- we can reduce the value of G in $n$ iterations.

.

One way to decrease the potential function, $G$, is to fix $s$ and move $x$ in a manner that follows the opposite direction of $g$, the Gradient of the potential function $G$ with respect to $x$ as evaluated at $(x^{(i)}, x^{(i)})$. First, lets assume that $x^{(i)}$ is the one vector **1**, i.e. $x^{(i)} = 1$.

$$\begin{aligned} g = \nabla_x G(x, s)\Big|_{(x^{(i)}, s^{(i)})} &= \frac{n+\sqrt{n}}{\langle x, s\rangle}s - \begin{pmatrix} 1/x_1 \\ \vdots \\ 1/x_n \end{pmatrix}\Bigg|_{(x^{(i)}, s^{(i)})} \\ &= \frac{n+\sqrt{x}}{\sum s_i}s - 1 \end{aligned}$$

We cannot simply move $x^{(i)}$ in the direction of $-g$ since $A(x^{(i)} - \epsilon g) = b - \epsilon Ag$ is generally not equal to $b$, i.e. $x^{(i)} - \epsilon g$ is not feasible. Consequently, the step must be restricted to the nullspace of $A$ to satisfy the constrains.

$$\begin{aligned} Ax^{(i)} &= b \\ A(x^{(i)} - \epsilon d) &= b \\ Ax^{(i)} - \epsilon Ad &= b \Rightarrow Ad = 0 \end{aligned}$$

So we move in the direction opposite of $d$ where $d$ is the projection of $g$ onto the nullspace of $A$.

$$d = (I - A^T(AA^T)^{-1}A)g$$

Now, we need to determine the size of the step in direction $-d$. Let the size of the step be $\frac{1}{4}\, x^{(i+1)} \geq 0$

$$\begin{aligned} x^{(i+1)} &= x^{(i)} - \frac{1}{4}\frac{d}{\parallel d \parallel_2} \\ x_j^{(i+1)} &= 1 - \frac{1}{4}\frac{d}{\parallel d_j \parallel_2} \geq \frac{3}{4} \\ s^{(i+1)} &= s^{(i)} \end{aligned}$$

It can be shown that if $\parallel d \parallel_2 \geq 0.4$, then $G(x^{(i)}, s^{(i)}) - G(x^{(i+1)}, s(i+1)) \geq \frac{7}{120}$ (see [1] for a proof). Up to now we assumed that $x^{(i)} = 1$, we now show how we can transform the problem by a linear transformation ,$\widehat{X}$, so that $x' = \widehat{X}^{-1}x = 1$,

$$\widehat{X} = \begin{pmatrix} \widehat{x}_1 & & \\ & \ddots & \\ & & \widehat{x}_n \end{pmatrix}$$

Let $\widehat{A} = A\widehat{X}$ and $\widehat{c} = c\widehat{X}$. Then, the transformed problem is given by:

$$\begin{gathered} \min\langle \widehat{c}, x' \rangle \\ \widehat{A}x' = b \\ x' \geq 0 \end{gathered}$$

In the dual, the transformation is $S \rightarrow \widehat{X}s$, so $x'_j s'_j = x_j s_j$ and $G$ is related to $x$ and $s$ through $x_j s_j$. Now we provide a pseudocode for IPM:

**Algorithm 1.3.**
Primal-dual interior point method

**Input:** $m, n \in \mathbb{N}, \epsilon > 0$ and $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$ and initial $x^{(0)}, y^{(0)}, s^{(0)}$ such that $Ax^{(0)} = b, x^{(0)} > 0, y^{(0)T}A + s^{(0)T} = c^T, s^{(0)} > 0.$
**Output:** A feasible pair $(x^{(k)}, s^{(k)})$ such that $s^{(k)T}x^{(k)} < \epsilon$

**while**$(s^{(k)T}x^{(k)} < \epsilon)$

1. Transform the feasible solution with respect to $x^{(k)}$
   $x'^{(k)} = \widehat{X}^{-1}x^{(k)} = \mathbf{1}, s'^{(k)} = \widehat{X}s^{(k)}, A' = A\widehat{X}$.

2. $g \leftarrow \frac{n+\sqrt{x^{(k)}}}{\sum s_i^{(k)}} s^{(k)} - \mathbf{1}$

3. $d \leftarrow (I - A'^T(A'A'^T)^{-1}A')g$

4. if $\| d \|_2 \geq 0.4$, then do a primal step $x'^{(k+1)} \leftarrow \mathbf{1} - \frac{1}{4\|d\|_2}d$
   else do a dual step: $s'^{(k+1)} \leftarrow \frac{s'^T\mathbf{1}}{n+\sqrt{n}}(d+1)$

5. Transform the solution back to the original domain
   $x^{(k+1)} \leftarrow \widehat{X}x'^{(k+1)}, s^{(k+1)} \leftarrow \widehat{X}^{-1}s'^{(k+1)}$

6. $k \leftarrow k + 1$

To summarize, Ye showed that:

1. An initial solution $(x^{(0)}, s^{(0)})$ can always be constructed such that $G(x^{(0)}, s^{(0)}) = O(\sqrt{n})$

2. The algorithm requires $O(\sqrt{n}L)$ iterations for the duality gap $\langle x, s\rangle$ to be minimal, that is, to find an optimal $x$.

The IPM present a polynomial linear programming algorithm which is competitive with the simplex algorithm and it tends to perform better than simplex on large, massively degenerate problems. Furthermore, in each iteration of the IPM the most computational expensive step is the computation of the projection, $d$, of the gradient which requires $O(n^3)$ operations.

# 2 Semi-Definite Programming

## 2.1 Introduction

Semidefinite programming (SDP), an extension of LP, is a relatively new field in optimization which has gained extensive popularity for several reasons. First, many practical problems in operations research and combinatorial optimization can be modeled or approximated as semidefinite programming problems. Furthermore, SDPs are a special case of cone programming and can be efficiently solved by interior point methods.

## 2.2 Review of LP

Lets first review some properties of LP which will give us an intuition about SDPs.Consider the following LP in standard form:

$$\begin{aligned} \text{LP: min } & \langle c, x\rangle \\ \text{s.t. } & a_i x = b_i, i = 1, \ldots, m \\ & x \geq 0 \end{aligned}$$

Here $x$ is a vector in $(\mathbb{R}^+)^n$ and $a_i$ is the row of matrix $A$, $(\mathbb{R}^+)^n = \{x \in \mathbb{R}^n | x \geq 0\}$, we call $(\mathbb{R}^+)^n$ the non-negative orthand. $K = (\mathbb{R}^+)^n$ is called a *closed convex cone*.

**Definition 2.1.** $K$ is called a *closed cone* if it satisfies the following properties:

- if $x, y \in K$, then $\alpha x + \beta y \in K$ for all scalar s $\alpha, \beta \geq 0$.
- $K$ is a closed set.

Now we can re-write the definition of LP as:

$$\begin{aligned} &\text{LP: min} < c, x > \\ &\text{s.t. } a_i x = b_i, i = 1, \ldots, m \\ &\quad x \in (\mathbb{R}^+)^n \end{aligned}$$

## 2.3 Properties of positive semidefinite matrices

**Definition 2.2.** An $n \times n$ matrix $X$ is *positive semi-definite (PSD)* if

$$a^T X a \geq 0, \forall a \in \mathbb{R}^n$$

**Lemma 2.3.** *The following are equivalent:*

1. *$n \times n$ matrix $X$ is PSD*
2. *all eigenvalues of $X$ are non-negative*
3. *there is an $n \times n$ matrix $M$ so that $MM^T = X$*

*proof*
$3 \Rightarrow 1$: If (3), then $\exists M \in \mathbb{R}^{n \times n}$, and $\forall a \in \mathbb{R}^n$ we have:

$$a^T X a = a^T M M^T a = ||Ma||_2^2 \geq 0$$

$1 \Rightarrow 2$: Let $\lambda$ be an eigenvalue of $X \in PSD_n$ and let $a$ be its corresponding eigenvector, then we have $Xa = \lambda a$, so:

$$a^T \lambda a = \lambda ||a||_2^2 = a^T X a \geq 0$$

$2 \Rightarrow 3$: First we recall that if $X$ is symmetric then we can write $X = UDU^T$ for some orthonormal matrix $U$ and some diagonal matrix $D$, where $U$ is orthonormal means that $U^{-1} = U^T$. Now, if $X = UDU^T$ with orthonormal $U$ and diagonal $D$ then columns of $U$ form a set of $n$ orthogonal eigenvectors of $X$, whose eigenvalues are the corresponding diagonal entries of $D$. By (2) all of the eigenvalues of $X$ are positive, then, we can write:

$$\begin{aligned} D &= \sqrt{D}\sqrt{D} \\ X &= (U^T\sqrt{D})(\sqrt{D}U) \\ X &= (\sqrt{D}U)^T(\sqrt{D}U) \\ X &= MM^T,\, M = (\sqrt{D}U) \end{aligned}$$

**Remark 2.4.** How do we decompose $X$? We note that we can diagonalize a symmetric matrix using symmetric row and column operations:

$$\begin{pmatrix} 1 & 3 & 0 \\ 3 & 11 & 2 \\ 0 & 2 & 5 \end{pmatrix} \xrightarrow{R_2=R_2-3R_1} \begin{pmatrix} 1 & 3 & 0 \\ 0 & 8 & 2 \\ 0 & 2 & 5 \end{pmatrix} \xrightarrow{C_2=C_2-3C_1} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 8 & 2 \\ 0 & 2 & 5 \end{pmatrix}$$

We notice that rows and column operations are obtained by modifying **I** and multiplying from left or right:

$$\begin{pmatrix} 1 & 0 & 0 \\ -3 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 3 & 0 \\ 3 & 11 & 2 \\ 0 & 2 & 5 \end{pmatrix} \begin{pmatrix} 1 & -3 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Thus we note the decomposition of $X$ can be done efficiently.

Note that if $X$ is a PSD, then $\forall\, a \in \mathbb{R}^n, a^T X u \geq 0$: $\Rightarrow a^T U^T D U a \geq 0, \forall a \in \mathbb{R}^n \Rightarrow w^T D w \geq 0 \forall w \in \mathbb{R}^n$

Since $X = MM^T = \begin{pmatrix} --v_1-- \\ --v_2-- \\ --v_3-- \end{pmatrix} \begin{pmatrix} | & | & | \\ v_1 & v_2 & v_3 \\ | & | & | \end{pmatrix}$

then $X_{ij} = \langle M_{i:}, M_{:j}^T \rangle = \langle M_i, M_j \rangle = \langle v_i, v_j \rangle$. This indicates that we can think of each element $x_{ij}$ of $X \in PSD_n$ as the value of dot product between two vector $\langle v_i, v_j \rangle$. $X$ can be represented by $n$ vectors so that $v_{ij}$ is $\langle v_i, v_j \rangle$, this is called a Gram product of $v_1, \ldots, v_n$.

## 2.4 Semidefinite Programs

If $X \in PSD_n$ we can think of $X$ as (1) a matrix of $n \times n$ or (2) as a vector with $n^2$ components (i.e. $vec(X) = (x_{11}, x_{12}, \ldots, x_{nn})$). A semidefinite program can be defined as an optimization problem as follows:

$$\begin{gathered} \text{SDP:} \\ \min \quad C \bullet X \\ \text{s.t } A_i X = b_i, i = 1, \ldots, m \\ X \in PSD_n \end{gathered}$$

Where $X, C, A_i$ are $n \times n$ matrices and $C \bullet X$ denotes the matrix inner product such that $C \bullet X = \sum_{i,j} c_{ij} x_{ij}$. This definition of matrix inner product is equivalent to the vector inner product of the vectorized version of $X$ and $C$ which have $n^2$ elements.

**Example 2.5.**

$$\begin{gathered} \min \sum x_{ij} \\ x_{11} + 2x_{13} + 16x_{23} = 0 \\ 4x_{12} + 6x_{22} + 4x_{33} = -2 \\ (x) \in PSD_3 \end{gathered}$$

In this example $C = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$ and $A_1 = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 0 & 16 \\ 0 & 0 & 4 \end{pmatrix}$

**Example 2.6.** Given the following matrix:

$$X = \begin{pmatrix} 5 & \star & \star \\ -3 & 4 & \star \\ \star & \star & \star \end{pmatrix}$$

complete $X$ (i.e fill out the $\star$ entries) such that $A \in PSD_3$ and sum of off diagonal entries is minimize.

$$\begin{gathered} \min \textstyle\sum_{i \neq j} x_{ij} \\ \text{s.t } x_{11} = 5, x_{21} = -3, x_{22} = 4 \\ X \in PSD_3 \end{gathered}$$

Equivalently, we can write the above in matrix form:

$$\min \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \bullet X$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \bullet X = 5, \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \bullet X = -3, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \bullet X = 4$$

$$X \in PSD_3$$

We note that SDP looks very similar to an LP with a few differences: (1) the standard LP constraint that $x \in \mathbb{R}^{+n}$ is replaced with the constraint that $X$ (which can also be thought of as a vector with $n^2$ components) must lie in cone of positive semidefinite matrices.In fact, LP is special case of SDP. Here is how we transform an LP in standard form into an SDP problem:

$$\begin{aligned} LP &\implies SDP \\ \min\langle c, x\rangle &\implies \min \begin{pmatrix} c_1 & & 0 \\ & \ddots & \\ 0 & & c_n \end{pmatrix} \bullet X \\ \langle a_i, x\rangle = b_i, i = 1, \ldots, m &\implies \begin{pmatrix} a_i^{(1)} & & 0 \\ & \ddots & \\ 0 & & a_i^{(n)} \end{pmatrix} \bullet X = b_i, i = 1, \ldots, m \\ x \geq 0 &\implies X \in PSD_n, X_{ij} = 0, \texttt{for} \;\; i \neq j \end{aligned}$$

**Remark 2.7.** We have shown that we can transform an LP to an SDP. In fact, LP is a special case of SDP and an SDP can be thought of as an LP which has infinitely many inequality constrains:

$$\forall a \;\; a^T X a \geq 0 \iff \forall a \;\; (aa^T) \bullet X \geq 0$$

where $aa^T$ is the outer product of vector $a$.

$X$ is PSD is equivalent to the following stating that all principle minors of $X$, $\Lambda_1, \ldots, \Lambda_n$ have non-negative determinants (i.e. det $\Lambda_i > 0, i = 1, \ldots, n$). The $i^{th}$ principle minor of $X$, denoted as $\Lambda_i$ is formed by the first $i$ rows and columns of $X$. Thus, the first principle minor of $X$ is just $x_{11}$.

Now we present an example that gives us an intuition about the geometry of a set of feasible solutions to an SDP.
**eg.**

**Example 2.8.** $X \in PSD, X = \begin{pmatrix} 1 & x & y \\ x & 1 & 0 \\ y & 0 & 1 \end{pmatrix}$
Denoting that all the principle minors are non-negative, we write the constrains as follows:

$$\begin{aligned} 1 &\geq 0 \\ 1 - x^2 &\geq 0 \\ 1 - x^2 - y^2 &\geq 0 \end{aligned}$$

We notice that $x^2 \leq 1$ is redundant in the context of $1 - x^2 - y^2 \geq 0$, so we can summarize the above constraints by the following constraints: $x^2 + y^2 \leq 1$ which is a circle in $\mathbb{R}^2$. Thus, the geometry of the feasible set is not a polyhedron.

**Example 2.9.** Given two set of point in $\mathbb{R}^n$, $P = \{p_1, \ldots, p_r\}$ and $Q = \{q_1, \ldots, q_s\}$ find an ellipsoid which has centre at origin and includes all points in $p_i \in P$ and excludes all point $q_i \in Q$ (we allow $q_i$ to lie on the boundary of the ellipsoid). Figure 1 shows points in $p_i's$ as circles and $q_i's$ as x.

First recall that an ellipse in $\mathbb{R}^n$, centered at the origin, is defined by $\{x | x^t A x \geq 1, A \in PSD_n\}$

We want to find the solution to SDP with the following constrains:

$$\begin{aligned} \forall p_i, p_i^T X p_i &\geq 1 \\ \forall q_i, q_i^T X q_i &\leq 1 \\ X &\in PSD \end{aligned}$$

# References


[1] G. Meinsma. Interior point methods. Mini course, Spring 1997.

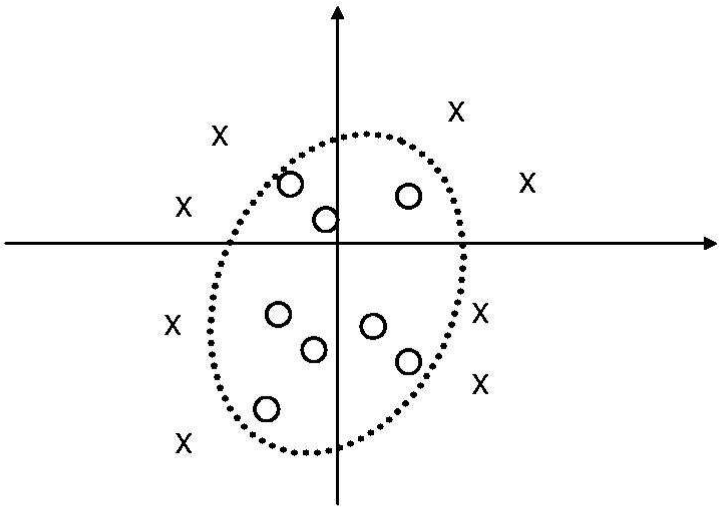


Figure 1: An example SDP problem