Semidefinite Programming: Algorithms, Part I

Angelika Wiegele

April 2014

overview

algorithms for solving semidefinite programs

- interior point methods

overview

algorithms for solving semidefinite programs

- interior point methods
- spectral bundle method

overview

algorithms for solving semidefinite programs

- interior point methods
- spectral bundle method
- bundle method

overview

algorithms for solving semidefinite programs

- interior point methods
- spectral bundle method
- bundle method
- projection methods

semidefinite programs: primal and dual

$$
(\mathbf{SDP}) \begin{cases} \min & \langle C, X \rangle \\ \text{s.t.} & \mathcal{A}(X) = b \\ & X \succeq 0 \end{cases}
$$

semidefinite programs: primal and dual

$$(\textbf{SDP}) \begin{cases} \min & \langle C, X \rangle \\ \text{s.t.} & \mathcal{A}(X) = b \\ & X \succeq 0 \end{cases}$$

$$\min_{X \succeq 0} \max_{y \in \mathbb{R}^m} \langle C, X \rangle + \langle b - \mathcal{A}(X), y \rangle \geq \max_{y \in \mathbb{R}^m} \min_{X \succeq 0} \langle b, y \rangle + \langle X, C - \mathcal{A}^\top(y) \rangle$$

semidefinite programs: primal and dual

$$(\textbf{SDP})\begin{cases} \min & \langle C, X\rangle \\ \text{s.t.} & \mathcal{A}(X) = b \\ & X \succeq 0 \end{cases}$$

$$\min_{X\succeq 0}\max_{y\in\mathbb{R}^m} \langle C, X\rangle + \langle b - \mathcal{A}(X), y\rangle \geq \max_{y\in\mathbb{R}^m}\min_{X\succeq 0} \langle b, y\rangle + \langle X, C - \mathcal{A}^\top(y)\rangle$$

$$(\textbf{DSDP})\begin{cases} \max & b^\top y \\ \text{s.t.} & \mathcal{A}^\top(y) + Z = C \\ & y \in \mathbb{R}^m, Z \succeq 0 \end{cases}$$

interior point methods

strong duality (primal $=$ dual and optima are attained) holds if we assume that both the primal and the dual problem have strictly feasible points, i.e. (X, y, Z) feasible and $X, Z \succ 0$.

interior point methods

strong duality (primal = dual and optima are attained) holds if we assume that both the primal and the dual problem have strictly feasible points, i.e. (X, y, Z) feasible and $X, Z \succ 0$.

Then it follows from the general Karush-Kuhn-Tucker theory that

$$(X, y, Z) \text{ is optimal} \quad \Longleftrightarrow \begin{cases} \mathcal{A}(X) = b, X \succeq 0 \\ \mathcal{A}^\top(y) - Z = C, Z \succeq 0 \\ ZX = 0 \end{cases}$$

interior point methods

strong duality (primal $=$ dual and optima are attained) holds if we assume that both the primal and the dual problem have strictly feasible points, i.e. (X, y, Z) feasible and $X, Z \succ 0$.

Then it follows from the general Karush-Kuhn-Tucker theory that

$$(X, y, Z) \text{ is optimal} \quad \Longleftrightarrow \begin{cases} \mathcal{A}(X) = b, X \succeq 0 \\ \mathcal{A}^\top(y) - Z = C, Z \succeq 0 \\ ZX = 0 \end{cases}$$

note: ZX not symmetric $\longrightarrow$ too many equations.

interior point methods

assumption for rest of talk: (**SCQ**) holds: both the primal and the dual problem have strictly feasible points, i.e., $\exists (X, y, Z)$ feasible and $X, Z \succ 0$.

interior point methods

assumption for rest of talk: (**SCQ**) holds: both the primal and the dual problem have strictly feasible points, i.e., $\exists(X, y, Z)$ feasible and $X, Z \succ 0$.

Consider, for $\mu > 0$ the system:

$$(\mathbf{CP}) \begin{cases} \mathcal{A}(X) = b, X \succeq 0 \\ \mathcal{A}^\top(y) - Z = C, Z \succeq 0 \\ ZX = \mu I \end{cases}$$

interior point methods

assumption for rest of talk: (**SCQ**) holds: both the primal and the dual problem have strictly feasible points, i.e., $\exists (X, y, Z)$ feasible and $X, Z \succ 0$.

Consider, for $\mu > 0$ the system:

$$(\mathbf{CP}) \begin{cases} \mathcal{A}(X) = b, X \succeq 0 \\ \mathcal{A}^\top(y) - Z = C, Z \succeq 0 \\ ZX = \mu I \end{cases}$$

Fundamental Theorem for interior point methods (see e.g. SDP Handbook, Chapter 10):

$$(\mathbf{CP}) \text{ has a unique solution } \forall \mu > 0 \iff (\mathbf{SCQ}) \text{ holds.}$$

interior point methods

assumption for rest of talk: (**SCQ**) holds: both the primal and the dual problem have strictly feasible points, i.e., $\exists (X, y, Z)$ feasible and $X, Z \succ 0$.

Consider, for $\mu > 0$ the system:

$$(\mathbf{CP}) \begin{cases} \mathcal{A}(X) = b, X \succeq 0 \\ \mathcal{A}^\top(y) - Z = C, Z \succeq 0 \\ ZX = \mu I \end{cases}$$

Fundamental Theorem for interior point methods (see e.g. SDP Handbook, Chapter 10):

$$(\mathbf{CP}) \text{ has a unique solution } \forall \mu > 0 \iff (\mathbf{SCQ}) \text{ holds.}$$

this solution $(X(\mu)), y(\mu), Z(\mu))$ forms a smooth curve, called central path.

interior point methods

path following methods: follow the central path by finding points (close to it) for a decreasing sequence of μ.

interior point methods

path following methods: follow the central path by finding points (close to it) for a decreasing sequence of μ.

primal-dual path-following methods: maintain $X, Z \succeq 0$ and try to reach feasibility and optimality. Use Newtons method applied to perturbed problem $ZX = \mu I$ (or variant), and iterate for $\mu \to 0$.

interior point methods

path following methods: follow the central path by finding points (close to it) for a decreasing sequence of μ.

primal-dual path-following methods: maintain $X, Z \succeq 0$ and try to reach feasibility and optimality. Use Newtons method applied to perturbed problem $ZX = \mu I$ (or variant), and iterate for $\mu \to 0$.

idea: **starting** at an interior point $(X \succ 0, y, Z \succ 0)$, find a **search direction** $(\Delta X, \Delta y, \Delta Z)$ such that

$$(X, y, Z) + (\Delta X, \Delta y, \Delta Z)$$

comes closer to the central path for given μ, then **reduce** μ and iterate.

interior point methods

generic primal-dual interior point algorithm

Input.

starting point $(X_0 \succ 0, y_0, Z_0 \succ 0)$, $\varepsilon > 0$.

Initialization.

$\mu_0 := \langle X_0, Z_0 \rangle / n$, $k := 0$.

interior point methods

generic primal-dual interior point algorithm

Input.

starting point $(X_0 \succ 0, y_0, Z_0 \succ 0)$, $\varepsilon > 0$.

Initialization.

$\mu_0 := \langle X_0, Z_0 \rangle / n$, $k := 0$.

while $\mu_k > \varepsilon$ or $\|\mathcal{A}(X_k - b)\|_\infty > \varepsilon$ or $\|\mathcal{A}^\top(y_k) - C - Z_k\|_\infty > \varepsilon$

interior point methods

generic primal-dual interior point algorithm

Input.

starting point $(X_0 \succ 0, y_0, Z_0 \succ 0)$, $\varepsilon > 0$.

Initialization.

$\mu_0 := \langle X_0, Z_0 \rangle / n$, $k := 0$.

while $\mu_k > \varepsilon$ or $\|\mathcal{A}(X_k - b)\|_\infty > \varepsilon$ or $\|\mathcal{A}^\top(y_k) - C - Z_k\|_\infty > \varepsilon$

determine search direction $(\Delta X_k, \Delta y_k, \Delta Z_k)$ from a linearized model for $\mu(\mu_k)$ such that ΔX_k and ΔZ_k symmetric.

interior point methods

generic primal-dual interior point algorithm

Input.

starting point $(X_0 \succ 0, y_0, Z_0 \succ 0)$, $\varepsilon > 0$.

Initialization.

$\mu_0 := \langle X_0, Z_0 \rangle / n$, $k := 0$.

while $\mu_k > \varepsilon$ or $\|\mathcal{A}(X_k - b)\|_\infty > \varepsilon$ or $\|\mathcal{A}^\top(y_k) - C - Z_k\|_\infty > \varepsilon$

determine search direction $(\Delta X_k, \Delta y_k, \Delta Z_k)$ from a linearized model for $\mu(\mu_k)$ such that ΔX_k and ΔZ_k symmetric.

$(X_{k+1}, y_{k+1}, Z_{k+1}) = (X_k, y_k, Z_k) + \alpha_k(\Delta X_k, \Delta y_k, \Delta Z_k)$

interior point methods

generic primal-dual interior point algorithm

Input.

starting point $(X_0 \succ 0, y_0, Z_0 \succ 0)$, $\varepsilon > 0$.

Initialization.

$\mu_0 := \langle X_0, Z_0 \rangle / n$, $k := 0$.

while $\mu_k > \varepsilon$ or $\|\mathcal{A}(X_k - b)\|_\infty > \varepsilon$ or $\|\mathcal{A}^\top(y_k) - C - Z_k\|_\infty > \varepsilon$

determine search direction $(\Delta X_k, \Delta y_k, \Delta Z_k)$ from a linearized model for $\mu(\mu_k)$ such that ΔX_k and ΔZ_k symmetric.

$(X_{k+1}, y_{k+1}, Z_{k+1}) = (X_k, y_k, Z_k) + \alpha_k(\Delta X_k, \Delta y_k, \Delta Z_k)$ with α_k such that $X_{k+1} \succ 0$, $Z_{k+1} \succ 0$

interior point methods

generic primal-dual interior point algorithm

Input.

starting point $(X_0 \succ 0, y_0, Z_0 \succ 0)$, $\varepsilon > 0$.

Initialization.

$\mu_0 := \langle X_0, Z_0 \rangle / n$, $k := 0$.

while $\mu_k > \varepsilon$ or $\|\mathcal{A}(X_k - b)\|_\infty > \varepsilon$ or $\|\mathcal{A}^\top(y_k) - C - Z_k\|_\infty > \varepsilon$

determine search direction $(\Delta X_k, \Delta y_k, \Delta Z_k)$ from a linearized model for $\mu(\mu_k)$ such that ΔX_k and ΔZ_k symmetric.

$(X_{k+1}, y_{k+1}, Z_{k+1}) = (X_k, y_k, Z_k) + \alpha_k(\Delta X_k, \Delta y_k, \Delta Z_k)$ with α_k such that $X_{k+1} \succ 0$, $Z_{k+1} \succ 0$

$\mu_{k+1} = \langle X_{k+1}, Z_{k+1} \rangle / n$

$k = k + 1$

end

interior point methods

system to be solved to find appropriate $(\Delta X, \Delta y, \Delta Z)$

$$
\begin{aligned}
\mathcal{A}(X + \Delta X) &= b \\
\mathcal{A}^\top(y + \Delta y) - C &= Z + \Delta Z \\
(X + \Delta X)(Z + \Delta Z) &= \mu I
\end{aligned}
$$

interior point methods

system to be solved to find appropriate $(\Delta X, \Delta y, \Delta Z)$

$$\begin{aligned} \mathcal{A}(X + \Delta X) &= b \\ \mathcal{A}^\top(y + \Delta y) - C &= Z + \Delta Z \\ (X + \Delta X)(Z + \Delta Z) &= \mu I \end{aligned}$$

$m + \frac{n(n+1)}{2} + n^2$ equations in $2\frac{n(n+1)}{2} + m$ variables (product of symmetric matrices not symmetric in general) $\longrightarrow$ overdetermined.

interior point methods

system to be solved to find appropriate $(\Delta X, \Delta y, \Delta Z)$

$$\begin{aligned} \mathcal{A}(X + \Delta X) &= b \\ \mathcal{A}^\top(y + \Delta y) - C &= Z + \Delta Z \\ (X + \Delta X)(Z + \Delta Z) &= \mu I \end{aligned}$$

$m + \frac{n(n+1)}{2} + n^2$ equations in $2\frac{n(n+1)}{2} + m$ variables (product of symmetric matrices not symmetric in general) $\longrightarrow$ overdetermined. Many variations to fix this:

Replace $ZX - \mu I = 0$ by

- $Z - \mu X^{-1} = 0$
- $X - \mu Z^{-1} = 0$
- $ZX + XZ - 2\mu I = 0$

interior point methods

system to be solved to find appropriate $(\Delta X, \Delta y, \Delta Z)$

$$\begin{aligned} \mathcal{A}(X + \Delta X) &= b \\ \mathcal{A}^\top(y + \Delta y) - C &= Z + \Delta Z \\ (X + \Delta X)(Z + \Delta Z) &= \mu I \end{aligned}$$

$m + \frac{n(n+1)}{2} + n^2$ equations in $2\frac{n(n+1)}{2} + m$ variables (product of symmetric matrices not symmetric in general) $\longrightarrow$ overdetermined.
Many variations to fix this:

Replace $ZX - \mu I = 0$ by

- $Z - \mu X^{-1} = 0$
- $X - \mu Z^{-1} = 0$
- $ZX + XZ - 2\mu I = 0$

$\longrightarrow$ different variants lead to different linearizations.

interior point methods

At start of iteration: $(X \succ 0, y, Z \succ 0)$

interior point methods

At start of iteration: $(X \succ 0, y, Z \succ 0)$

$$\begin{aligned} \mathcal{A}(X + \Delta X) &= b \\ \mathcal{A}^\top(y + \Delta y) - C &= Z + \Delta Z \\ (X + \Delta X)(Z + \Delta Z) &= \mu I \end{aligned}$$

interior point methods

At start of iteration: $(X \succ 0, y, Z \succ 0)$

$$\begin{aligned} \mathcal{A}(X + \Delta X) &= b \\ \mathcal{A}^\top(y + \Delta y) - C &= Z + \Delta Z \\ (X + \Delta X)(Z + \Delta Z) &= \mu I \end{aligned}$$

Linearized system (**CP**) to be solved for $(\Delta X, \Delta y, \Delta Z)$:

interior point methods

At start of iteration: $(X \succ 0, y, Z \succ 0)$

$$\begin{aligned} \mathcal{A}(X + \Delta X) &= b \\ \mathcal{A}^\top(y + \Delta y) - C &= Z + \Delta Z \\ (X + \Delta X)(Z + \Delta Z) &= \mu I \end{aligned}$$

Linearized system (**CP**) to be solved for $(\Delta X, \Delta y, \Delta Z)$:

$\mathcal{A}(\Delta X) = r_P := b - \mathcal{A}(X)$ primal residue

interior point methods

At start of iteration: $(X \succ 0, y, Z \succ 0)$

$$\begin{aligned} \mathcal{A}(X + \Delta X) &= b \\ \mathcal{A}^\top(y + \Delta y) - C &= Z + \Delta Z \\ (X + \Delta X)(Z + \Delta Z) &= \mu I \end{aligned}$$

Linearized system $(\mathbf{CP})$ to be solved for $(\Delta X, \Delta y, \Delta Z)$:

$\mathcal{A}(\Delta X) = r_P := b - \mathcal{A}(X)$ primal residue
$\mathcal{A}^\top(\Delta y) - \Delta Z = r_D := Z + C - \mathcal{A}^\top(y)$ dual residue

interior point methods

At start of iteration: $(X \succ 0, y, Z \succ 0)$

$$\begin{aligned} \mathcal{A}(X + \Delta X) &= b \\ \mathcal{A}^\top(y + \Delta y) - C &= Z + \Delta Z \\ (X + \Delta X)(Z + \Delta Z) &= \mu I \end{aligned}$$

Linearized system (**CP**) to be solved for $(\Delta X, \Delta y, \Delta Z)$:

$\mathcal{A}(\Delta X) = r_P := b - \mathcal{A}(X)$	primal residue
$\mathcal{A}^\top(\Delta y) - \Delta Z = r_D := Z + C - \mathcal{A}^\top(y)$	dual residue
$Z\Delta X + \Delta Z X = \mu I - ZX$	path residue

The last equation can be reformulated in many ways, which all are derived from the complementarity condition $ZX = 0$.

interior point methods

direct approach: using the second and third equation to eliminate ΔX and ΔZ, and substituting into the first gives

$$\Delta Z = \mathcal{A}^\top(\Delta y) - r_D$$
$$\Delta X = \mu Z^{-1} - X - Z^{-1}\Delta Z X$$

interior point methods

direct approach: using the second and third equation to eliminate ΔX and ΔZ, and substituting into the first gives

$$\Delta Z = \mathcal{A}^\top(\Delta y) - r_D$$
$$\Delta X = \mu Z^{-1} - X - Z^{-1}\Delta Z X$$

and the final system in Δy to be solved:

$$\mathcal{A}(Z^{-1}\mathcal{A}^\top(\Delta y)X) = \mu\mathcal{A}(Z^{-1}) - b + \mathcal{A}(Z^{-1}r_D X)$$

interior point methods

direct approach: using the second and third equation to eliminate ΔX and ΔZ, and substituting into the first gives

$$\Delta Z = \mathcal{A}^\top(\Delta y) - r_D$$
$$\Delta X = \mu Z^{-1} - X - Z^{-1}\Delta Z X$$

and the final system in Δy to be solved:

$$\mathcal{A}(Z^{-1}\mathcal{A}^\top(\Delta y)X) = \mu\mathcal{A}(Z^{-1}) - b + \mathcal{A}(Z^{-1}r_D X)$$

Note that the left hand side is a linear system

$$\mathcal{A}(Z^{-1}\mathcal{A}^\top(\Delta y)X) = M\Delta y,$$

but the $m \times m$ matrix M may be expensive to form.
[m... number of constraints of (**SDP**)]

interior point methods

computational effort:

- explicitely determine Z^{-1} $O(n^3)$
- several matrix multiplications $O(n^3)$
- final system of order m to compute Δy $O(m^3)$
- forming the final system matrix $O(mn^3 + m^2n^2)$
- line search to determine $X^+ := X + \alpha\Delta X, Z^+ := Z + \alpha\Delta Z$ is at least $O(n^3)$

note: effort to form system matrix depends on structure of $\mathcal{A}(.)$.

interior point methods

computational effort:

- explicitely determine Z^{-1} $O(n^3)$
- several matrix multiplications $O(n^3)$
- final system of order m to compute Δy $O(m^3)$
- forming the final system matrix $O(mn^3 + m^2 n^2)$
- line search to determine $X^+ := X + \alpha\Delta X, Z^+ := Z + \alpha\Delta Z$ is at least $O(n^3)$

note: effort to form system matrix depends on structure of $\mathcal{A}(.)$.

Limitations: $n \approx 1000, m \approx 10000$.
See benchmark website [H. Mittelmann] at
`http://plato.asu.edu/bench.html`

interior point methods

example: consider the basic SDP relaxation of max-cut, i.e.,

$$
(\mathbf{MC}) \begin{cases} \max & \langle L, X \rangle \\ \text{s.t.} & \operatorname{diag}(X) = e \\ & X \succeq 0 \end{cases}
$$

$X \ldots n \times n$ matrix, and n simple equations $x_{ii} = 1$.

interior point methods

example: consider the basic SDP relaxation of max-cut, i.e.,

$$(\mathbf{MC}) \begin{cases} \max & \langle L, X \rangle \\ \text{s.t.} & \operatorname{diag}(X) = e \\ & X \succeq 0 \end{cases}$$

$X \ldots n \times n$ matrix, and n simple equations $x_{ii} = 1$.

$$\Delta Z = \operatorname{Diag}(\Delta y), \quad \Delta X = -Z^{-1} \Delta Z X + \mu Z^{-1} - X$$

and symmetrize

interior point methods

example: consider the basic SDP relaxation of max-cut, i.e.,

$$
\textbf{(MC)} \begin{cases} \max & \langle L, X \rangle \\ \text{s.t.} & \operatorname{diag}(X) = e \\ & X \succeq 0 \end{cases}
$$

$X \ldots n \times n$ matrix, and n simple equations $x_{ii} = 1$.

$$
\Delta Z = \operatorname{Diag}(\Delta y), \quad \Delta X = -Z^{-1} \Delta Z X + \mu Z^{-1} - X
$$

and symmetrize

solve for Δy

$$
\operatorname{diag}(Z^{-1} \operatorname{Diag}(\Delta y) X) = \mu \operatorname{diag}(Z^{-1}) - e
$$

interior point methods

example: consider the basic SDP relaxation of max-cut, i.e.,

$$(\mathbf{MC}) \begin{cases} \max & \langle L, X\rangle \\ \text{s.t.} & \operatorname{diag}(X) = e \\ & X \succeq 0 \end{cases}$$

$X \ldots n \times n$ matrix, and n simple equations $x_{ii} = 1$.

$$\Delta Z = \operatorname{Diag}(\Delta y), \quad \Delta X = -Z^{-1}\Delta Z X + \mu Z^{-1} - X$$

and symmetrize

solve for Δy

$$\operatorname{diag}(Z^{-1}\operatorname{Diag}(\Delta y)X) = \mu \operatorname{diag}(Z^{-1}) - e$$

i.e.,

$$(Z^{-1} \circ X)\Delta y = \mu \operatorname{diag}(Z^{-1}) - e$$

mcpsd.m

```
function [X,y,iter,secs] = mcpsd(L,digits);

% input: L ... symmetric matrix
% output: X     ... primal matrix
%         y     ... dual variables
% solves: max tr(LX): diag(X)=e, X psd
%         min e'y:    Diag(y)-L=Z psd
% call:   [X,y,iter,secs] = mcpsd(L,digits);
% f. rendl, 2/99

start=cputime;

% initialize data
[n] = size(L,1);
if nargin == 1; digits = 5.5; end;
```

mcpsd.m

```
e = ones(n,1);
X = diag(e);
y = sum(abs(L))' + 1.;
Z = diag(y) - L;
phi = e' * y;
psi = L(:)' * X(:);
delta = phi-psi;

mu = Z(:)' * X(:)/(4*n);
alphap = 1; alphad = 1; iter = 0;
```

mcpsd.m

```
while delta > max([abs(phi) 1]) * 10^(-digits)
% while duality gap too large

   Zi = inv(Z); iter = iter + 1;
   dzi = diag(Zi);
   Zi = (Zi + Zi')/2;
   % solve for dy:
   dy = (Zi .* X) \ (mu * dzi - e);
   tmp = zeros(n);
   for j=1:n
     tmp(:,j) = Zi(:,j)*dy(j);
   end;
   dX = -tmp * X + mu*Zi -X;
   dX = (dX + dX')/2;        % symmetrize
```

mcpsd.m

```
% find steplengths alphap and alphad
alphap =  1;
[Zi,posdef] = chol(X + alphap * dX);
while posdef ~= 0,
  alphap = alphap * .8;
  [Zi,posdef] = chol(X + alphap * dX);
end;
% stay away from boundary
if alphap < 1, alphap = alphap * .95; end;
X = X + alphap * dX;
```

mcpsd.m

```
alphad = 1;
dZ = sparse(diag(dy));
[Zi,posdef] = chol(Z + alphad * dZ);
while posdef ~= 0;
  alphad = alphad * .8;
  [Zi,posdef] = chol(Z + alphad * dZ);
end;
if alphad < 1, alphad = alphad * .95; end;

% update
y = y + alphad * dy;
Z = Z + alphad * dZ;
```

mcpsd.m

```
    mu = X(:)'* Z(:)/(2*n);
    % reduce mu, if stepsize good:
    if alphap + alphad > 1.6
      mu = mu * .75;
    end;
    if alphap + alphad > 1.9
      mu = mu/(1.+.1 * iter);
    end;

    phi = e' * y;
    psi = L(:)'* X(:);
    delta = phi-psi;

    disp([iter alphap alphad log10(delta) psi phi]);
end;  % end of main loop

secs = cputime - start;
```

interior point methods

run times for various graphs when solved using mcpsd.m

n	seconds
200	2
400	7
600	16
800	35
1000	80
1500	260
2000	500

interior point methods

some implementations of interior point methods:

- SeDuMi [J. Sturm 98]: works under Matlab and Octave
- SDPT3 [K. Toh, M. Todd, R. Tutuncu]: Matlab
- CSDP [B. Borchers]: C-library
- SDPA [K. Fujisawa, M. Fukuda, Y. Futakata, K. Kobayashi, M. Kojima, K. Nakata, M. Nakata, M. Yamashita, 95-14]: C-libary, Matlab-interface

interior point methods

some implementations of interior point methods:

- SeDuMi [J. Sturm 98]: works under Matlab and Octave
- SDPT3 [K. Toh, M. Todd, R. Tutuncu]: Matlab
- CSDP [B. Borchers]: C-library
- SDPA [K. Fujisawa, M. Fukuda, Y. Futakata, K. Kobayashi, M. Kojima, K. Nakata, M. Nakata, M. Yamashita, 95-14]: C-libary, Matlab-interface

example: solve max-cut relaxation from before using SeDuMi

mcpsd.m

```
function [x,y,info] = mcpsd(L);
% solve basic max-cut relaxation using SeDuMi
% input: Laplace matrix L
% call: [x,y,info] = mcpsd(L);

n = size(L,1); % number of nodes

% n constraints: diag(X) = e
At = [];
for i=1:n
   B = sparse(i,i,1,n,n);
   At(:,i) = B(:);
end;

b = ones(n,1);
```

mcpsd.m

```
% objective function:
% max <L,X> = min -vec(L)'*vec(X)
c = -L(:);

% semidefiniteness constraint
K.s = [n];

[x,y,info] = sedumi(At,b,c,K);
y = -y;
```

interior point methods

example: random SDP where each A_i is nonzero only on randomly chosen 4×4 submatrix, main diagonal is 0; solved using SeDuMi.

n	m	seconds
100	1000	11
100	2000	159
200	2000	151
200	5000	2607
300	5000	2395

No attempt with larger m due to **memory** and **time**.

interior point methods

example: random SDP where each A_i is nonzero only on randomly chosen 4×4 submatrix, main diagonal is 0; solved using SeDuMi.

n	m	seconds
100	1000	11
100	2000	159
200	2000	151
200	5000	2607
300	5000	2395

No attempt with larger m due to **memory** and **time**.

more results: check out benchmark website [H. Mittelmann] at
`http://plato.asu.edu/bench.html`

widely used format: sdpa format

widely used format: sdpa format
max-cut relaxation for

$$L = \begin{pmatrix} -7 & 0 & 4 & 0 & 3 \\ 0 & 7 & -7 & 0 & 0 \\ 4 & -7 & 5 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & -2 & 0 & -1 \end{pmatrix}$$

widely used format: sdpa format
max-cut relaxation for

$$L = \begin{pmatrix} -7 & 0 & 4 & 0 & 3 \\ 0 & 7 & -7 & 0 & 0 \\ 4 & -7 & 5 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & -2 & 0 & -1 \end{pmatrix}$$

```
5 = mDIM
1 = nBLOCK
5 = bLOCKsTRUCT
1 1 1 1 1
0 1 1 1 -7
0 1 1 3 4
0 1 1 5 3
0 1 2 2 7
0 1 2 3 -7
0 1 3 3 5
0 1 3 5 -2
0 1 5 5 -1
1 1 1 1 1
2 1 2 2 1
3 1 3 3 1
4 1 4 4 1
5 1 5 5 1
```

interior point methods

interior point methods summarized

- based on Newton's method
- currently best convergence results
- many different kind of solvers (SeDuMi, CSDP, SDPA, SDPT3, etc.) see website of benchmarks by H. Mittelmann
- computational effort depends strongly on:
  - matrix dimension n
  - number of constraints m (in each iteration, one needs to solve a dense linear system of order m).
- limit of interior point methods: $n \approx 1000$, $m \approx 10\,000$