Semidefinite Programming: Algorithms (Part 2)

Franz Rendl
Institut für Mathematik
Universität Klagenfurt
A - 9020 Klagenfurt, Austria
franz.rendl@uni-klu.ac.at

Waterloo, May 2004

Practical Performance of Interior-Point Methods for SDP (1)

$$\max\{\langle C, X\rangle : A(X) = b, X \succeq 0\} = \quad \min\{b^T y : A^T(y) - C = Z \succeq 0\}$$

Primal-Dual Path-following Methods:
At start of iteration: $(X \succ 0, y, Z \succ 0)$
Linearized system to be solved for $(\Delta X, \Delta y, \Delta Z)$:

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

$$A^T(\Delta y) - \Delta Z = r_D := Z + C - A^T(y) \quad \text{dual residue}$$

$$Z\Delta X + \Delta Z X = \mu I - ZX \quad \text{path residue}$$

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

This is not a square linear system in $(\Delta X, \Delta y, \Delta Z)$ because there are

$$2\binom{n+1}{2} + m$$

variables but the number of equations is

$$\binom{n+1}{2} + n^2 + m.$$

Practical Performance of Interior-Point Methods for SDP (2)

Direct approach with partial elimination:
Using the second and third equation to eliminate ΔX and ΔZ, and substituting into the first gives

$$\Delta Z = A^T(\Delta y) - r_D, \;\; \Delta X = \mu Z^{-1} - X - Z^{-1}\Delta Z X,$$

and the final system to be solved:

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

Computational effort:

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

Effort to determine system matrix depends on structure of $A(.)$

Practical Performance of Interior-Point Methods for SDP (3)

Example 1: SDP Relaxation for Max-Cut:

$$\max\{\langle C, X\rangle : \text{diag}(X) = e, X \succeq 0\} = \quad \min\{e^T y : \text{Diag}(y) - C = Z \succeq 0\}$$

$$\text{Here: } m = n,\ A(X) = \text{diag}(X),\ A^T(y) = \text{Diag}(y)$$

and the system matrix becomes

$$\text{diag}(Z^{-1}\text{Diag}(\Delta y)X) = (Z^{-1} \circ X)\Delta y.$$

It can be computed in $O(n^2)$

n	seconds
400	8.92
600	24.10
800	51.45
1000	99.27
1500	314.99
2000	714.21

Computation times (seconds) to solve the SDP on a PC (Pentium 4, 1.7 Ghz).

see Helmberg, Rendl, Vanderbei, Wolkowicz: SIOPT (6) 1996, 342ff

Practical Performance of Interior-Point Methods for SDP (4)

Example 2: Lovasz Theta function:

Given a graph $G = (V, E)$ with $|V| = n, \ |E| = m$.

$$\max\{\langle J, X\rangle : \text{tr}(X) = 1, x_{ij} = 0 \ \forall (ij) \in E, \ X \succeq 0\}$$

Here the number of constraints depends on the edge set $|E|$.

If $m >> n^2/4$ then system size impractical.

Use explicit representation of X, i.e. express X through main diag and non-edge variables

This gives final system of size $n^2/2 - m$ which is smaller than m.

This allows to compute $\vartheta(G)$ for very sparse and very dense graphs. The computationally difficult class are graphs with $m \approx n^2/4$, i.e. about half the possible edges are present.

see: Dukanovic, Rendl, technical report, Klagenfurt 2004

Practical Performance of Interior-Point Methods for SDP (5)

Iterative solution of linear system:

To avoid computing Z^{-1} explicitely, and forming the system matrix, one could use iterative methods to compute $(\Delta X, \Delta y, \Delta Z)$.

Preconditioned Conjugate Gradient method

Spectral Bundle Method for SDP (1)

Dual as eigenvalue optimization problem:

Assume that $A(X) = b$ implies that $\text{tr}(X) = a > 0$. (Holds for many combinatorially derived SDP!)

Reformulate dual

$$\min\{b^T y : A^T(y) - C = Z \succeq 0\}$$

as follows. Adding (redundant) primal constraint $\text{tr}(X) = a$ introduces new dual variable, say λ, and dual becomes

$$\min\{b^T y + a\lambda : A^T(y) - C + \lambda I = Z \succeq 0\}$$

At optimality, Z is singular, hence $\lambda_{\min}(Z) = 0$.

compute dual variable λ explicitely:

$$\lambda_{\max}(-Z) = \lambda_{\max}(C - A^T(y)) - \lambda = 0, \Rightarrow \lambda = \lambda_{\max}(C - A^T(y))$$

Dual equivalent to

$$\min a\lambda_{\max}(C - A^T(y)) + b^T y : y \in \mathbb{R}^m$$

This is non-smooth unconstrained convex problem in y.

Spectral Bundle Method for SDP (2)

Minimizing $f(y) = \lambda_{\max}(C - A^T(y)) + b^T y$:

Note: Evaluating $f(y)$ at y amounts to computing largest eigenvalue of $C - A^T(y)$.

Can be done by iterative methods for very large (sparse) matrices.

If we have some y, how do we move to a better point?

$$\lambda_{\max}(X) = \max\{\langle X, W\rangle : \text{tr}(W) = 1,\ W \succeq 0\}$$

Define

$$L(W, y) := \langle C - A^T(y), W\rangle + b^T y.$$

Then $f(y) = \max\{L(W, y) : \text{tr}(W) = 1,\ W \succeq 0\}$.

Idea 1: Minorant for $f(y)$

Fix some $m \times k$ matrix P. $k \geq 1$ can be chosen arbitrarily. The choice of P will be explained later.

Consider W of the form $W = PVP^T$ with new $k \times k$ matrix variable V.

$$\hat{f}(y) := \max\{L(W, y) : W = PVP^T,\ V \succeq 0\} \leq f(y)$$

Spectral Bundle Method for SDP (3)

Idea 2: Proximal point approach

The function $\hat{f}$ depends on P and will be a good approximation to $f(y)$ only in some neighbourhood of the current iterate $\hat{y}$.
Instead of minimizing $f(y)$ we minimize

$$\hat{f}(y) + \frac{u}{2}\|y - \hat{y}\|^2.$$

This is a strictly convex function, if $u > 0$ is fixed.
Substitution of definition of $\hat{y}$ gives

$$\min_y \max_W L(W, y) + \frac{u}{2}\|y - \hat{y}\|^2 = \ldots$$

$$= \max_{W,\ y = \hat{y} + \frac{1}{u}(A(W) - b)} L(W, y) + \frac{u}{2}\|y - \hat{y}\|^2$$

$$= \max_W \langle C - A^T(\hat{y}), W\rangle + b^T\hat{y} - \frac{1}{2u}\langle A(W) - b, A(W) - b\rangle.$$

Note that this is a quadratic SDP in the $k \times k$ matrix V. Once V is computed, we get with $W = PVP^T$ that $y = \hat{y} + \frac{1}{u}(A(W) - b)$

see: Helmberg, Rendl: SIOPT 10, (2000), 673ff

Spectral Bundle Method for SDP (4)

Update of P:

Having new point y, we evaluate f at y (sparse eigenvalue computation), which produces also an eigenvector v to $\lambda_{\max}$.
The vector v is added as new column to P, and P is purged by removing unnecessary other columns.

Convergence is slow, once close to optimum

Can approximately solve SDP with quite large matrices, $n \approx 5000$.

see also DIMACS challenge for SDP - DIMACS web-page

Bundle methods and SDP (1)

Dealing with SDP with too many inequality constraints

The number m of equality constraints is clearly always less than $\binom{n+1}{2}$, because this is the dimension of the space of S_n.

But there can be an arbitrary number of inequality constraints !

If we do not know whether a contraint is active or not, we introduce a nonnegative slack variable and make the constraint an equality.

This increases also the dimension of the space (we added one more variable).

Many SDP from combinatorial optimization can be tightened by introducing combinatorial cutting planes (=linear inequalities)

Example: Max-Cut SDP Relaxation with Triangle inequalities

Triangle inequalities for Max-Cut:

A simple observation: if x is an arbitrary cut-vector:

$$x \in \{-1, 1\}^n, \quad f = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad \Rightarrow |x^T f| \geq 1$$

Translated to $X = xx^T$:

$$x^T f \; f^T x = \langle (xx^T), (ff^T) \rangle = \langle X, ff^T \rangle \geq 1$$

Can be applied to any **triangle** $i < j < k$. Nonzero elements of f can also be -1. This gives $4\binom{n}{3}$ linear inequalities.

Triangle Relaxation

$$x_{ij} + x_{ik} + x_{jk} \geq -1 \qquad x_{ij} - x_{ik} - x_{jk} \geq -1 \quad \forall i < j < k$$

Deza, Laurent: Hypermetric Inequalities, Padberg: Quadric Boolean Polytope

Dealing with the Triangle Relaxation

n	$\binom{n}{2}$	$4\binom{n}{3}$
50	1.225	78.400
100	4.950	646.800
200	19.900	5.253.600
500	124.750	82.834.000

Triangle constraints as n increases

Only $\binom{n}{2}$ constraints determine off-diagonal part of X. Good candidates for active constraints ?? Explicitely maintaining all these constraints is infeasible for $n \geq 25$.

Computation times with only a limited number of triangle inequlities included.

Computation times for SDP with k triangles included

	$n = 100$	$n = 200$	$n = 300$
$k = 500$	21 (19)	34 (20)	49 (19)
$k = 1000$	103 (21)	136 (22)	164 (21)
$k = 1500$	304 (24)	358 (24)	422 (24)
$k = 2000$	643 (25)	763 (26)	816 (24)
$k = 2500$	1090 (24)	1313 (26)	1360 (24)

Computation times (seconds) on a PC (Pentium 4, 1.7 GHz) to compute the semidefinite relaxation of Max-Cut for a graph with n nodes and k triangle inequalities. The number of interior point iterations is given in parentheses.

Bundle methods and SDP (2)

If there are too many inequality constraints, we can look at their Lagrangian Dual.
Maintain only part of constraints explicitely

$$X \in F := \{X : \text{diag}(X) = e,\ X \succeq 0\}$$

Remaining constraints ($A(X) \leq b$) are dualized through Lagrangian:

$$L(X, \gamma) = \langle C, X \rangle + \gamma^T (b - A(X))$$

$$\max_{X \in F,\ A(X) \leq b} \langle C, X \rangle = \max_{X \in F} \min_{\gamma \geq 0} L(X, \gamma)$$

Dual functional:

$$f(\gamma) := \max_{X \in F} L(X, \gamma)$$

Minimizing f is equivalent to original problem
f is convex, but non-smooth

Bundle methods and SDP (3)

Bundle methods:

minimize f using function and subgradient evaluation only

Note: function evaluation means solving over $X \in F$.

Big graphs (from Helmberg).

The number of bundle iterations is 50 for $n = 800$, and 30 for $n = 2000$.

problem	n	$\|E\|$	cut	initial bd	gap (%)	final bd	gap (%)	m	time
G1	800	19176	11612	12083.2	4.06	12005.4	3.39	7372	51.76
G6	800	19176	2172	2656.2	22.29	2566.2	18.15	6983	43.11
G11	800	1600	564	629.2	11.56	572.7	1.54	15946	60.20
G14	800	4694	3054	3191.6	4.51	3140.7	2.84	8973	59.68
G18	800	4694	985	1166.0	18.38	1063.4	7.96	17635	69.19
G22	2000	19990	13293	14135.9	6.34	14045.8	5.66	18325	278.06
G27	2000	19990	3293	4141.7	25.77	4048.4	22.94	15178	406.66
G39	2000	11779	2373	2877.7	21.27	2672.7	12.63	26471	533.36

These are currently the best bounds for these problems, see Fischer et al., technical report, Klagenfurt, 2004

Approximation results using SDP

Goemans-Williamson hyperplane rounding technique for Max-Cut
Nesterov Analysis for Max-Cut
Karger-Motwani-Sudan technique for Graph Coloring