Dual and Lagrangian dual interior-point methods for semidefinite programs

SIAM Journal on Optimization **Vol 12, No.4, 1007-1031 (2002)**

Mituhiro Fukuda[†] *(mituhiro@is.titech.ac.jp)*
Masakazu Kojima[†] *(kojima@is.titech.ac.jp)*
Masayuki Shida[‡] *(shida@cc.nda.ac.jp)*

† **Tokyo Institute of Technology**
‡ **National Defense Academy of Japan**

This talk

1. **Semidefinite Program (SDP).**
2. **Major difficulties in solving large scale (sparse) SDPs by primal-dual interior-point methods.**
3. **Lagrangian Dual Interior-Point Method (LDIPM) — main part.**
4. **Preliminary numerical results.**

1. Semidefinite Program (SDP)

$$\text{Primal} \begin{cases} \textbf{max.} & \boldsymbol{C} \bullet \boldsymbol{X} \\ \textbf{sub.to} & \boldsymbol{A}_p \bullet \boldsymbol{X} = a_p \ (1 \leq p \leq m), \ \ \boldsymbol{X} \succeq \boldsymbol{O} \end{cases}$$

$$\text{Dual} \begin{cases} \textbf{min.} & \sum_{p=1}^{m} a_p y_p \\ \textbf{sub.to} & \sum_{p=1}^{m} \boldsymbol{A}_p y_p - \boldsymbol{S} = \boldsymbol{C}, \ \ \boldsymbol{S} \succeq \boldsymbol{O} \end{cases}$$

large scale if
n **and/or** m **: large**
e.g.,
$n \geq 1,000 \sim 10,000$,
$m \geq 1,000 \sim 100,000$.

where

$\mathcal{S}^n$ **:** $n \times n$**-symmetric matrices**

$\boldsymbol{C}, \boldsymbol{A}_1, \cdots, \boldsymbol{A}_m \in \mathcal{S}^n, \quad a_1, a_2, \ldots, a_m \in \mathbb{R}$ **are given data**

$\boldsymbol{X} \in \mathcal{S}^n$ **: primal matrix variable**

$\boldsymbol{S} \in \mathcal{S}^n$ **: dual matrix variable**

$\boldsymbol{A} \bullet \boldsymbol{X}$ **: inner product** $\displaystyle\sum_{p=1}^{n}\sum_{q=1}^{n} A_{pq}X_{pq}$

$\boldsymbol{X} \succeq \boldsymbol{O}$ **:** $\boldsymbol{X}$ **is a symm. positive semidefinite matrix**

Our objective :
Solve large-scale (sparse) SDPs with high accuracy

— **a challenging problem although many studies (***Benson-Ye-Zhang SIOPT '00, Helmberg-Rendl SIAM'00, Burer-Monteiro-Zhang '99, Vanderbei-H.Benson, Fukuda-Kojima-Murota-Nakata SIOPT '01, etc.* **) have been done extensively and intensively form various directions.**

More specifically,

- **Overcome major difficulties involved in primal-dual IPMs**

2. Major difficulties in primal-dual IPM — 1

♠ **The primal $\boldsymbol{X}$ becomes dense even when $\boldsymbol{A}_0, \boldsymbol{A}_1, \ldots, \boldsymbol{A}_m$ are sparse.**

- **The dual $\boldsymbol{S} = \sum_{p=1}^{m} \boldsymbol{A}_p y_p - \boldsymbol{C}$ inherits sparsity from $\boldsymbol{A}_0, \boldsymbol{A}_1, \ldots, \boldsymbol{A}_m$.**
- **IPMs which work only in the dual space have a clear advantage.**

In LDIPM:

♢ **Evaluate $\boldsymbol{X}$ only when $\boldsymbol{X}\boldsymbol{S} = \mu \boldsymbol{I}$ for some $\mu > 0$.**
Store the sparse Cholesky factorization $\boldsymbol{S} = \boldsymbol{L}\boldsymbol{L}^T$.
Then $\boldsymbol{X} = \mu \boldsymbol{L}^{-T} \boldsymbol{L}^{-1}$ is easily retrieved.

♢ **No line search in $\boldsymbol{X}$.**

Major difficulties in Primal-dual IPM — 2

♠ **Fully dense $m \times m$ linear system $\boldsymbol{B}d\boldsymbol{y} = \boldsymbol{r}$, called the Schur complement equation, to compute search direction, where $\boldsymbol{B}$ and $\boldsymbol{r}$ are functions of iterates $(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{S})$**

- **We can use the CG method, but need an effective preconditioner because $\boldsymbol{B}$ becomes ill-conditioned as $(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{S}) \rightarrow$ an opt. solution.**

In LDIPM:

♢ **Corrector: BFGS quasi-Newton method.**
♢ **Predictor: CG method using the BFGS quasi-Newton matrix as an effective preconditioner**

Existing methods to resolve and/or avoid these difficulties

(I) Dual interior-point methods — *Benson-Ye-Zhang SIOPT '00*

(II) Spectral bundle method — *Helmberg-Rendl SIAM'00*

(III) Nonlinear programming formulation

— *Burer-Monteiro-Zhang '99, Vanderbei-H.Benson '00*

(IV) Positive semidefinite matrix completion techniques

— *Fukuda-Kojima-Murota-Nakata SIOPT '01*

"Solving general large scale SDPs in high accuracy" is still a challenging problem

3. Lagrangian Dual Interior-Point Method

Semidefinite Program solved by LDIPM

$$\textbf{Primal} \begin{cases} \textbf{max.} & \boldsymbol{C} \bullet \boldsymbol{X} \\ \textbf{sub.to} & \boldsymbol{A}_p \bullet \boldsymbol{X} = a_p \ (1 \leq p \leq m), \ \boldsymbol{I} \bullet \boldsymbol{X} = b, \boldsymbol{X} \succeq \boldsymbol{O} \end{cases}$$

$$\textbf{Dual} \begin{cases} \textbf{min.} & \sum_{p=1}^m a_p y_p + bw \\ \textbf{sub.to} & \sum_{p=1}^m \boldsymbol{A}_p y_p + \boldsymbol{I} w - \boldsymbol{S} = \boldsymbol{C}, \ \boldsymbol{S} \succeq \boldsymbol{O} \end{cases}$$

Here $b > 0$**.**

- **"Simplex constraint"** $\{\boldsymbol{X} \succeq \boldsymbol{O} : \boldsymbol{I} \bullet \boldsymbol{X} = b\}$**, which was assumed in some existing works.**
- **Restrictive, but many applications;**
  SDPs having known bounded feasible regions $\Rightarrow$ **Primal Problem**

Assumption

1. $\exists \boldsymbol{X}^0 \succ \boldsymbol{O}$ **feasible for Primal SDP (Slater c.q.)**

2. Data matrices $\boldsymbol{A}_p$ $(1 \leq p \leq m)$ **and** $\boldsymbol{I}$ **are linearly independent**

$\diamondsuit$ **Basic idea of LDIPM: For $\forall \boldsymbol{y} \in \mathbb{R}^m$ and $\forall \mu > 0$, let**

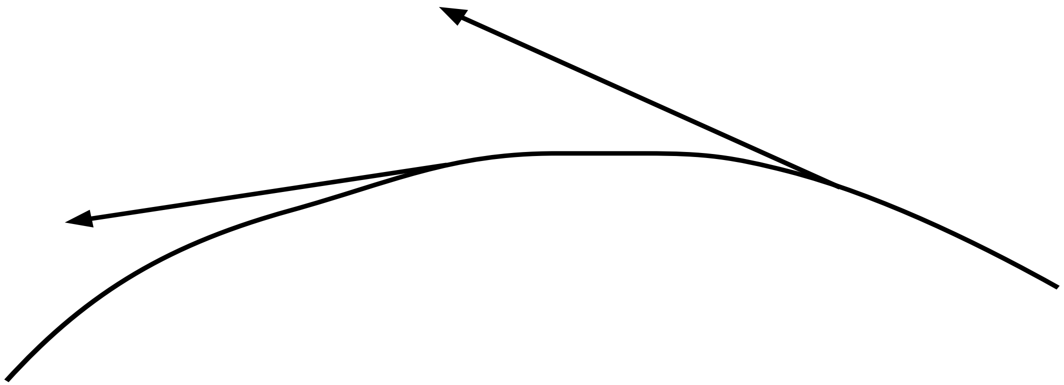
$$g(\boldsymbol{y}, \mu) \equiv \boxed{\textbf{(D)}_{(\boldsymbol{y},\mu)} \left\{ \begin{array}{ll} \textbf{min.} & \sum_{p=1}^m a_p y_p + bw - \mu \log\det \boldsymbol{S} \\ \textbf{sub.to} & \boldsymbol{I}w - \boldsymbol{S} = \boldsymbol{C} - \sum_{p=1}^m \boldsymbol{A}_p y_p, \ \ \boldsymbol{S} \succ \boldsymbol{O} \end{array} \right.}$$

$\exists^1$ **min. sol.** $\boldsymbol{w}(\boldsymbol{y}, \mu)$, $\boldsymbol{S}(\boldsymbol{y}, \mu)$

$\Downarrow$ **Unconstrained convex minimization (Lagrangian dual):**

Given $\mu > 0$, $\boxed{\textbf{min. } g(\boldsymbol{y}, \mu) \ \ \textbf{sub.to } \boldsymbol{y} \in \mathbb{R}^m}$

LDPIM – **Trace the minimizer $\boldsymbol{y}(\mu)$ of $g(\boldsymbol{y}, \mu)$ or the solutions of $\boldsymbol{\nabla}_y g(\boldsymbol{y}, \mu) = \boldsymbol{0}$ $(\mu \rightarrow 0)$ by predictor-corrector.**



Common coefficient matrix $\boldsymbol{\nabla}_{yy} g(\boldsymbol{y}, \mu)$ is used!

Morales-Nocedal '01.

$$\textbf{Lin.sys. behind corrector:} \quad \boldsymbol{\nabla}_{yy} g(\hat{\boldsymbol{y}}^k, \mu^\ell) \boldsymbol{dy}_c = -\boldsymbol{\nabla}_y g(\hat{\boldsymbol{y}}^k, \mu^\ell) \qquad (1)$$

$$\textbf{Lin.sys. behind predictor:} \quad \boldsymbol{\nabla}_{yy} g(\boldsymbol{y}^\ell, \mu^\ell) \boldsymbol{dy}_p = +\boldsymbol{\nabla}_{y\mu} g(\boldsymbol{y}^\ell, \mu^\ell) \qquad (2)$$

$\diamondsuit$ **BFGS q-Newton method to** $\Rightarrow \hat{\boldsymbol{y}}^0, \hat{\boldsymbol{y}}^1 \ldots \to \boldsymbol{y}^\ell \approx \boldsymbol{y}(\mu^\ell)$**.**

$\diamondsuit$ **CG method to (2) with effective precond. from BFGS.**

Computation of $g(\boldsymbol{y},\mu)$, $\boldsymbol{\nabla}_y g(\boldsymbol{y},\mu), \boldsymbol{\nabla}_{yy} g(\boldsymbol{y},\mu)$, $\boldsymbol{\nabla}_{y\mu} g(\boldsymbol{y},\mu)$ **is based on KKT condition of (D)**$_{(\boldsymbol{y},\mu)}$**:**

$(w(\boldsymbol{y},\mu), \boldsymbol{S}(\boldsymbol{y},\mu))$ **is the optimal sol. of** **iff** $\exists\ \boldsymbol{X}(\boldsymbol{y},\mu)$**;**

$$\boxed{\begin{array}{c} \boldsymbol{I} \bullet \boldsymbol{X}(\boldsymbol{y},\mu) = b, \quad \boldsymbol{I} w(\boldsymbol{y},\mu) - \boldsymbol{S}(\boldsymbol{y},\mu) = \boldsymbol{C} - \sum_{p=1}^{m} \boldsymbol{A}_p y_p, \\ \boldsymbol{X}(\boldsymbol{y},\mu)\boldsymbol{S}(\boldsymbol{y},\mu) = \mu \boldsymbol{I}, \quad \boldsymbol{X}(\boldsymbol{y},\mu) \succeq \boldsymbol{O}, \quad \boldsymbol{S}(\boldsymbol{y},\mu) \succeq \boldsymbol{O}. \end{array}} \Rightarrow$$

d.feasible but not p.feasible in general

$\Rightarrow$ $\boldsymbol{X}$ **is evaluated only when** $\boldsymbol{X}\boldsymbol{S} = \mu \boldsymbol{I}$**. In addition,**

$\boxed{\boldsymbol{A}_p \bullet \boldsymbol{X}(\boldsymbol{y}(\mu),\mu) = a_p\ (1 \leq p \leq, m) \text{ at min. } \boldsymbol{y}(\mu) \text{ of } g(\boldsymbol{y},\mu).}$ $\Rightarrow$ **p.feasible**

$\Rightarrow$ $(\boldsymbol{X}(\boldsymbol{y}(\mu),\mu), \boldsymbol{y}(\mu), \boldsymbol{S}(\boldsymbol{y}(\mu),\mu))$ **lies on the central trajectory.**

Some other features — 1.

Second order predictor using

$$\nabla_{yy} g(\boldsymbol{y}(\mu), \mu)\dot{\boldsymbol{y}}(\mu) = \exists \boldsymbol{a}(\boldsymbol{y}, \mu) \text{ — the 1st order derivative},$$
$$\nabla_{yy} g(\boldsymbol{y}(\mu), \mu)\ddot{\boldsymbol{y}}(\mu) = \exists \boldsymbol{b}(\boldsymbol{y}, \mu) \text{ — the 2nd order derivative}.$$

We need to compute $\dot{\boldsymbol{y}}(\mu)$ and $\ddot{\boldsymbol{y}}(\mu)$ by using the CG method.

Some other features — 2.

Dual IP method, a simpler version for the dual SDP

$$\texttt{Dual}: \textbf{min. } \sum_{p=1}^m a_p y_p \textbf{ sub.to } \boldsymbol{S} = \sum_{p=1}^m \boldsymbol{A}_p y_p - \boldsymbol{C} \succeq \boldsymbol{O}$$

based on

$$\tilde{g}(\boldsymbol{y}, \mu) \equiv \boxed{\sum_{p=1}^m a_p y_p + bw - \mu \log\det \boldsymbol{S}} \quad (\forall \textbf{ int.feas. sol. } \boldsymbol{y} \textbf{ and } \mu > 0) \textbf{ and}$$

$$\boxed{\textbf{min. } \tilde{g}(\boldsymbol{y}, \mu) \textbf{ sub.to } \boldsymbol{y} \textbf{ : int.feas. sol.}} \quad (\mu > 0)$$

Preliminary numerical results

- **Macintosh (400MHz) with MATLAB v.5.2.**
- 8 **variants of LDIPMs:**

  Dual or Lagrangian dual IPMs.

  The 1st order or the 2nd order predictor.

  Newton or BFGS quasi-Newton method for corrector steps.
- **Randomly generated test problems.** 5 **problems / each type.**

  (a) SDP relaxation of box constrained quadratic ± 1 **programs:** $(n, m) = (101, 100),\ (201, 200)$.

  (b) Norm minimization problems: $(n, m) = (50, 100),\ (50, 200)$.

  (c) Linear matrix inequality: $(n, m) = (50, 100),\ (50, 200)$.

Box Constrained Quadratic ±1 Program

- **Average of** 5 **problems** $\{\max \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x}$ **sub.to** $x_i^2 = 1,\ (1 \le i \le n)\}$
- **Matrix size** $n = 200$

Corrector Predictor	Newton 1st-order	Newton 2nd-order	BFGS 1st-order	BFGS 2nd-order
major # it.	13.4	10.8	12.6	10.2
CPU	3252s	1529s	763s	585s
Newton # it.	27.0	19.6	-	-
BFGS # it.	-	-	210.2	180.0
Cholesky	285.4	165.8	795.8	567.8
CG	-	-	188.4	177.2
$\kappa(\boldsymbol{\nabla}^2 \boldsymbol{g}(\boldsymbol{y}, \mu))$	6.2e+7	3.4e+7	2.7e+7	2.2e+7
$\kappa(\boldsymbol{H} \boldsymbol{\nabla}^2 \boldsymbol{g}(\boldsymbol{y}, \mu))$	-	-	7.8e+1	8.6e+1

Stopping criterion

$$\textbf{relative error} = \frac{|\textbf{primal obj. - dual obj.}|}{\max\{\textbf{primal obj.}, 1.0\}} < 1.0e - 6$$

$$\textbf{primal feasibility error} = \max_{1 \le p \le m} |a_p - \boldsymbol{A}_p \bullet \boldsymbol{X}| < 1.0e - 6$$

Norm Minimization Problem

- **Average of** 5 **problems**
- **Matrix size** $n = 50$**, constraints** $m = 200$

Corrector Predictor	Newton 1st-order	Newton 2nd-order	BFGS 1st-order	BFGS 2nd-order
major # it.	14.8	12.6	14.2	12.6
CPU	843s	544s	240s	210s
Newton # it.	39.2	28.0	-	-
BFGS # it.	-	-	340.0	319.8
Cholesky	198.6	107.8	608.2	509.4
CG	-	-	228.2	262.2
$\kappa(\nabla^2 \boldsymbol{g}(\boldsymbol{y}, \mu))$	7.8e+9	9.2e+9	4.8e+9	1.2e+10
$\kappa(\boldsymbol{H}\nabla^2 \boldsymbol{g}(\boldsymbol{y}, \mu))$	-	-	3.3e+2	1.7e+3

Stopping criterion

$$\textbf{relative error} = \frac{|\textbf{primal obj. - dual obj.}|}{\max\{\textbf{primal obj.}, 1.0\}} < 1.0e-6$$

$$\textbf{primal feasibility error} = \max_{1 \le p \le m} |a_p - \boldsymbol{A}_p \bullet \boldsymbol{X}| < 1.0e-6$$

Typical result along the iterations of LDIPM

- **Box Constrained Quadratic** ± 1 **Program**
- **Matrix size** $n = 200$, **constraints** $m = 201$

k	μ^k	p.f.error	rel.error	$\kappa(\nabla^2 g)$	$\kappa(H^k \nabla^2 g)$	#CG 1	#CG 2
1	1.4e+1	9.81e−4	+2.81e+1	2.17e+2	3.03e+3	4	1
2	3.8e+0	1.61e−3	+1.87e+0	3.42e+2	9.00e+2	9	3
3	2.0e+0	1.59e−3	+6.17e−1	7.57e+2	6.75e+2	14	4
4	8.2e−1	1.07e−3	+1.92e−1	1.75e+3	1.07e+3	24	8
5	2.2e−1	1.14e−3	+4.62e−2	2.58e+3	2.86e+1	16	5
6	4.2e−2	7.92e−4	+8.48e−3	3.01e+3	5.94e+1	18	3
7	4.2e−3	4.13e−4	+8.47e−4	1.32e+4	1.93e+4	44	3
8	4.2e−4	3.82e−5	+8.47e−5	1.33e+5	1.62e+2	18	1
9	4.2e−5	3.29e−6	+8.43e−6	1.33e+6	3.37e+1	14	0
10	4.2e−6	4.11e−7	+8.45e−7	1.33e+7	5.18e+2	16	0

Summary

⟹ **New type of predictor-corrector dual IP method for SDP**

$$\begin{cases} \text{dual feasible, primal infeasible} \\ \boldsymbol{XS} = \mu \boldsymbol{I} \end{cases}$$

⟹ **(CORRECTOR Step)**

Quasi-Newton BFGS instead of Newton method

⟹ **(PREDICTOR Step)**

BFGS matrix $\boldsymbol{H}$ is a good preconditioner for the CG ($\boldsymbol{\nabla}^2 g(\boldsymbol{y}, \mu)$)

⟹ **Can be extended to Linear Optimization Problems over convex cones (LP, SOCP)**

Further Directions

⟹ **Limited memory BFGS for large scale problems**

⟹ **Improve numerical convergence**

⟹ **Implementation in C/C++**