Alternating Direction Augmented Lagrangian Algorithms for Convex Optimization

Donald Goldfarb

Joint work with Bo Huang, Shiqian Ma, Tony Qin, Katya Scheinberg, Zaiwen Wen and Wotao Yin

Department of IEOR, Columbia University

MOPTA Conference
Lehigh University
August 19, 2010

Alternating direction augmented Lagrangian methods

- Long history: goes back to Gabay and Mercier, Glowinski and Marrocco, Lions and Mercier, and Passty etc.
- Optimization models from partial differential equation
- Maximal monotone operators
- Variational inequalities
- Nonlinear convex optimization
- Linear programming
- Nonsmooth ℓ_1-minimization, compressive sensing

Outline

Outline:

1. ADAL methods for minimizing a convex function that can be written as a sum of simpler functions
2. ADAL methods for solving SDPs

Introduction

SUM-K	SUM-2
$\min F(x) \equiv \sum_{i=1}^{K} f_i(x)$	$\min F(x) \equiv f(x) + g(x)$

- Minimize the sum of convex functions
- Assume the following problem is easy

$$\min_x \quad \tau f_i(x) + \frac{1}{2}\|x - y\|^2$$

- Examples of f_i: $\|x\|_1$, $\|x\|_2$, $\|Ax - b\|^2$, $\|X\|_*$, $-\log\det(X)$, ...

Examples

- Compressed sensing:

$$\min \quad \rho\|x\|_1 + \frac{1}{2}\|Ax - b\|_2^2$$

- Matrix Rank Min:

$$\min \quad \rho\|X\|_* + \frac{1}{2}\|\mathcal{A}(X) - b\|_2^2$$

- Robust PCA:

$$\min_{X,Y} \quad \|X\|_* + \rho\|Y\|_1 : X + Y = M$$

- Sparse Inverse Covariance Selection:

$$\min -\log\det(X) + \langle \Sigma, X\rangle + \rho\|X\|_1$$

Variable Splitting

$$(SUM-2) \quad \min \quad f(x)+g(x)$$

- Variable splitting

$$\begin{array}{ll} \min & f(x)+g(y) \\ \text{s.t.} & x=y \end{array}$$

- Augmented Lagrangian function:

$$\mathcal{L}(x,y;\lambda) := f(x)+g(y)-\langle \lambda, x-y\rangle + \frac{1}{2\mu}\|x-y\|^2$$

- Augmented Lagrangian Method:

$$\begin{cases} (x^{k+1},y^{k+1}) & := \quad \arg\min_{(x,y)} \mathcal{L}(x,y;\lambda^k) \\ \lambda^{k+1} & := \quad \lambda^k - (x^{k+1}-y^{k+1})/\mu \end{cases}$$

Alternating Direction Augmented Lagrangian (ADAL)

- $\mathcal{L}(x, y; \lambda) := f(x) + g(y) - \langle \lambda, x - y \rangle + \frac{1}{2\mu}\|x - y\|^2$
- Solve augmented Lagrangian subproblem alternatingly

$$\begin{cases} x^{k+1} & := & \arg\min_x \mathcal{L}(x, y^k; \lambda^k) \\ y^{k+1} & := & \arg\min_y \mathcal{L}(x^{k+1}, y; \lambda^k) \\ \lambda^{k+1} & := & \lambda^k - (x^{k+1} - y^{k+1})/\mu \end{cases}$$

Symmetric ADAL

- $\mathcal{L}(x, y; \lambda) := f(x) + g(y) - \langle \lambda, x - y \rangle + \frac{1}{2\mu}\|x - y\|^2$
- Symmetric version

$$\begin{cases} x^{k+1} & := & \arg\min_x \mathcal{L}(x, y^k; \lambda^k) \\ \lambda^{k+\frac{1}{2}} & := & \lambda^k - (x^{k+1} - y^k)/\mu \\ y^{k+1} & := & \arg\min_y \mathcal{L}(x^{k+1}, y; \lambda^{k+\frac{1}{2}}) \\ \lambda^{k+1} & := & \lambda^{k+\frac{1}{2}} - (x^{k+1} - y^{k+1})/\mu \end{cases}$$

- Optimality conditions lead to (assuming f and g are smooth)

$$\lambda^{k+\frac{1}{2}} = \nabla f(x^{k+1}), \qquad \lambda^{k+1} = -\nabla g(y^{k+1})$$

Alternating Linearization Method

$$(SUM-2) \quad \min F(x) \equiv f(x) + g(x)$$

- Define

$$Q_g(u,v) := f(u) + g(v) + \langle \nabla g(v), u - v \rangle + \frac{1}{2\mu}\|u - v\|^2$$

$$Q_f(u,v) := f(u) + \langle \nabla f(u), v - u \rangle + \frac{1}{2\mu}\|u - v\|^2 + g(v)$$

- Alternating Linearization Method (ALM)

$$\begin{cases} x^{k+1} & := & \arg\min_x Q_g(x, y^k) \\ y^{k+1} & := & \arg\min_y Q_f(x^{k+1}, y) \end{cases}$$

- Proposed by Kiwiel, Rosa and Ruszczynski for stochastic programming.
- Gauss-Seidel like algorithm

Complexity Bound of ALM

Theorem (Goldfarb, Ma and Scheinberg, 2009)

Assume ∇f and ∇g are Lipschitz continuous with constants $L(f)$ and $L(g)$. For $\mu \leq 1/\max\{L(f), L(g)\}$, ALM satisfies

$$F(y^k) - F(x^*) \leq \frac{\|x^0 - x^*\|^2}{4\mu k}$$

Therefore,

- convergence in objective value
- $O(1/\epsilon)$ iterations for an ϵ-optimal solution

- The first complexity result for splitting and alternating direction type methods
- Question: Can we improve the complexity ?

Optimal Gradient Methods

$\min f(x)$ (assuming ∇f is Lipschitz continuous)

- ϵ-optimal solution $f(x) - f(x^*) \leq \epsilon$
- Classical gradient method

$$x^k = x^{k-1} - \tau_k \nabla f(x^{k-1})$$

Complexity $O(1/\epsilon)$

- Nesterov's acceleration technique (1983)

$$\begin{cases} x^k & := & y^{k-1} - \tau_k \nabla f(y^{k-1}) \\ y^k & := & x^k + \frac{k-1}{k+2}(x^k - x^{k-1}) \end{cases}$$

Complexity $O(1/\sqrt{\epsilon})$

- Optimal first-order method; best one can get

ISTA and FISTA (Beck and Teboulle, 2009)

- Assume g is smooth

$$\min \quad F(x) \equiv f(x) + g(x)$$

- Fixed Point Algorithm (Also called ISTA in compressed sensing)

$$x^{k+1} := \arg\min_x Q_f(x, x^k)$$

or equivalently

$$x^{k+1} := \arg\min_x \tau f(x) + \frac{1}{2}\|x - (x^k - \tau \nabla g(x^k))\|^2$$

- Never minimize g
- Iteration complexity: $O(1/\epsilon)$ for an ϵ-optimal solution $(F(x^k) - F(x^*) \leq \epsilon)$

ISTA and FISTA (Beck and Teboulle, 2009)

$$\min F(x) \equiv f(x) + g(x)$$

- Fast ISTA (FISTA)

$$\begin{cases} x^k & := & \arg\min_x \tau f(x) + \frac{1}{2}\|x - (y^k - \tau \nabla g(y^k))\|^2 \\ t_{k+1} & := & \left(1 + \sqrt{1 + 4t_k^2}\right)/2 \\ y^{k+1} & := & x^k + \frac{t_k - 1}{t_{k+1}}(x^k - x^{k-1}) \end{cases}$$

Complexity $O(1/\sqrt{\epsilon})$

Fast Alternating Linearization Method

- ALM

$$\begin{cases} x^{k+1} & := & \arg\min_x Q_g(x, y^k) \\ y^{k+1} & := & \arg\min_y Q_f(x^{k+1}, y) \end{cases}$$

- Accelerate ALM in the same way as FISTA
- Fast alternating linearization method (FALM)

$$\begin{cases} x^k & := & \arg\min_x Q_g(x, z^k) \\ y^k & := & \arg\min_y Q_f(x^k, y) \\ w^k & := & (x^k + y^k)/2 \\ t_{k+1} & := & \left(1 + \sqrt{1 + 4t_k^2}\right)/2 \\ z^{k+1} & := & w^k + \frac{1}{t_{k+1}}(t_k(y^k - w^{k-1}) - (w^k - w^{k-1})) \end{cases}$$

- computational effort at each iteration is almost unchanged

Complexity Bound for FALM

$$\min F(x) \equiv f(x) + g(x)$$

Theorem (Goldfarb, Ma and Scheinberg, 2009)

Assume ∇f and ∇g are Lipschitz continuous with constants $L(f)$ and $L(g)$. For $\mu \leq 1/\max\{L(f), L(g)\}$, FALM satisfies

$$F(y^k) - F(x^*) \leq \frac{\|x^0 - x^*\|^2}{\mu(k+1)^2}$$

Therefore,

- convergence in objective value
- $O(1/\sqrt{\epsilon})$ iterations for an ϵ-optimal solution
- Optimal first-order method

ALM with skipping steps

At k-th iteration of ALM-S:

- $x^{k+1} := \arg\min_x \mathcal{L}_\mu(x, y^k; \lambda^k)$
- If $F(x^{k+1}) > \mathcal{L}_\mu(x^{k+1}, y^k; \lambda^k)$, then $x^{k+1} := y^k$
- $y^{k+1} := \arg\min_y Q_f(y, x^{k+1})$
- $\lambda^{k+1} := \nabla f(x^{k+1}) - (x^{k+1} - y^{k+1})/\mu$

Note that only one function is required to be smooth.

Complexity result of ALM-S

Theorem (Goldfarb and Scheinberg, 2010)

Assume ∇f is Lipschitz continuous. For $\mu \leq 1/L(f)$, the iterates y^k in ALM-S satisfy:

$$F(y^k) - F(x^*) \leq \frac{\|x^0 - x^*\|^2}{2\mu(k + k_{ns})}, \forall k,$$

where k_{ns} is the number of iterations until the k-th for which $F(x^{k+1}) \leq \mathcal{L}_\mu(x^{k+1}, y^k; \lambda^k)$. Thus $O(1/\epsilon)$ iterations to obtain an ϵ-optimal solution.

Similar algorithm can be designed for FALM with $O(1/\sqrt{\epsilon})$ complexity and only one function is required to be smooth.

Complexity result of ADAL

Theorem (Goldfarb and Huang, 2010)

Assume both ∇f and ∇g are Lipschitz continuous. For $\mu \leq 1/\max\{L(f), L(g)\}$, ADAL needs $O(1/\epsilon)$ iterations to obtain an ϵ-optimal solution.

- Thus far we have been unable to develop a fast $O(1/\sqrt{\epsilon})$ version of ADAL.

SUM-K

From P.L.Lions and B.Mercier's 1979 paper on operator splitting

- Generalization from 2 to K is possible, but
- It seems proving convergence for $K \geq 3$ is "difficult".

$$\min \quad F(x) \equiv f(x) + g(x) + h(x)$$

- Define

$$Q_{gh}(u,v,w) := f(u) + g(v) + \langle \nabla g(v), u - v \rangle + \|u - v\|^2/2\mu \\ + h(w) + \langle \nabla h(w), u - w \rangle + \|u - w\|^2/2\mu.$$

$$\begin{cases} x^{k+1} & := & \arg\min Q_{gh}(x, y^k, z^k) \\ y^{k+1} & := & \arg\min Q_{fh}(x^{k+1}, y, z^k) \\ z^{k+1} & := & \arg\min Q_{fg}(x^{k+1}, y^{k+1}, z) \end{cases}$$

- However, no complexity results for Gauss-Seidel like algorithm!

MSA: A Jacobi Type Algorithm

$$\min \quad F(x) \equiv f(x) + g(x) + h(x)$$

- Multiple Splitting Algorithm (MSA)

$$\begin{cases} x^{k+1} & := & \arg\min Q_{gh}(x, w^k, w^k) \\ y^{k+1} & := & \arg\min Q_{fh}(w^k, y, w^k) \\ z^{k+1} & := & \arg\min Q_{fg}(w^k, w^k, z) \\ w^{k+1} & := & (x^{k+1} + y^{k+1} + z^{k+1})/3 \end{cases}$$

- Jacobi type algorithm
- Can be done in parallel
- We have complexity result!

Complexity Bound for MSA

$$\min F(x) \equiv f(x) + g(x) + h(x)$$

Theorem (Goldfarb and Ma, 2009)

Assume ∇f, ∇g and ∇h are Lipschitz continuous with constants $L(f)$, $L(g)$ and $L(h)$. For $\mu \leq 1/\max\{L(f), L(g), L(h)\}$, MSA satisfies

$$\min\{F(x^k), F(y^k), F(z^k)\} - F(x^*) \leq \frac{\|x_0 - x^*\|^2}{\mu k}.$$

Therefore,

- convergence in objective value
- $O(1/\epsilon)$ iterations for an ϵ-optimal solution

Fast Multiple Splitting Algorithm (FaMSA)

$$\begin{cases} x^k & := & \arg\min Q_{gh}(x, w_x^k, w_x^k) \\ y^k & := & \arg\min Q_{fh}(w_y^k, y, w_y^k) \\ z^k & := & \arg\min Q_{fg}(w_z^k, w_z^k, z) \\ w^k & := & (x^k + y^k + z^k)/3 \\ t_{k+1} & := & \left(1 + \sqrt{1 + 4t_k^2}\right)/2 \\ w_x^{k+1} & := & w^k + \frac{1}{t_{k+1}}[t_k(x^k - w^k) - (w^k - w^{k-1})] \\ w_y^{k+1} & := & w^k + \frac{1}{t_{k+1}}[t_k(y^k - w^k) - (w^k - w^{k-1})] \\ w_z^{k+1} & := & w^k + \frac{1}{t_{k+1}}[t_k(z^k - w^k) - (w^k - w^{k-1})] \end{cases}$$

Complexity Bound for FaMSA

$$\min F(x) \equiv f(x) + g(x) + h(x)$$

Theorem (Goldfarb and Ma, 2009)

Assume ∇f, ∇g and ∇h are Lipschitz continuous with constants $L(f)$, $L(g)$ and $L(h)$. For $\mu \leq 1/\max\{L(f), L(g), L(h)\}$, FaMSA satisfies

$$\min\{F(x^k), F(y^k), F(z^k)\} - F(x^*) \leq \frac{4\|x_0 - x^*\|^2}{\mu(k+1)^2}$$

Therefore,

- convergence in objective value
- $O(1/\sqrt{\epsilon})$ iterations for an ϵ-optimal solution
- optimal first-order method

Comparison of ALM/FALM and MSA/FaMSA

ALM/FALM

- Gauss-Seidel like algorithms
- expected to be faster than MSA/FaMSA since the information from current iteration is used
- complexity results for (SUM-2), no results for (SUM-K) when $K \geq 3$
- only one function needs to be smooth

MSA/FaMSA

- Jacobi like algorithms
- can be done in parallel
- complexity results for (SUM-K) for any $K \geq 2$

ALM and FALM for Robust PCA

- Robust PCA: $f(X) = \|X\|_*, \quad g(Y) = \rho\|Y\|_1$

$$\min_{X,Y\in\mathbb{R}^{m\times n}} \quad \{\|X\|_* + \rho\|Y\|_1 : X + Y = M\}$$

- Subproblem wrt X (a matrix shrinkage operator, corresponds to an SVD):

$$X^{k+1} := \arg\min_X f(X) + g(Y^k) + \langle\nabla g(Y^k), M - X - Y^k\rangle + \|X + Y^k - M\|_F^2/2\mu$$

- Subproblem wrt Y (a vector shrinkage operator):

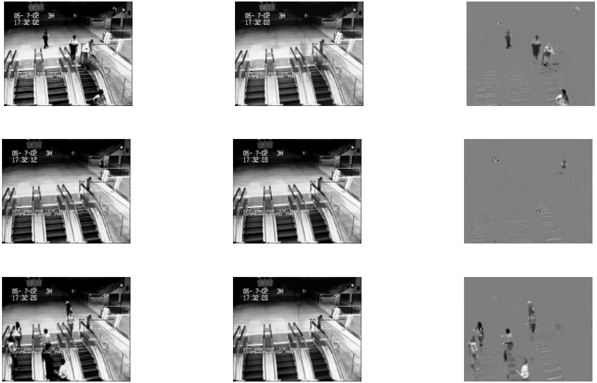
$$Y^{k+1} := \arg\min_Y f(X^{k+1}) + \langle\nabla f(X^{k+1}), M - X^{k+1} - Y\rangle + \|X^{k+1} + Y - M\|_F^2/2\mu + g(Y)$$

Numerical Results

- Surveillance video

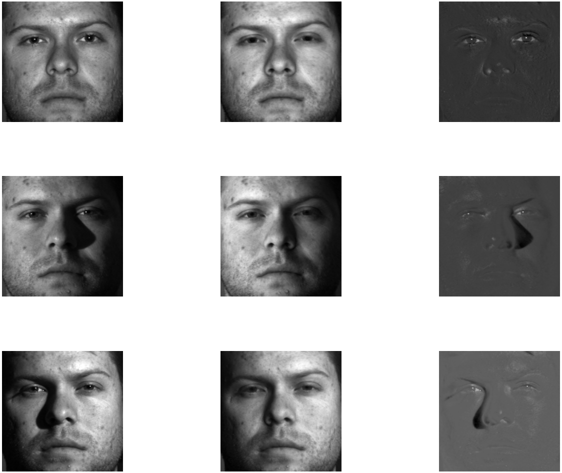
Numerical Results (cont.)

- Surveillance video



Numerical Results (cont.)

- Shadow and specularities removal from face images



Numerical Results (cont.)

- Video denoising

Summary of Numerical Results

Problem	n_1	n_2	SVDs	CPU
Hall	25344	200	43	04:03
Escalator	20800	300	45	05:53
Campus (color)	20480	960	46	01:01:01
Face	40000	65	42	01:39
Video denoising	25344	900	44	01:12:49

Conclusion and future work

Our contribution

- Alternating linearization and multiple splitting methods
- Optimal first-order methods
- First complexity results for splitting and alternating direction methods

Current and Future work

- Extension of ALM/FALM, MSA/FaMSA to constrained problems
- Extension of MSA/FaMSA to nonsmooth problems
- Development of Fast ADAL with $O(1/\sqrt{(}\epsilon))$ complexity
- Line search variants
- Applications in many fields such as Medical Imaging, Machine Learning, Model Selection, Optimal acquisition basis selection (radar), etc.

$$(D) \quad \begin{cases} \min_{y\in\mathbb{R}^m, S\in S^n} & -b^\top y \\ \text{s.t.} & \mathcal{A}^*(y) + S = C, \quad S \succeq 0, \end{cases}$$

- $\mathcal{A}(X) := \left(\left\langle A^{(1)}, X\right\rangle, \cdots, \left\langle A^{(m)}, X\right\rangle\right)^\top$
- $\mathcal{A}^*(y) := \sum_{i=1}^m y_i A^{(i)}$
- Let $A := \left(\mathrm{vec}(A^{(1)}), \cdots, \mathrm{vec}(A^{(m)})\right)^\top \in \mathbb{R}^{m\times n^2}$.
- Constraints: $\mathcal{A}(X) = b \Longrightarrow A\mathrm{vec}(X) = b$
- Operators related to $\mathcal{A}(\cdot)$:

$$\mathcal{A}(\mathcal{A}^*(y)) := (AA^\top)y, \quad \mathcal{A}^*(\mathcal{A}(X)) := \mathrm{mat}\left(\left(A^\top A\right)\mathrm{vec}(X)\right)$$

- Assumption: The matrix A has full row rank and the Slater condition holds for the primal SDP.

Augmented Lagrangian method

Augmented Lagrangian function:

$$\mathcal{L}_\mu(X,y,S) = -b^\top y + \langle X, \mathcal{A}^*(y) + S - C\rangle + \frac{1}{2\mu}\|\mathcal{A}^*(y)+S-C\|_F^2.$$

Algorithmic framework:

- compute y^{k+1} and S^{k+1} at k-th iteration

$$(DL) \qquad \min_{y\in\mathbb{R}^m, S\in S^n} \mathcal{L}_\mu(X^k,y,S), \qquad \text{s.t.} \qquad S \succeq 0.$$

- update the primal variable X^{k+1} by

$$X^{k+1} := X^k + \frac{\mathcal{A}^*(y^{k+1}) + S^{k+1} - C}{\mu}.$$

Pros and Cons:

- Pros: rich theory, well understood and a lot of algorithms
- Cons: $\mathcal{L}_\mu(X^k,y,S)$ is not separable in y and S, and the subproblem (DL) is difficult to minimize

Alternating direction method

- Divide variables into different blocks according to their roles
- Minimize the augmented Lagrangian function with respect to one block at a time while all other blocks are fixed

Framework

$$
\begin{aligned}
y^{k+1} &:= \arg\min_{y \in \mathbb{R}^m} \mathcal{L}_\mu(X^k, y, S^k), \\
S^{k+1} &:= \arg\min_{S \in S^n} \mathcal{L}_\mu(X^k, y^{k+1}, S), \quad S \succeq 0, \\
X^{k+1} &:= X^k + \frac{\mathcal{A}^*(y^{k+1}) + S^{k+1} - C}{\mu}.
\end{aligned}
$$

Alternating direction method

Solving subproblem: $y^{k+1} = \arg\min_{y\in\mathbb{R}^m} \mathcal{L}_\mu(X^k, y, S^k)$

- First-order optimality condition:

$$\nabla_y \mathcal{L}_\mu := \mathcal{A}(X^k) - b + \frac{1}{\mu}\mathcal{A}(\mathcal{A}^*(y^{k+1}) + S^k - C) = 0.$$

- Since $\mathcal{A}\mathcal{A}^*$ is invertible,

$$y^{k+1} := y(S^k, X^k) := -(\mathcal{A}\mathcal{A}^*)^{-1}\left(\mu(\mathcal{A}(X^k) - b) + \mathcal{A}(S^k - C)\right).$$

- Compute $\mathcal{A}\mathcal{A}^*$ prior to executing the algorithm.
- The operator $\mathcal{A}\mathcal{A}^*$ is the identity for many SDP relaxations of combinatorial optimization problems, such as the maxcut, maximum stable set and bisection problems

Alternating direction method

Solving subproblem $S^{k+1} = \arg\min_{S \in S^n} \mathcal{L}_\mu(X^k, \ y^{k+1}, \ S), S \succeq 0$

- Equivalent formulation:

$$\min_{S \in S^n} \quad \left\| S - V^{k+1} \right\|_F^2, \quad S \succeq 0,$$

  where $V^{k+1} := V(S^k, X^k) = C - \mathcal{A}^*(y(S^k, X^k)) - \mu X^k$.

- Hence, the solution is

$$S^{k+1} := V_\dagger^{k+1} := Q_\dagger \Sigma_+ Q_\dagger^\top$$

  where

$$V^{k+1} = Q \Sigma Q^\top = \begin{pmatrix} Q_\dagger & Q_\ddagger \end{pmatrix} \begin{pmatrix} \Sigma_+ & \mathbf{0} \\ \mathbf{0} & \Sigma_- \end{pmatrix} \begin{pmatrix} Q_\dagger^\top \\ Q_\ddagger^\top \end{pmatrix}$$

Alternating direction method

Updating the Lagrange multiplier X^{k+1}

- Updating formula:

$$X^{k+1} := X^k + \frac{\mathcal{A}^*(y^{k+1}) + S^{k+1} - C}{\mu}$$

- Equivalent formulation:

$$X^{k+1} = \frac{1}{\mu}(S^{k+1} - V^{k+1}) = \frac{1}{\mu} V_{\ddagger}^{k+1},$$

  where $V_{\ddagger}^{k+1} := -Q_{\ddagger}\Sigma_{-}Q_{\ddagger}$.

- Note that X^{k+1} is also the optimal solution of

$$\min_{X \in S^n} \quad \left\|\mu X + V^{k+1}\right\|_F^2, \quad X \succeq 0.$$

Convergence results

- Let $\mathcal{P}(V) := \left(V_{\dagger}, V_{\ddagger}\right)$
- Fixed point formulation:

$$\begin{aligned} y^{k+1} &= y(S^k, X^k) \\ \left(S^{k+1}, \mu X^{k+1}\right) &= \mathcal{P}(V^{k+1}) = \mathcal{P}(V(S^k, X^k)). \end{aligned}$$

- Equivalence of the optimality conditions:
  - (X, y, S) satisfies the KKT optimality conditions

$$\mathcal{A}(X) = b, \quad \mathcal{A}^*(y) + S = C, \quad SX = 0, \quad X \succeq 0, \quad Z \succeq 0.$$

  - (X, y, S) satisfies

$$y = y(S, X) \text{ and } \left(S, \mu X\right) = \mathcal{P}(V(S, X)).$$

- Main result: the sequence $\{(X^k, y^k, S^k)\}$ generated by our algorithm from any starting point converges to a solution $(X^*, y^*, S^*) \in \mathcal{X}^*$, where $\mathcal{X}^*$ is the set of primal and dual solution pairs. No complexity results currently known.

Convergence results

- For any $V, \widehat{V} \in S^n$, $\left\| \mathcal{P}(V) - \mathcal{P}(\widehat{V}) \right\|_F \leq \|V - \widehat{V}\|_F$, with equality holding if and only if $V_\dagger^\top \widehat{V}_\ddagger = 0$ and $V_\ddagger^\top \widehat{V}_\dagger = 0$.
- For any $S, X, \widehat{S}, \widehat{X} \in S_+^n$,

$$\|V(S, X) - V(\widehat{S}, \widehat{X})\|_F \leq \left\| \left(S - \widehat{S}, \mu(X - \widehat{X}) \right) \right\|_F$$

  with equality holding if and only if

$$V - \widehat{V} = (S - \widehat{S}) - \mu(X - \widehat{X}).$$

- Let (X^*, y^*, S^*), be an optimal solution. If

$$\|\mathcal{P}(V(S, X)) - \mathcal{P}(V(S^*, X^*))\|_F = \left\| (S - S^*, \mu(X - X^*)) \right\|_F,$$

  then, $(S, \mu X)$ is a fixed point, and hence, (X, y, S) is a primal and dual optimal solution pair.

An expanded problem

- Primal problem:

$$\begin{aligned} \min \quad & \langle C, X \rangle , \\ \text{s.t.} \quad & \mathcal{A}(X) = b, \quad \mathcal{B}(X) \geq d, \\ & X \succeq 0, \quad X \geq 0, \end{aligned}$$

where $\mathcal{B}(X) := \left(\left\langle B^{(1)}, X \right\rangle, \cdots, \left\langle B^{(q)}, X \right\rangle \right)^\top, B^{(i)} \in S^n$.

- Dual problem

$$\begin{aligned} \min_{y \in \mathbb{R}^m, v \in \mathbb{R}^q, S \in S^n, Z \in S^n} \quad & -b^\top y - d^\top v, \\ \text{s.t.} \quad & \mathcal{A}^*(y) + \mathcal{B}^*(v) + S + Z = C, \\ & v \geq 0, \quad S \succeq 0, \quad Z \geq 0 \end{aligned}$$

A multiple splitting scheme

- Augmented Lagrangian function

$$\mathcal{L}_\mu = -b^\top y - d^\top v + \langle X, \mathcal{A}^*(y) + \mathcal{B}^*(v) + S + Z - C \rangle + \frac{1}{2\mu} \| \mathcal{A}^*(y) + \mathcal{B}^*(v) + S + Z - C \|_F^2,$$

Framework

$$y^{k+1} := \arg\min_{y \in \mathbb{R}^m} \mathcal{L}_\mu(X^k, y, v^k, Z^k, S^k),$$

$$v^{k+1} := \arg\min_{v \in \mathbb{R}^q} \mathcal{L}_\mu(X^k, y^{k+1}, v, Z^k, S^k), \quad v \geq 0,$$

$$Z^{k+1} := \arg\min_{Z \in S^n} \mathcal{L}_\mu(X^k, y^{k+1}, v^{k+1}, Z, S^k), \quad Z \geq 0,$$

$$S^{k+1} := \arg\min_{S \in S^n} \mathcal{L}_\mu(X^k, y^{k+1}, v^{k+1}, Z^{k+1}, S), \quad S \succeq 0,$$

$$X^{k+1} := X^k + \frac{\mathcal{A}^*(y^{k+1}) + \mathcal{B}^*(v^{k+1}) + S^{k+1} + Z^{k+1} - C}{\mu}.$$

Eigenvalue decomposition

- Partial eigenvalue decomposition:
  - X and S share the same set of eigenvectors since $XS = 0$.
  - Only need to compute either $V_{\dagger}^{k}$ or $V_{\ddagger}^{k}$.
  - Adaptive scheme (suppose that $V_{\dagger}^{k-1}$ is known)
    - $\kappa_{+}(V^{k-1})$: the total number of the pos. eigenvalues of V^{k-1}.
    - compute $V_{\dagger}^{k}$ for S^{k} if $\kappa_{+}(V^{k-1}) \leq \frac{n}{2}$
    - otherwise, compute $V_{\ddagger}^{k}$ for X^{k}.
- Direct and iterative methods
  - "DSYEVX" in LAPACK and "ARPACK"
- Warm start techniques
  - $V_{\dagger}^{k}$ is the optimal solution of $\min_{S \succeq 0} \|S - V^{k}\|_F^2$
  - Nonlinear programming approaches:

$$R^{k+1} := \arg \min_{R \in \mathbb{R}^{n \times \kappa}} \|RR^\top - V^{k+1}\|_F^2$$

starting from $R := Q_{\dagger}^{k}(\Sigma_{+}^{k})^{\frac{1}{2}}$,

Adjusting penalty parameter μ

- Tune μ so that the primal and dual infeasibilities are balanced
- The primal and dual infeasibilities are proportional to $\frac{1}{\mu}$ and μ, respectively.
- We decrease (increase) μ by a factor γ, $0 < \gamma < 1$, if the primal infeasibility is less than (greater than) a multiple of the dual infeasibility for a number of consecutive iterations.
- μ is required to remain within an interval $[\mu_{\text{min}}, \mu_{\text{max}}]$, where $0 < \mu_{\text{min}} < \mu_{\text{max}} < \infty$.

Other practical issues

- step size for updating X:

$$
\begin{aligned}
X^{k+1} &= X^k + \rho \frac{\mathcal{A}^*(y^{k+1}) + S^{k+1} - C}{\mu} \\
&= (1-\rho)X^k + \frac{\rho}{\mu}(S^{k+1} - V^{k+1}) = (1-\rho)X^k + \rho \bar{X}^{k+1},
\end{aligned}
$$

- termination rule:
  - optimality and feasiblities:

$$
\text{pinf} = \frac{\|\mathcal{A}(X) - b\|_2}{1 + \|b\|_2}, \text{dinf} = \frac{\|C + S + Z - \mathcal{A}^*(y)\|_F}{1 + \|C\|_1}, \text{gap} = \frac{|b^\top y - \langle C, X \rangle|}{1 + |b^\top y| + \langle C, X \rangle}.
$$

  - Stop when $\delta := \max\{\text{pinf, dinf, gap}\} \le \epsilon$
  - Stagnation: "it_stag" is an iteration counter

$$
(\text{it\_stag} > h_1 \text{ and } \delta <= 10\epsilon) \text{ or } (\text{it\_stag} > h_2 \text{ and } \delta <= 10^2\epsilon) \text{ or } (\text{it\_stag} > h_3 \text{ and } \delta <= 10^3\epsilon),
$$

  where $1 < h_1 < h_2 < h_3$ are integers representing different levels of difficulty.

Numerical Results

- Frequency assignment problem
- Maximal stable set problem
- Comparison with SDPNAL (Zhao, Sun and Toh)
  - A semi-smooth Newton-CG method for minimizing the dual augmented Lagrangian function
- Codes were written in C Language MEX-files in MATLAB (Release 7.3.0) and all experiments were performed on a Dell Precision 670 workstation with an Intel Xeon 3.4GHZ CPU and 6GB of RAM.

Numerical Results: frequency assignment problems

- Graph: $G = (V, E)$; Weight matrix $W \in S^r$
- Formulation:

$$\begin{array}{ll} \min & \langle \frac{1}{2k}\mathrm{diag}(We) + \frac{k-1}{2k} W, X \rangle \\ s.t. & X_{i,j} \geq \frac{-1}{k-1}, \quad \forall (i,j) \in E \backslash T, \\ & X_{i,j} = \frac{-1}{k-1}, \quad \forall (i,j) \in T, \\ & \mathrm{diag}(X) = e, \quad X \succeq 0. \end{array}$$

- Operator $\mathcal{A}$: the edges in T and $\mathrm{diag}(X) = e$
- Operator $\mathcal{B}$: the edges in $E \backslash T$
- Scaling: $X_{ij}/\sqrt{2} = \frac{-1}{\sqrt{2}(k-1)}$ and $X_{ij}/\sqrt{2} \geq \frac{-1}{\sqrt{2}(k-1)}$

$$\begin{aligned} y^{k+1} &= -\left(\mu(\mathcal{A}(X^k) - b) + \mathcal{A}(S^k - C)\right) \\ v^{k+1} &= \max\left(-\left(\mu(\mathcal{B}(X^k) - d) + \mathcal{B}(S^k - C)\right), \mathbf{0}\right). \end{aligned}$$

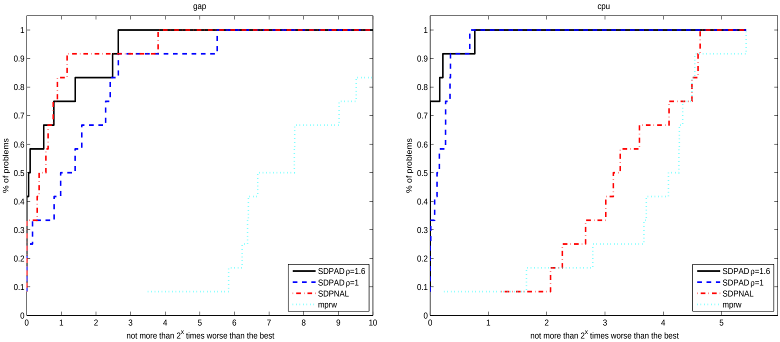
Numerical Results: frequency assignment problems

Table: Computational results

			ADAL					SDPNAL	
name	n	$\hat{m}$	pinf	dinf	itr	gap	cpu	gap	cpu
fap01	52	1378	1.72e-07	3.48e-07	610	7.76e-07	0.35	1.39e-7	6.31
fap02	61	1866	1.24e-06	8.77e-06	1121	9.61e-06	0.83	1.15e-5	4.12
fap03	65	2145	5.89e-06	4.30e-06	840	1.48e-06	0.73	2.27e-6	7.44
fap04	81	3321	1.56e-07	3.10e-07	929	9.58e-06	1.58	1.53e-5	19.69
fap05	84	3570	3.49e-06	3.76e-06	768	8.84e-06	1.29	1.09e-5	31.59
fap06	93	4371	3.17e-06	2.50e-06	568	9.71e-06	1.20	1.73e-5	29.84
fap07	98	4851	4.54e-06	4.95e-06	543	9.89e-06	1.24	5.75e-6	29.88
fap08	120	7260	1.85e-06	1.06e-06	699	4.04e-06	2.60	5.93e-6	25.25
fap09	174	15225	7.99e-07	9.98e-07	505	2.06e-07	4.64	2.86e-6	59.67
fap10	183	14479	3.22e-06	5.46e-06	1250	4.96e-04	14.29	7.93e-5	1:50
fap11	252	24292	3.26e-06	3.21e-06	1654	4.99e-04	48.01	1.89e-4	5:15
fap12	369	26462	4.70e-07	4.94e-07	5000	9.42e-05	6:39	1.60e-4	13:14

Numerical Results: frequency assignment problems

Figure: Performance profiles of two variants of ADAL, SDPNAL and mprw



(a) gap (b) cpu

Numerical Results: computing $\theta(G)$ and $\theta_+(G)$

- Formulation:

$$
\begin{aligned}
\theta(G) := \max \quad & \left\langle ee^\top, X \right\rangle \\
\text{s.t.} \quad & X_{ij} = 0, \quad (i,j) \in E, \quad \langle I, X \rangle = 1, \\
& X \succeq 0,
\end{aligned}
$$

$$
\begin{aligned}
\theta_+(G) := \max \quad & \left\langle ee^\top, X \right\rangle \\
\text{s.t.} \quad & X_{ij} = 0, \quad (i,j) \in E, \quad \langle I, X \rangle = 1, \\
& X \succeq 0, \quad X \geq 0,
\end{aligned}
$$

- Scaling: $X_{ij}/\sqrt{2} = 0$ and $\frac{1}{\sqrt{n}} \langle I, X \rangle = \frac{1}{\sqrt{n}}$.

Numerical Results: computing $\theta(G)$

Table: Computational results on computing $\theta(G)$

			ADAL					SDPNAL	
name	n	$\hat{m}$	pinf	dinf	itr	gap	cpu	gap	cpu
theta102	500	37467	8.76e-7	6.27e-7	256	9.67e-7	47	1.6e-8	50
theta103	500	62516	2.85e-7	9.42e-7	257	3.33e-7	57	4.6e-8	1:00
theta104	500	87245	3.40e-7	9.72e-7	260	6.53e-7	48	7.6e-8	58
theta123	600	90020	3.42e-7	9.71e-7	263	4.57e-7	1:43	4.1e-8	1:34
MANN-a27	378	703	8.45e-7	3.50e-6	503	8.81e-6	27	8.3e-8	07
sanr200-0.7	200	6033	7.83e-7	9.98e-7	219	3.04e-7	04	1.4e-7	04
c-fat200-1	200	18367	1.00e-6	1.76e-7	302	9.15e-7	04	8.5e-8	09
ham-10-2	1024	23041	4.72e-7	3.16e-6	597	9.51e-6	22:8	9.0e-8	02
ham-8-3-4	256	16129	9.92e-8	9.94e-7	199	8.04e-7	05	1.3e-8	10
ham-9-5-6	512	53761	5.70e-7	4.51e-6	1000	1.52e-5	2:48	1.4e-6	1:33
brock400-1	400	20078	9.75e-7	8.06e-7	254	8.30e-7	25	1.7e-8	26
keller4	171	5101	5.06e-7	9.88e-7	249	9.32e-7	03	1.3e-8	05
p-hat300-1	300	33918	7.58e-7	6.81e-7	764	9.96e-7	37	5.3e-7	1:45
G43	1000	9991	2.84e-6	3.91e-6	935	7.68e-6	21:17	4.2e-8	1:33
2dc.512	512	54896	2.24e-5	2.57e-5	1000	2.87e-4	5:51	1.7e-4	32:16
1dc.1024	1024	24064	7.82e-5	5.10e-5	747	2.79e-4	24:26	2.9e-6	41:26
1et.1024	1024	9601	1.93e-4	1.32e-4	603	7.11e-4	20:03	1.8e-6	1:01:14
1tc.1024	1024	7937	4.16e-4	5.21e-4	611	9.93e-4	21:47	2.2e-6	1:48:04
1zc.1024	1024	16641	9.72e-7	3.78e-7	608	4.16e-7	23:15	3.3e-8	4:15
2dc.1024	1024	169163	1.85e-5	2.71e-5	1000	3.39e-4	49:05	9.9e-5	2:57:56
1dc.2048	2048	58368	2.82e-4	2.16e-4	473	9.99e-4	2:50:44	1.5e-6	6:11:11
1et.2048	2048	22529	1.78e-4	4.05e-4	904	9.93e-4	4:54:57	8.8e-7	7:13:55
1tc.2048	2048	18945	1.79e-4	4.48e-4	1000	1.08e-3	5:15:14	7.9e-6	9:52:09
1zc.2048	2048	39425	1.42e-6	3.10e-6	941	9.17e-6	6:39:07	1.2e-6	45:16
2dc.2048	2048	504452	1.88e-5	8.37e-6	1000	3.17e-4	7:12:50	4.4e-5	15:13:19

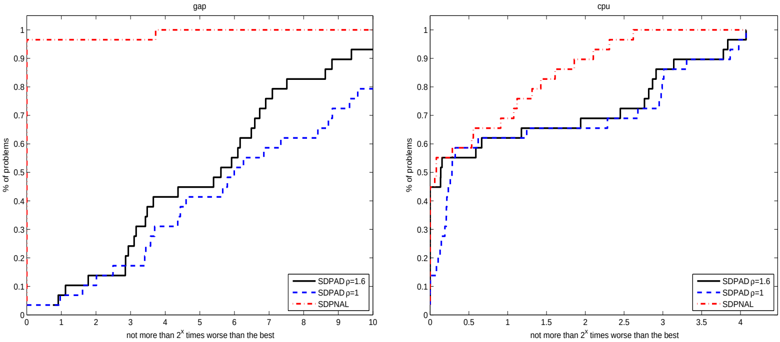
Numerical Results: computing $\theta_+(G)$

Table: Computational results on computing $\theta_+(G)$

			ADAL					SDPNAL	
name	n	$\hat{m}$	pinf	dinf	itr	gap	cpu	gap	cpu
theta102	500	37467	2.94e-7	9.58e-7	281	2.84e-7	1:01	8.4e-8	3:31
theta103	500	62516	3.27e-7	9.52e-7	262	4.85e-7	1:01	2.3e-8	3:28
theta104	500	87245	3.57e-7	9.55e-7	266	7.35e-7	52	1.6e-7	2:35
theta123	600	90020	3.77e-7	9.43e-7	267	6.02e-7	1:39	1.2e-7	6:44
MANN-a27	378	703	8.98e-7	4.08e-6	530	8.17e-6	32	1.6e-7	35
sanr200-0.7	200	6033	8.61e-7	9.57e-7	228	4.47e-7	05	2.9e-7	11
c-fat200-1	200	18367	4.71e-7	2.04e-7	306	8.40e-7	04	2.1e-7	36
ham-8-3-4	256	16129	9.91e-8	9.94e-7	199	8.05e-7	05	2.7e-10	05
ham-9-5-6	512	53761	6.49e-8	7.58e-7	472	3.25e-7	2:19	2.6e-7	42
ham-10-2	1024	23041	4.99e-8	5.49e-7	653	9.78e-7	27:58	4.2e-7	4:35
brock400-1	400	20078	9.98e-7	7.54e-7	258	9.83e-7	29	3.5e-9	1:45
keller4	171	5101	3.61e-6	9.01e-6	331	2.98e-6	05	3.7e-7	43
p-hat300-1	300	33918	1.34e-6	5.93e-7	567	1.50e-6	29	7.9e-7	6:50
G43	1000	9991	4.40e-6	5.45e-6	864	8.02e-6	20:22	2.1e-7	52:00
2dc.512	512	54896	9.43e-6	9.09e-6	1000	8.64e-5	5:10	3.8e-4	2:25:15
1dc.1024	1024	24064	2.13e-5	8.50e-6	1000	5.98e-5	57:20	1.4e-5	5:03:49
1et.1024	1024	9601	1.18e-4	2.16e-4	651	5.26e-4	36:04	1.1e-5	6:45:50
1tc.1024	1024	7937	1.65e-4	2.67e-4	799	3.53e-4	49:13	8.7e-4	10:37:57
1zc.1024	1024	16641	5.71e-5	1.80e-5	506	5.30e-4	28:22	1.6e-7	40:13
2dc.1024	1024	169163	3.81e-6	2.37e-6	1000	3.85e-5	50:34	7.3e-4	11:57:25
1dc.2048	2048	58368	2.01e-4	9.45e-5	517	9.99e-4	4:25:17	9.7e-5	35:52:44
1et.2048	2048	22529	1.76e-4	2.66e-4	659	6.80e-4	5:15:09	4.0e-5	80:48:17
1tc.2048	2048	18945	1.86e-4	3.97e-4	862	6.09e-4	6:35:33	1.4e-3	73:56:01
1zc.2048	2048	39425	3.58e-7	4.60e-6	953	8.27e-6	7:14:21	2.3e-7	2:13:04
2dc.2048	2048	504452	4.84e-6	2.79e-6	1000	4.95e-5	5:45:25	2.7e-3	45:21:42

Numerical Results: computing $\theta(G)$

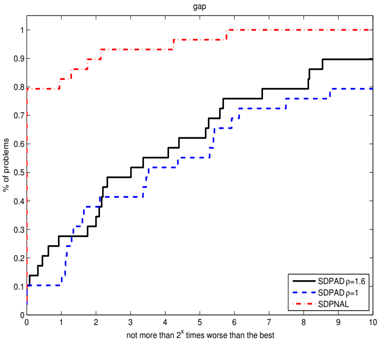
Figure: Performance profiles of two variants of ADAL and SDPNAL



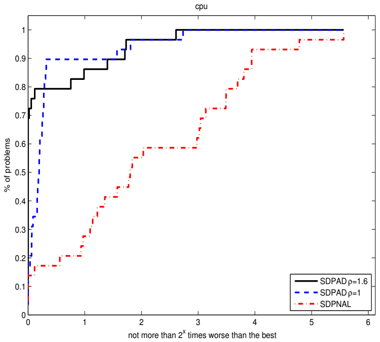
(a) gap (b) cpu

Numerical Results: computing $\theta_+(G)$

Figure: Performance profiles of two variants of ADAL and SDPNAL



(a) gap



(b) cpu

Numerical Results: BIQ problems

- The binary integer quadratic programming problem:

$$\min x^\top Qx, \text{ s.t. } x \in \{0,1\}^n$$

whose SDP relaxation is:

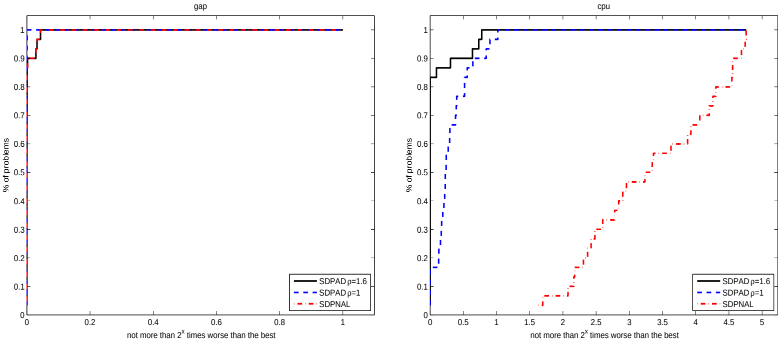
$$\begin{aligned} \min \quad & \left\langle \begin{pmatrix} Q & 0 \\ 0 & 0 \end{pmatrix}, X \right\rangle \\ s.t. \quad & X_{ii} - X_{n,i} = 0, i = 1, \cdots, n-1, \\ & X_{nn} = 1, \quad X \succeq 0, \quad X \geq 0, \end{aligned}$$

- Scaling: $\sqrt{\frac{2}{3}}(X_{ij} - X_{n,i}) = 0$
- "best upper bound" take from "Biq mac library" and

$$\%\text{dgap} \quad := \quad \left| \frac{\text{best upper bound} - \text{dobj}}{\text{best upper bound}} \right| \times 100\%.$$

Numerical Results: BIQ problems

Figure: Performance profiles of two variants of ADAL and SDPNAL



(a) %dgap (b) cpu