Bundle-type methods uniformly optimal for smooth and non-smooth convex optimization

Guanghui (George) Lan

Department of Industrial & Systems Engineering
University of Florida

Workshop on Numerical Methods on Conitnuous Optimization
IPAM, UCLA

October 12th, 2010

Outline

- Background and Motivation
- Review of Bundle-type Methods
- Accelerated Bundle-level Method (with full memory)
- Accelerated Prox-level Method (with limited memory)
- Extensions to Solve Strongly Convex Programming Problems
- Implementation and Numerical Illustration
- Summary

Basic convex programming (CP) problems and techniques

- Minimize $f^* \equiv \min_{x \in X} f(x)$, where X is a convex function and X is a simplex convex set.
  - f is represented by a *first-order oracle* to compute $f(x)$ and $g(x) \in \partial f(x)$.
- Two types of "black-box-oriented" optimizatin techniques:
  - techniques for unconstrained minimization of smooth convex functions, such as, Gradient Descent, Conjugate Gradient, Quasi-Newton, Nesterov's optimal method etc.
  - subgradient-type methods for constrained and/or non-smooth CP, such as, subgradient descent, mirror descent, cutting-plane methods (Kelley's method, level methods, analytical center) etc.

Certain dilemma

- Signifcantly better theoretical performance can be obtained for minimizing smooth CP problems such that $\|\nabla f(x) - \nabla f(x')\|_* \leq L\|x - x'\|, \forall x, x' \in X$.
  - In particular, Nesterov's method (1983) achieves the optimal rate of convergence, i.e., $f(x_t) - f^* \leq \mathcal{O}(1)LD_X^2/t^2$, where $D_X = \max_{x,x' \in X} \|x - x'\|$;
  - practical performance usually follows according to the theory;
- Some advanced non-smooth optimization techniques, such as, bundle-level methods (and certain variants with limited memory)
  - guaranteed rate of convergence $f(x_t) - f^* \leq \mathcal{O}(1)MD_X/\sqrt{t}$ for non-smooth CP, where $M = \sup_{x \in X} \|g(x)\|_*$;
  - linearly converged in practice for non-smooth CP: $MD_X \exp(-t/n)$ (see Ben-tal and Nemirovski 00);
  - rarely used for miniming smooth CP problems.

Some motivating questions

- Question#1: In view of the excellent practical performance of these advanced cutting plane methods for non-smooth CP, should we use them for solving smooth CP problems as well?
- Question#2: Could we achieve an optimal rate of convergence for smooth CP by cutting plane type methods?
- Question#3: Given that the first-order information is obtained via a black box, should an optimization algorithm really need to know any smoothness information, such as, whether a problem is smooth or not? and how big the Lipschitz constants L and M are?
  - Lan 09,10; Ghadimi and Lan 10 showed that Nesterov's method, after proper modification, is universally optimal for smooth, nonsmooth and stochastic CP. However, it does require the aformentioned global smoothness information.

Brief review of bundle-type methods

Let $x_1, \ldots, x_t \in X$ be given, an important construct, the *cutting plane model*, is given by

$$m_t(x) = \max \{ f(x_i) + \langle f'(x_i), x - x_i \rangle : 1 \leq i \leq t \}.$$

Clearly, $m_t(\cdot)$ underestimates $f(\cdot)$ for any $x \in X$.

- Kelley's (or the classical cutting-plane) method (Chenny and Goldstein 59, Kelley 60):

  $$x_{t+1} \in \operatorname{Argmin}_{x \in X} m_t(x),$$

  which is known to be very slow, both theoretically and practically (see for example, Nemirovski and Yudin 83).

Some refinements - bundle methods

Bundle methods (Kiwiel 83, 90, Lemaréchal 75, Mifflin 82)

$$x_{t+1} = \operatorname{argmin}_{x \in X} \left\{ m_t(x) + \frac{\rho_t}{2} \|x - x_t^+\|^2 \right\},$$

where the current prox-center x_t^+ is a certain point from $\{x_1, \ldots, x_t\}$ and γ_t is the current penalty:

- if $f(x_{t+1})$ has sufficiently decreased, set $x_{t+1}^+ = x_{t+1}$, o.w., set $x_{t+1}^+ = x_t^+$;
- the penalty reduces the influence of the model m_t's inaccuracy and hence the instabilities of the algorithm;
- it is usually hard to determine the penaly ρ_t, requiring certain on line adjustments or line-search;
- the related trust-region approach (e.g., Ruszczyński 03, Linderoth and Wright 03) encounters similar difficulties.

Bundle-level methods

Incorporating the level sets (Lemaréchal, Nemirovskii and Nesterov 95): choose a level l_t and set

$$x_{t+1} = \operatorname{argmin}_{x \in X} \left\{ \|x - x_t^+\|^2 : m_t(x) \leq l_t \right\}.$$

- compute $f_t = \min_{x \in X} m_t(x)$ and let f^t be the best objective value found so far;
- Set $\Delta_t = f^t - f_t$, and, for some $\lambda \in (0, 1)$, $l_t = \lambda f^t + (1 - \lambda) f_t = f_t + \lambda \Delta_t$;
- the prox-center x_t^+ can be chosen as the last iterate x_t;
- the rate of convergence: $f(x_t) - f^* \leq \mathcal{O}(1) C(\lambda) M D_X / \sqrt{t}$, where $C(\lambda)$ is a constant depending on λ.

Limited memory bundle-level methods

In the bundle-level methods, the number of constraints for the two subproblems

$$f_t = \min_{x \in X} m_t(x)$$
$$x_{t+1} = \operatorname{argmin}_{x \in X} \left\{ \|x - x_t^+\|^2 : m_t(x) \leq l_t \right\}$$

is equal to the size of the bundle.

- Kiwiel (95) in his Restricted Memory Prox-Level method presented novel rules of updating the prox-center, the bundle and the level, so as to eliminate the requirement of full memory;
- Truncated Proximal Bundle-level method and Non-Eucidean Restricted Memory Level Method (NERML) (Ben-tal and Nemirovski 00, 05):
  - the rate of convergence of the bundle-level methods is preserved.

Accelerated bundle-level (ABL) method

Objective: to solve CP problems $f^* \equiv \min_{x \in X} f(x)$ where X is a convex compact set and $f : X \to \Re$ is a closed convex function s.t.

$$f(y) - f(x) - \langle f'(x), y - x \rangle \leq \frac{L}{2}\|y - x\|^2 + 2M\|y - x\|, \quad \forall x, y \in X,$$

for some $L, M \geq 0$ and $f'(x) \in \partial f(x)$.

- For smooth CP problems, we have $M = 0$;
- For non-smooth CP problems, we have $L = 0$;
- Also covers CP problems given by the summation of smooth and non-smooth components;

The ABL algorithm - major modifications

- The sequence used to construct the model is different from the one used to compute the best objective value;
- Both sequences are taken as certain convex combinations of a perviously generated sequence which is also used as prox-centers;
- The steps of the algorithm are grouped into subsequence *segments*;
- Certain *stepsize rules* are used to determine how to take those convex combinations of the iterates in each segment;
- The stepsizes can be explicitly given without requiring any line search.

Initialization of the ABL algorithm

- Each step t, $t = 0, 1, 2, \ldots$, of segment s, $s = 1, 2, \ldots$, maintains three interwined sequences;
  - $x^{lb}_{s,t}$: search point to generate a lower bound on f^*;
  - $x^{ub}_{s,t}$: search point to generate an upper bound on f^*;
  - $x_{s,t}$: search point to be used as a prox-center.
- Initialization: given an initial point $p_0 \in X$,
  - set $x^{lb}_{1,0} = p_0$, compute $f(x^{lb}_{1,0})$, $f'(x^{lb}_{1,0})$.
  - Compute $\mathrm{lb}_{1,0} := \min_{x \in X} f(x^{lb}_{1,0}) + \langle f'(x^{lb}_{1,0}), x - x^{lb}_{1,0} \rangle$.
  - Let $x^{ub}_{1,0} \in \mathrm{Argmin}_{x \in X} f(x^{lb}_{1,0}) + \langle f'(x^{lb}_{1,0}), x - x^{lb}_{1,0} \rangle$ and $\mathrm{ub}_0 = f(x^{ub}_{1,0})$.
  - Let $x_{1,0} \in X$ be arbitrarily chosen, say $x_{1,0} = p_0$. Also let $\Delta_{1,0} = \mathrm{ub}_{1,0} - \mathrm{lb}_{1,0}$. [1]

[1] Note that essentially $x^{ub}_{1,0}$ can be arbitrarily chosen, as the initial gap $\Delta_{1,0}$ only logarithmically affects the convergence.

A step of the ABL algorithm

- Firstly, if $\mathrm{ub}_{s,t-1} - \mathrm{lb}_{s,t-1} \leq \epsilon$, **Terminate** the algorithm.
- Secondly, determine if a new segment should start, by checking if $\mathrm{ub}_{s,t-1} - \mathrm{lb}_{s,t-1} < \lambda \Delta_{s,0}$ holds for a given $\lambda \in (0, 1)$. If so, set

$$\begin{aligned} (x_{s+1,0}, x^{ub}_{s+1,0}, x^{lb}_{s+1,0}) &= (x_{s,t-1}, x^{ub}_{s,t-1}, x^{lb}_{s,t-1}), \\ \mathrm{lb}_{s+1,0} &= \mathrm{lb}_{s,t-1} \\ \mathrm{ub}_{s+1,0} &= \mathrm{ub}_{s,t-1} \\ \Delta_{s+1,0} &= \Delta_{s,t-1} \\ T_s &= t-1, \end{aligned}$$

  and pass to segment $s+1$, where T_s is used to count the number of steps in segment s.
- Otherwise, update $(x_{s,t-1}, x^{ub}_{s,t-1}, x^{lb}_{s,t-1})$ into $(x_{s,t}, x^{ub}_{s,t}, x^{lb}_{s,t})$ by going through the following procedure:

A step of the ABL algorithm: update the sequences

- Set $x^{lb}_{s,t} = (1-\alpha_t)\, x^{ub}_{s,t-1} + \alpha_t\, x_{s,t-1}$ for certain $\alpha_t \in (0,1]$.
- Set $\mathrm{lb}_{s,t} = \min_{x\in X} m_{s,t}(x)$, where $m_{s,t}(\cdot)$ is the model given by
$$\max\left\{ h(x^{lb}_{i,j}, x) : \begin{array}{l} j = 0, 1 \ldots, T_i - 1, \forall\, i = 1, \ldots, s-1, \\ j = 0, 1, \ldots, t \text{ if } i = s \end{array} \right\}.$$
- Compute the level $l_{s,t} = \lambda \mathrm{lb}_{s,t} + (1-\lambda)\,\mathrm{ub}_{s,t-1}$ and set
$$x_{s,t} = \mathrm{argmin}_{x\in X} \left\{ \|x - x_{s,t-1}\|^2 : m_{s,t}(x) \le l_{s,t} \right\}.$$
- Set $x^{ub}_{s,t} \in X$ s.t. $f(x^{ub}_{s,t}) \le \min\left\{ \mathrm{ub}_{s,t-1}, f(\alpha_t x_t + (1-\alpha_t) x^{ub}_{t-1}) \right\}$:
  - denoting $\tilde{x}^{ub}_{s,t} \equiv \alpha_t x_{s,t} + (1-\alpha_t) x^{ub}_{s,t-1}$, we can set $x^{ub}_{s,t} = \tilde{x}^{ub}_{s,t}$ if $f(\tilde{x}^{ub}_{s,t}) \le \mathrm{ub}_{s,t-1}$. Otherwise, set $x^{ub}_{s,t} = x^{ub}_{s,t-1}$.
- Set $\mathrm{ub}_{s,t} = f(x^{ub}_{s,t})$ and the gap $\Delta_{s,t} := \mathrm{ub}_{s,t} - \mathrm{lb}_{s,t}$.

A step of the ABL algorithm: determine the stepsize

- By construction, we have $\Delta_{1,0} \geq \Delta_{1,1} \geq \ldots \geq \Delta_{1,N_1-1} \geq \Delta_{2,0} \geq \Delta_{2,1} \geq \ldots \geq \Delta_{2,N_2-1} \geq \ldots \geq 0$.
- To guarantee the optimal convergence, choose $\alpha_t \in (0, 1]$ s.t.

$$\frac{\alpha_t^2}{\Gamma_t} \leq C_1, \quad \Gamma_t \leq \frac{C_2}{t^2} \quad \text{and} \quad \Gamma_t \left[\sum_{\tau=1}^{t} \left(\frac{\alpha_\tau}{\Gamma_\tau}\right)^2\right]^{\frac{1}{2}} \leq \frac{C_3}{\sqrt{t}}, \ \forall t \geq 1$$

  for some $C_1, C_2, C_3 \in \Re_{++}$, where $\Gamma_1 = 1$ and

$$\Gamma_t := [1 - \lambda\, \alpha_t]\, \Gamma_{t-1}, t \geq 2.$$

  - if $\lambda \in (0, 1/2]$ and $\alpha_t = 2/[\lambda(t+3)]$, the conditions hold with $C_1 = \frac{2}{3\lambda^2}$, $C_2 = 6$ and $C_3 = \frac{1}{3\sqrt{3}\,\lambda}$;
  - if α_t are recursively defined by

$$\alpha_1 = \Gamma_1 = 1, \quad \Gamma_t = \alpha_t^2 = [1 - \lambda\alpha_t]\Gamma_{t-1}, \quad \forall\, t \geq 2,$$

    then the conditions hold with

$$C_1 = \frac{4}{\lambda^2}, C_2 = 1 \text{ and } C_3 = \frac{4}{\sqrt{3\lambda}}.$$

- The selection of α_t does not depend on L, M and D_X.

Convergence of the ABL algorithm

Theorem: The total number of steps performed by the ABL algorithm before termination does not exceed $\mathcal{N}(\epsilon) + \mathcal{S}(\epsilon)$ with

$$\mathcal{N}(\epsilon) := \frac{1}{1-\sqrt{\lambda}} \sqrt{\frac{3C_1 C_2 L D_X^2}{2\lambda\,\epsilon}} + \frac{1}{1-\lambda^2} \left(\frac{3C_3 M D_X}{\lambda\,\epsilon}\right)^2,$$

$$\mathcal{S}(\epsilon) := \left\lceil 1 + \left(\frac{3C_2}{\lambda}\right)^{\frac{1}{2}} \right\rceil \left\lceil \log_{\frac{1}{\lambda}} \left(\frac{L D_X^2}{2\epsilon} + \frac{M D_X}{\epsilon}\right) \right\rceil.$$

- When terminates, we have $f(x_{s,t}^{ub}) - f^* \leq \mathrm{ub}_{s,t} - \mathrm{lb}_{s,t} \leq \epsilon$;
- It can be easily seen that $S(\epsilon) = \mathcal{O}(N(\epsilon))$;
- Uniformly optimal for smooth and non-smooth CP without requiring any smoothness information.

Accelerated prox-level method with limited memory

- One apparent problem for the ABL algorithm is that, as the algorithm proceeds, the subproblems
$$\mathrm{lb}_{s,t} = \min_{x \in X} m_{s,t}(x)$$
$$x_{s,t} = \operatorname{argmin}_{x \in X} \left\{ \|x - x_{s,t-1}\|^2 : m_{s,t}(x) \leq l_{s,t} \right\}$$
become more and more difficult to solve. As a result, each step of the ABL algorithm becomes computationally more and more expensive.
- To remedy this issue, we present the accelerated prox-level (APL) method, in which the number of constraints and thus the complexity of its subproblems are under our full control.
- Moreover, non-Euclidean prox-functions can be employed in place of $\|\cdot\|^2$ for the second subproblem, in order to make use of the geometry of the feasible set X to get (nearly) dimension-independent convergence rates.

The APL algorithm with limited memory

- The steps of the APL algorithm are divided into subsequent phases, corresponding to segments in the ABL algorithm.
- Phase s, $s = 1, 2, \ldots$, is associated with a prox-center c_s and a level $l_s \in \Re$ s.t.
  - the values of $f(c_s)$ and $f'(c_s)$ are known when Phase s starts;
  - $l_s = \lambda \tilde{\text{lb}}_s + (1 - \lambda)\tilde{\text{ub}}_s$, where $\lambda \in (0, 1)$ and $\tilde{\text{ub}}_s$ and $\tilde{\text{lb}}_s$, respectively, are the smallest objective value and the largest lower bound on f^* found when Phases s start.

The prox-function

Let $\omega : X \to \Re$ be a given differentiable and strongly convex function with modulus σ (e.g., $\omega(x) = \|x\|^2/2$). At phase s, we define the prox-function

$$\omega_s(x) \equiv \omega(x) - [\omega(c_s) + \nabla\omega(c_s)^T(x - c_s)].$$

- Note that $\omega_s(x)$ is also differentiable and strongly convex with modulus σ, i.e.,
$$\omega_s(y) \geq \omega_s(x) + \langle \nabla\omega_s(x), y - x\rangle + \tfrac{\sigma}{2}\|y - x\|^2.$$
- We have $\nabla\omega_s(c_s) = 0$ and hence that $c_s = \arg\min_{x\in X} \omega_s(x)$.

The APL algorithm: initialization

- Given an initial point $p_0 \in X$, we start the entire process by computing a valid lower bound on f^*,i.e.,
$$\tilde{\text{lb}}_1 = \min_{x \in X} \{f(p_0) + \langle f'(p_0), x - p_0 \rangle\} .$$
- Let $\tilde{p}_0$ be an optimal solution of the previous problem, we compute an upper bound on f^* by setting $\tilde{\text{ub}}_1 = f(\tilde{p}_0)$. [2]
- The prox-center $c_1 \in X$ of the very first phase can be chosen arbitrarily, say p_0 or $\tilde{p}_0$.

[2] Essentially, the upper bound $\tilde{\text{ub}}_1$ can be the function value of an arbitrary feasible point in X, since the initial gap $\tilde{\text{ub}}_1 - \tilde{\text{lb}}_1$ only logarithmically affects the rate of convergence of the APL algorithm.

A step of the APL algorithm

- Similar to the ABL method, each step t, $t = 0, 1, \ldots$, of phase s, $s = 1, 2, \ldots$, updates three interwined search points, namely, $(x_{s,t}, x^{ub}_{s,t}, x^{lb}_{s,t})$.
- When generating $(x_{s,t}, x^{ub}_{s,t}, x^{lb}_{s,t})$, we have already in our disposal $(x_{s,t-1}, x^{ub}_{s,t-1}, x^{lb}_{s,t-1})$, a valid lower bound $\mathrm{lb}_{s,t-1}$ on f^*, and a convex compact set called *localizer,* $X_{s,t-1} \subseteq X$*, such that*
$$\mathcal{L}_s := \{x \in X : f(x) \leq l_s\} \subseteq X_{s,t-1}.$$
  - *In the beginning of phase* s*, we set* $x_{s,0} = c_s$ *and* $\mathrm{lb}_{s,0} = \tilde{\mathrm{lb}}_s$*;*
  - $(x^{ub}_{s,0}, x^{lb}_{s,0}) \in X \times X$ *can be chosen arbitrarily;*
  - $X_{s,0}$ *can be chosen as the* X *intersected with a few "cuts" from the previous phases.*
- *To update* $(x_{s,t-1}, x^{ub}_{s,t-1}, x^{lb}_{s,t-1}, X_{s,t-1})$ *into* $(x_{s,t}, x^{ub}_{s,t}, x^{lb}_{s,t}, X_{s,t})$*, the APL follows the following steps.*

A step of the APL algorithm: update sequences

- Set $x^{lb}_{s,t} = (1-\alpha_t)x^{ub}_{s,t-1} + \alpha_t x_{s,t-1}$, where $\alpha_t \in (0,1]$.
- Compute $h^* := \min_{x \in X_{s,t-1}} h(x^{lb}_{s,t}, x)$, and observe that $\hat{\mathrm{lb}} := \min\{h^*, l_s\}$ is a lower bound on f^*. Set
$$\mathrm{lb}_{s,t} := \max\{\mathrm{lb}_{s,t-1}, \hat{\mathrm{lb}}\}.$$
- Depending on the value of $\mathrm{lb}_{s,t}$, we consider two cases:

  Case I: Significant progress on the lower bound. If
  $$\mathrm{lb}_{s,t} \geq l_s - \theta(l_s - \tilde{\mathrm{lb}}_s),$$
  where $\theta \in (0,1)$ is a parameter, we pass to phase $s+1$, by setting
  $$\tilde{\mathrm{lb}}_{s+1} = \mathrm{lb}_{s,t},$$
  $$\tilde{\mathrm{ub}}_{s+1} = \min\left\{\tilde{\mathrm{ub}}_s, \min_{0 \leq \tau \leq t-1} f(x^{ub}_{s,\tau})\right\}$$
  The prox-center c_{s+1} can be chosen in X arbitrarily, e.g.,
  $$c_{s+1} = x^{ub}_{s,t-1};$$

A step of the APL algorithm: update sequences

- Depending on the value of $\mathrm{lb}_{s,t}$, we consider two cases:

  Case II: No significant progress on the lower bound.
  First compute:

  $$x_{s,t} \equiv \mathrm{argmin}_x \left\{ \omega_s(x) : x \in X_{s,t-1}, h(x_t^{lb}, x) \leq l_s \right\}.$$

  Note that the above subproblem must be feasible, o.w., the current branch cannot happen. Then we set $x_{s,t}^{ub} \in X$ s.t.

  $$f(x_{s,t}^{ub}) \leq f(\alpha_t x_{s,t} + (1 - \alpha_t) x_{s,t-1}^{ub}).$$

  The simplest option is $x_{s,t}^{ub} = \alpha_t x_{s,t} + (1 - \alpha_t) x_{s,t-1}^{ub}$.
  Then, check if the progress on the the objective value is significant. In particular, we consider the following two subcases.

A step of the APL algorithm: update sequences

- Two subcases after computing $x_{s,t}^{ub} \in X$:
  - **Case II.a):** Significant progress on upper bound. If $f(x_{s,t}^{ub}) - l_s \leq \theta(\tilde{\text{ub}}_s - l_s)$, pass to phase $s+1$ by setting
    $$\tilde{\text{lb}}_{s+1} = \text{lb}_{s,t} \quad \tilde{\text{ub}}_{s+1} = \min\left\{\tilde{\text{ub}}_s, \min_{0\leq\tau\leq t} f(x_{s,\tau}^{ub})\right\}.$$
  - **Case II.b):** No significant progress on upper bound. Continue phase s and update the localizer $X_{s,t}$ as an arbitrary convex compact set such that
    $$\underline{X}_{s,t} \subseteq X_{s,t} \subseteq \overline{X}_{s,t},$$
    where $\underline{X}_{s,t} \equiv \left\{x \in X_{s,t-1} : h(x_{s,t}^{lb}, x) \leq l_s\right\}$ and
    $$\overline{X}_{s,t} \equiv \{x \in X : \langle \nabla\omega_s(x_t), x - x_{s,t}\rangle \geq 0\}.$$

    Observations:
    1. $\emptyset \neq \underline{X}_{s,t} \subseteq \overline{X}_{s,t}$;
    2. $\mathcal{L}_s \subseteq \underline{X}_{s,t} \subseteq X_{s,t}$ if $\mathcal{L}_s \subseteq X_{s,t-1}$.

A step the APL algorithm: determine the stepsize

- To guarantee the optimal convergence, choose $\alpha_t \in (0, 1]$ s.t.

  $\alpha_1 = 1, \frac{\alpha_t^2}{\tilde{\Gamma}_t} \leq \tilde{C}_1, \tilde{\Gamma}_t \leq \frac{\tilde{C}_2}{t^2}$ and $\tilde{\Gamma}_t \left[\sum_{\tau=1}^{t} \left(\frac{\alpha_\tau}{\tilde{\Gamma}_\tau}\right)^2\right]^{\frac{1}{2}} \leq \frac{\tilde{C}_3}{\sqrt{t}}, t \geq 1$

  for some $\tilde{C}_1, \tilde{C}_2, \tilde{C}_3 \in \Re_{++}$, where $\tilde{\Gamma}_1 = 1$ and $\tilde{\Gamma}_t = \tilde{\Gamma}_{t-1}(1 - \alpha_t)$ for $t \geq 2$.

  - If $\alpha_t = 2/(t+1)$, then the above conditions hold with
    $$\tilde{C}_1 = 2, \tilde{C}_2 = 2 \text{ and } \tilde{C}_3 = 2/\sqrt{3}.$$
  - If α_t are recursively defined by
    $$\alpha_1 = \tilde{\Gamma}_1 = 1, \quad \alpha_t^2 = (1 - \alpha_t)\tilde{\Gamma}_{t-1} = \tilde{\Gamma}_t, \ \forall\, t \geq 2,$$
    then the above conditions hold with
    $$\tilde{C}_1 = 1, \tilde{C}_2 = 4 \text{ and } \tilde{C}_3 = 4/\sqrt{3}.$$

- The selection of α_t does not depend on L, M and D_X.

Convergence of the APL algorithm

Theorem: Denote $q \equiv q(\theta, \lambda) := 1 - (1-\theta)\min\{\lambda, 1-\lambda\}$. The total number of steps performed by the APL algorithm before finding a point $\bar{x} \in X$ s.t. $f(\bar{x}) - f^* \leq \epsilon$, does not exceed $\tilde{\mathcal{N}}(\epsilon) + \tilde{\mathcal{S}}(\epsilon)$, where

$$\tilde{\mathcal{N}}(\epsilon) = \frac{1}{1-\sqrt{q}} \sqrt{\frac{2\tilde{C}_1 \tilde{C}_2 L \mathcal{D}^2_{\omega,X}}{\sigma\theta\lambda\,\epsilon}} + \frac{1}{1-q^2}\left(\frac{2\sqrt{2}\tilde{C}_3 M \mathcal{D}_{\omega,X}}{\sqrt{\sigma}\theta\lambda\,\epsilon}\right)^2,$$

$$\tilde{\mathcal{S}}(\epsilon) := 1 + \max\left\{0, \log\left(\frac{L\mathcal{D}^2_{\omega,X}}{\sigma\epsilon} + \sqrt{\frac{2}{\sigma}}\frac{M\mathcal{D}_{\omega,X}}{\epsilon}\right)\right\},$$

$$\mathcal{D}^2_{\omega,X} := \max_{x,y\in X}\{\omega(y) - \omega(x) - \langle\nabla\omega(x), y - x\rangle\}.$$

- It can be easily seen that $\tilde{\mathcal{S}}(\epsilon) = \mathcal{O}(\tilde{\mathcal{N}}(\epsilon))$;
- Uniformly optimal for smooth and non-smooth CP without requiring any smoothness information.

Minimizing strongly convex functions

Assume that for some $\mu > 0$,
$f(y) - f(x) - \langle f'(x), y - x\rangle \geq \frac{\mu}{2}\|y - x\|^2, \ \ \forall\, x, y \in X.$

Theorem: If either one of the following two conditions hold:

i) $f(c_s) = \mathrm{ub}_s, \forall s \geq 1$ and $x^* \in X_{s,t}$,

ii) $\underline{X}_{s,t} \subseteq X_{s,t} \subseteq \overline{X}_{s,t} \cap \left\{x \in X : \|x - c_s\|^2 \leq \frac{2\tilde{\Delta}_s}{\mu}\right\}$.

Then, the total number of steps performed by the APL algorithm before finding a point $\bar{x} \in X$ s.t. $f(\bar{x}) - f^* \leq \epsilon$, does not exceed

$$\begin{cases} \mathcal{O}(1)\sqrt{\frac{L}{\mu}}\left[1 + \max\left(0, \log\frac{LD^2_{\omega,X}}{\sigma\epsilon}\right)\right] & \text{if } M = 0; \\ \mathcal{O}(1)\frac{M^2}{\mu\epsilon} & \text{if} L = 0. \end{cases}$$

Minimizing strongly convex functions

How to ensure either one of the following conditions?

i) $f(c_s) = \mathrm{ub}_s, \forall s \geq 1$ and $x^* \in X_{s,t}$,

ii) $\underline{X}_{s,t} \subseteq X_{s,t} \subseteq \overline{X}_{s,t} \cap \left\{ x \in X : \|x - c_s\|^2 \leq \frac{2\tilde{\Delta}_s}{\mu} \right\}$.

- To ensure Condition i):
  - choose c_s as the best solution found to ensure the first relation;
  - the second relation is automatically satisfied in under certain specific assumptions, for example, when the optimal value f^* is known and the initial lower bound lb_1 is set to f^*;
  - do not need to do any modification to the definition of $X_{s,t}$.
- To ensure Condition ii):
  - incorporate an additional constraint, namely, $\|x - c_s\|^2 \leq 2\tilde{\Delta}_s^2/\mu$, into the definition of $X_{s,t}$. The basic idea is to shrink the feasible set whenever a new phase starts.

Defining and solving the ABL subproblems

The subproblems for **the ABL algorithm:**

$$\mathrm{lb}_{s,t} = \min_{x \in X} m_{s,t}(x)$$
$$x_{s,t} = \operatorname{argmin}_{x \in X} \left\{ \|x - x_{s,t-1}\|^2 : m_{s,t}(x) \leq l_{s,t} \right\}.$$

- In practice, only choosing the most n recently generated linear forms in the definition of $m_{s,t}(x)$ does not slow down the convergence.
- For simple X, use interior point method to solve these subproblems;

Defining the APL subproblems

The subproblems for **the APL algorithm:**

$$h^* := \min_{x \in X_{s,t-1}} h(x^{lb}_{s,t}, x);$$
$$x_{s,t} \equiv \operatorname{argmin}_x \left\{ \omega_s(x) : x \in X_{s,t-1}, h(x^{lb}_t, x) \leq l_s \right\}.$$

- Let $\{x^{lb}_{i,j}\}$ be the sequence of search points generated for computing the lower bound. We can set
$$X_{s,t} = \{x \in X : \nabla \omega_s(x_t), x - x_{s,t} \geq 0\} \bigcap \mathcal{M}_{s,t},$$
where $\mathcal{M}_{s,t}$ denotes the intersection of at most $B = 5$ or 10 half spaces of the form $\left\{ x : h(x^{lb}_{i,j}, x) \leq l_s, \right\}$.
- In our implementation, we define $\mathcal{M}_{s,t}$ as the intersection of the most recent B half subspaces obtained in this manner.
- The subproblems have at most $B + 1$ constraints in additions to $x \in X$.

Solving the APL subproblems

Several ways for solving the APL subproblems

$$h^* := \min_{x \in X_{s,t-1}} h(x^{lb}_{s,t}, x);$$
$$x_{s,t} \equiv \operatorname{argmin}_x \left\{ \omega_s(x) : x \in X_{s,t-1}, h(x^{lb}_t, x) \leq l_s \right\}.$$

- Using IPM solver.
- The Lagrangian dual problem only has a very small number of variables. Moreover,
  - the Lagrangian dual of the first subproblem is non-smooth;
  - the Lagrangian dual of the second subproblem is smooth;
  - the first-order information of the Lagrangian dual can be easily computed if X is simple enough.
- We can solve the Lagrangian dual of these subproblems by any efficient algorithms for lower-dimensional CP, such as the ABL algorithm or the Ellipsoid algorithm.

Numerical results

- The quadratic problem: $\min_{\|x\| \le R} \|Ax - b\|^2$, where $A \in \Re^{m \times n}$.
  - A and b are generated randomly, so that the optimal value is 0.
- Compare the following algorithms:
  - APL algorithm with subproblems solved by Mosek. The initial lower bound can be set to $-\infty$ or 0.
  - APL algorithm with subproblems solved by the ABL algorithm. The initial lower bound can be set to $-\infty$ or 0.
  - NERML (Ben-tal and Nemirovski 00) algorithm with subproblem solved by Mosek. The initial lower bound can be set to $-\infty$ or 0.
  - Nesterov's optimal method (Nesterov 05).
- Implemented in MATLAB2007, Windows Vista.

Numerical results

Table: INST1: $n = 4000$, $m = 2000$, $L = 2.00e6$ and $err_0 = 3.85e4$

Alg.	Ini. LB	Iter.	Acc.	Time(sec.)
APL(Mosek)	0	81	$9.23e-7$	70.35
	$-\infty$	270	$7.10e-7$	222.16
APL(ABL)	0	115	$8.75e-7$	91.71
	$-\infty$	248	$7.20e-7$	262.87
NERML(Mosek)	0	157	$9.97e-7$	132.92
	$-\infty$	300	$6.30e-4$	323.73
NEST	-	$10,000$	$2.28e-5$	257.41

Note:

- terminate NSET in $10,000$ iterations or when accuracy $< 1.0e-6$.
- terminate other algorithms in 300 iterations or when accuracy $< 1.0e-6$.

Numerical results

Table: INST2: $n = 6000$, $m = 3000$, $L = 4.45e6$ and $err_0 = 3.85e4$

Alg.	Ini. LB	Iter.	Acc.	Time(sec.)
APL(Mosek)	0	79	$8.04e-7$	138.32
	$-\infty$	291	$6.72e-7$	471.73
APL(ABL)	0	149	$9.46e-7$	138.44
	$-\infty$	300	$1.31e-6$	348.58
NERML(Mosek)	0	148	$8.41e-7$	249.10
	$-\infty$	300	$1.64e-4$	504.21
NEST	-	$10,000$	$2.28e-5$	1016.80

Numerical results

Table: INST3: $n = 8000$, $m = 4000$, $L = 8.0e6$ and $err_0 = 3.2e6$

Alg.	Ini. LB	Iter.	Acc.	Time(sec.)
APL(Mosek)	0	77	$6.39e-7$	224.21
	$-\infty$	254	$7.50e-7$	689.82
APL(ABL)	0	144	$9.44e-7$	158.19
	$-\infty$	279	$7.74e-7$	356.75
NERML(Mosek)	0	152	$9.33e-7$	424.69
	$-\infty$	300	$1.17e-3$	832.38
NEST	-	$10,000$	$5.34e-5$	1803.13

Some work in progress

- Adapt the smooth APL method for solving non-smooth problems of the form
$$\min_{x \in X} \max_{y \in Y} \langle Ax, y \rangle;$$
- "Smoothing" the objective function at each stage with a decreasing smooth parameter θ_s, i.e.,
$$\min_{x \in X} \max_{y \in Y} \langle Ax, y \rangle + \theta_s \tilde{\omega}(y);$$
  - the size of θ_s is proportional to $\tilde{\Delta}_s$, without requiring to know the final accuracy ϵ;
  - do not need to know the operator norm $\|A\|_{X,Y}$
- Quite promising computational results are obtained for solving matrix games and eigenvalue problems.

Summary

- Previous studies on Bundle-type methods focused on non-smooth problem:
  - Bundle-level methods and their certain variants, e.g., NERML, are known to be optimal for general non-smooth CP;
- Propose new bundle-type methods, including the ABL and APL algorithms,
  - uniformly optimal for both non-smooth and smooth CP problem;
  - do not require any smoothness information, such as whether a problem is smooth or not, how big the Lipschitz constant is, etc;
  - can be easily extended for solving strongly convex problems, sometimes without any modifications.

Summary

- Our preliminary numerical results demonstrate that
  - direct applications of the existing cutting plane methods, e.g., NERML, to smooth CP, might not work well, especially when there is no given initial lower bound;
  - the APL algorithm with limited memory can significantly outperform the existing optimal method for smooth CP especially when the Lipschitz constant is big and/or the desired accuracy is high.