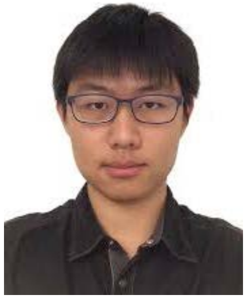
Primal-dual Alg for Time-varying Nonconvex Opt

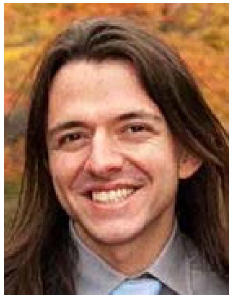


Yujie Tang

Emiliano Dall'Anese (UC Boulder)

Steven Low

Caltech



Andrey Bernstein (NREL)

December 2018

Outline

Problem formulation

- Motivation: realtime OPF
- Time-varying optimization

Tracking algorithm

- Regularized primal-dual alg
- Tracking performance
- Simulations

To appear:
IEEE CDC 2018

Motivations

OPF is solved routinely for

- network control & optimization decisions
- market operations & pricing
- at timescales of mins, hours, days, ...
- Huge literature since 1962

Two problems we focus on

- Nonconvexity
- Time-varying

$$\boxed{\min\ c(x) \quad \text{s. t.} \quad F(x) = 0,\ \ x \le \bar{x}}$$

Optimal power flow

$$
\begin{array}{lll l}
\text{min} & \mathrm{tr}\left(CVV^H\right) & & \text{gen cost, power loss} \\
\text{over} & (V, s, l) & & \\
\text{subject to} & s_j = \mathrm{tr}\left(Y_j^H VV^H\right) & & \text{power flow equation} \\
 & l_{jk} = \mathrm{tr}\left(B_{jk}^H VV^H\right) & & \text{line flow} \\
 & \underline{s}_j \le s_j \le \overline{s}_j & & \text{injection limits} \\
 & \underline{l}_{jk} \le l_{jk} \le \overline{l}_{jk} & & \text{line limits} \\
 & \underline{V}_j \le |V_j| \le \overline{V}_j & & \text{voltage limits}
\end{array}
$$

- Y_j^H describes network topology and impedances
- s_j is net power injection (generation) at node j

Optimal power flow

$$
\begin{array}{ll}
\min & \mathrm{tr}\left(CVV^H\right) \\
\text{over} & (V, s, l) \\
\text{subject to} & s_j = \mathrm{tr}\left(Y_j^H VV^H\right) \\
& l_{jk} = \mathrm{tr}\left(B_{jk}^H VV^H\right) \\
& \underline{s}_j \le s_j \le \overline{s}_j \\
& \underline{l}_{jk} \le l_{jk} \le \overline{l}_{jk} \\
& \underline{V}_j \le |V_j| \le \overline{V}_j
\end{array}
$$

gen cost, power loss

power flow equation

line flow

injection limits

line limits

voltage limits

nonconvex feasible set (nonconvex QCQP)

- Y_j^H not Hermitian (nor positive semidefinite)
- C is positive semidefinite (and Hermitian)

Realtime OPF

Track optimal solution of time-varying OPF

- Uncertainty will continue to increase
- Real-time measurements increasingly become available on seconds timescale
- Must, and can, close the loop in the future

Time-varying optimization

$$\min_{x \in \mathcal{X}_t} \quad c_t(x)$$

$$\text{s.t.} \quad f_t(x) \leq 0$$

Assumptions

- $c_t : \mathbb{R}^n \to \mathbb{R}$ and $f_t : \mathbb{R}^n \to \mathbb{R}^m$ twice cont. differentiable
  possibly nonconvex
- $\mathcal{X}_t \subset \mathbb{R}^n$ compact convex

Time-varying optimization

$$\min_{x \in \mathcal{X}_t} \quad c_t(x)$$

$$\text{s.t.} \quad f_t(x) \leq 0$$

KKT condition:

$$(x_t^*, \lambda_t^*) \in \mathcal{X}_t \times \mathbb{R}_+^m$$ primal-dual feasibe

$$\nabla c_t(x_t^*) + J_{f_t}(x_t^*)^T \lambda_t^* \in -N_{\mathcal{X}_t}(x_t^*)$$ stationarity

$$f_t(x_t^*) \leq 0, \quad {\lambda_t^*}^T f_t(x_t^*) = 0$$ complementary slackness

Goal: algorithm to track KKT trajectory (e.g. local opt)

Newton algorithm

Quasi-Newton algorithm [Tang, Dj & L TSG 2017]:

$$x(t+1) = \left[x(t) - \eta\left(H(t)\right)^{-1} \frac{\partial f_t}{\partial x}(x(t)) \right]_{X_t} \quad \text{active control}$$

$$y(t) = y(x(t)) \quad \text{law of physics}$$

Hessian calculation computationally expensive

- Motivates first-order algorithm
- Add regularization for better tracking

Time-varying optimization

$$\begin{aligned} \min_{x \in \mathcal{X}_t} \quad & c_t(x) \\ \text{s.t.} \quad & f_t(x) \leq 0 \end{aligned}$$

Difficulties:

- c_t and f_t can be nonconvex
- Jacobian $J_{f_t}(x)$ can be difficult to compute

Outline

Problem formulation

- Motivation: realtime OPF
- Time-varying optimization

Tracking algorithm

- Regularized primal-dual alg
- Tracking performance
- Simulations

To appear:
IEEE CDC 2018

Tracking algorithm

$$\min_{x \in \mathcal{X}_t} \quad c_t(x)$$
$$\text{s.t.} \quad f_t(x) \le 0$$

Dealing with nonconvexity

- Approximate c_t and f_t by linearizations

$$\min_{x \in \mathcal{X}_t} \quad \nabla c_t(\widehat{x}_{t-1})^T (x - \widehat{x}_{t-1})$$
$$\text{s.t.} \quad f_t(\widehat{x}_{t-1}) + J_{f_t}(\widehat{x}_{t-1})(x - \widehat{x}_{t-1}) \le 0$$

Tracking algorithm

$$\min_{x \in \mathcal{X}_t} \quad c_t(x)$$
$$\text{s.t.} \quad f_t(x) \leq 0$$

Dealing with nonconvexity

- Approximate c_t and f_t by linearizations
- with quadratic regularizer

$$\min_{x \in \mathcal{X}_t} \quad \nabla c_t(\widehat{x}_{t-1})^T (x - \widehat{x}_{t-1}) + \frac{\nu}{2} \|x - \widehat{x}_{t-1}\|^2$$
$$\text{s.t.} \quad f_t(\widehat{x}_{t-1}) + J_{f_t}(\widehat{x}_{t-1})(x - \widehat{x}_{t-1}) \leq 0$$

- Approximate problem is a simple quadratic program (QP)

Tracking algorithm

$$\min_{x \in \mathcal{X}_t} \quad c_t(x)$$
$$\text{s.t.} \quad f_t(x) \leq 0$$

Dealing with nonconvexity

- Apply primal-dual algorithm with quadratic dual regularizer
- Lagragian $L_t(x, \lambda)$ quadratic in both (x, λ)

$$\max_{\lambda \in \mathbb{R}_+^m} \min_{x \in \mathcal{X}_t} \quad \nabla c_t(\widehat{x}_{t-1})^T (x - \widehat{x}_{t-1}) + \frac{\nu}{2} \|x - \widehat{x}_{t-1}\|^2$$
$$+ \lambda^T \left(f_t(\widehat{x}_{t-1}) + J_{f_t}(\widehat{x}_{t-1})(x - \widehat{x}_{t-1}) \right)$$
$$- \frac{\epsilon}{2} \|\lambda\|^2$$

Tracking algorithm

$$\max_{\lambda \in \mathbb{R}^m_+} \min_{x \in \mathcal{X}_t} \quad \nabla c_t(\widehat{x}_{t-1})^T (x - \widehat{x}_{t-1}) + \frac{\nu}{2}\|x - \widehat{x}_{t-1}\|^2$$
$$+ \lambda^T \left(f_t(\widehat{x}_{t-1}) + J_{f_t}(\widehat{x}_{t-1})(x - \widehat{x}_{t-1})\right)$$
$$- \frac{\epsilon}{2}\|\lambda\|^2$$

Primal-dual algorithm

$$\hat{x}_t = \left[\hat{x}_{t-1} - \tau\nu^{-1}\frac{\partial L}{\partial x}(\hat{x}_{t-1}, \hat{\lambda}_{t-1})\right]_{X_t}$$

$$\hat{\lambda}_t = \left[\hat{\lambda}_{t-1} - \tau\epsilon^{-1}\frac{\partial L}{\partial \lambda}(\hat{x}_{t-1}, \hat{\lambda}_{t-1})\right]_+$$

Difficulty: Jacobian $J_{f_t}(x)$ is difficult to compute

Tracking algorithm

$$\max_{\lambda \in \mathbb{R}^m_+} \min_{x \in \mathcal{X}_t} \quad \nabla c_t(\widehat{x}_{t-1})^T (x - \widehat{x}_{t-1}) + \frac{\nu}{2}\|x - \widehat{x}_{t-1}\|^2 + \lambda^T \left(f_t(\widehat{x}_{t-1}) + J_{f_t}(\widehat{x}_{t-1})(x - \widehat{x}_{t-1})\right) - \frac{\epsilon}{2}\|\lambda\|^2$$

Primal-dual algorithm

$$\widehat{x}_t = \mathcal{P}_{\mathcal{X}_t}\left(\widehat{x}_{t-1} - \tau\nu^{-1}\left(\nabla c_t(\widehat{x}_{t-1}) + J_{f_t}(\widehat{x}_{t-1})^T \widehat{\lambda}_{t-1}\right)\right)$$

$$\widehat{\lambda}_t = \mathcal{P}_{\mathbb{R}^m_+}\left((1-\tau)\widehat{\lambda}_{t-1} + \tau\epsilon^{-1} f_t(\widehat{x}_{t-1})\right)$$

Difficulty: Jacobian $J_{f_t}(x)$ is difficult to compute

Tracking algorithm

$$\max_{\lambda \in \mathbb{R}_+^m} \min_{x \in \mathcal{X}_t} \quad \nabla c_t(\widehat{x}_{t-1})^T (x - \widehat{x}_{t-1}) + \frac{\nu}{2}\|x - \widehat{x}_{t-1}\|^2$$
$$+ \lambda^T \left(f_t(\widehat{x}_{t-1}) + J_{f_t}(\widehat{x}_{t-1})(x - \widehat{x}_{t-1})\right)$$
$$- \frac{\epsilon}{2}\|\lambda\|^2$$

Computing Jacobian

- Use actual measurements: feedback-based algorithm
- Assumptions
  - Measurement maps input $x_t \in R^n$ to output $y_t(x) \in R^m$
  - Constraints $f_t(x) \leq 0$ becomes $H_t y_t(x) \leq 0$
- Jacobian becomes

$$J_{f_t}(x) = J_{h_t}(y_t(x)) J_{y_t}(x) = H_t J_{y_t}(x)$$

Tracking algorithm

$$\max_{\lambda \in \mathbb{R}_+^m} \min_{x \in \mathcal{X}_t} \quad \nabla c_t(\widehat{x}_{t-1})^T (x - \widehat{x}_{t-1}) + \frac{\nu}{2} \|x - \widehat{x}_{t-1}\|^2$$
$$+ \lambda^T \left(f_t(\widehat{x}_{t-1}) + J_{f_t}(\widehat{x}_{t-1})(x - \widehat{x}_{t-1}) \right)$$
$$- \frac{\epsilon}{2} \|\lambda\|^2$$

Primal-dual algorithm with output-feedback:

$$\widetilde{x}_t = \mathcal{P}_{\mathcal{X}_t}\Big(\widetilde{x}_{t-1} - \tau\nu^{-1}\big(\nabla c_t(\widetilde{x}_{t-1}) + [H_t J_{y_t}(\widetilde{x}_{t-1})]^T \widetilde{\lambda}_{t-1}\big)\Big)$$

$$\widetilde{\lambda}_t = \mathcal{P}_{\mathbb{R}_+^m}\Big((1-\tau)\widetilde{\lambda}_{t-1} + \tau\epsilon^{-1} h_t(\check{y}_t)\Big)$$

Tracking performance

Error $:= \quad \left\| \widetilde{z}_t - z_t^* \right\|$

$z_t^* := (x_t^*, \lambda_t^*)$: KKT trajectory (e.g. local opt)

Tracking algorithm trajectory

Tracking performance

Error $:=$ $\|\tilde{z}_t - z_t^*\|$

Theorem

$$\|\tilde{z}_t - z_t^*\| \le \frac{\rho_{\nu,\epsilon}(\delta,\tau)\sigma + \sqrt{2\nu^{-1}\epsilon}\tau(M_\lambda + \epsilon^{-1}L_h e_y)}{1-\rho_{\nu,\epsilon}(\delta,\tau)} \quad \text{for all } t$$

Error is small if Lagrangian is close to convex and problem changes slowly

- "local" convexity of $L(\cdot,\lambda)$
- rate of change: $\sigma = \sup_{t\in\mathcal{T}\setminus\{0\}} \|z_t^* - z_{t-1}^*\|$

Tracking performance

Error $:= \quad \|\widetilde{z}_t - z_t^*\|$

Theorem

$$\|\widetilde{z}_t - z_t^*\| \le \frac{\rho_{\nu,\epsilon}(\delta,\tau)\sigma + \sqrt{2\nu^{-1}\epsilon}\tau(M_\lambda + \epsilon^{-1}L_h e_y)}{1-\rho_{\nu,\epsilon}(\delta,\tau)} \quad \text{for all } t$$

Error due to regularization

$$\begin{aligned}\max_{\lambda\in\mathbb{R}^m_+}\min_{x\in\mathcal{X}_t} \quad & \nabla c_t(\widehat{x}_{t-1})^T(x-\widehat{x}_{t-1}) + \frac{\nu}{2}\|x-\widehat{x}_{t-1}\|^2 \\ & + \lambda^T\left(f_t(\widehat{x}_{t-1}) + J_{f_t}(\widehat{x}_{t-1})(x-\widehat{x}_{t-1})\right) \\ & - \frac{\epsilon}{2}\|\lambda\|^2\end{aligned}$$

Tracking performance

Error := $\|\widetilde{z}_t - z_t^*\|$

Theorem

$$\|\widetilde{z}_t - z_t^*\| \leq \frac{\rho_{\nu,\epsilon}(\delta,\tau)\sigma + \sqrt{2\nu^{-1}\epsilon}\tau(M_\lambda + \epsilon^{-1}L_h e_y)}{1-\rho_{\nu,\epsilon}(\delta,\tau)} \quad \text{for all } t$$

Error due to measurement

$$L_h = \sup_{t \in \mathcal{T}} \|H_t\|$$

Key message

Large network of DERs

- Real-time optimization at scale
- Computational challenge: power flow solution

Online optimization [feedback control]

- Network computes power flow solutions in real time at scale for free
- Exploit it for our optimization/control
- Naturally adapts to evolving network conditions

Examples

- Slow timescale: OPF
- Fast timescale: frequency control