Hopfield Methods: Application to Optimization of Distributed Energy Resources

Scott Moura

Assistant Professor | eCAL Director
University of California, Berkeley

2019 Grid Science Winter School & Conference
Santa Fe, New Mexico USA

eCAL

energy, controls, & applications lab


Dr. Hector PEREZ


Dr. Hongcai ZHANG


Dr. Milad MEMARZADEH


Laurel DUNN


Zach GIMA


Saehong PARK


Dong ZHANG


Victoria CHENG


Sangjae BAE


Bertrand TRAVACCA


Mathilde BADOUAL


Zhe ZHOU


Tianyu HU


Tianyu YANG


Ramon CRESPO


Armando DOMINGOS


Soomin WOO


Yiqi ZHAO


Dylan KATO


Teng ZENG


Patrick KEYANTUO

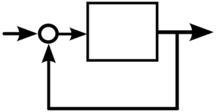

Aaron KANDEL


Pierre-François MASSIANI

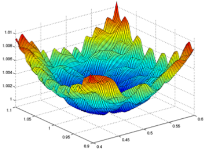

Emily YOU

eCAL
energy, controls, & applications lab

FUNDAMENTAL RESEARCH



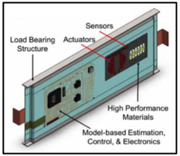
Dynamic Systems & Control



Optimization

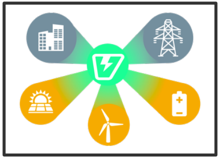
Data Science

APPLICATIONS



Battery Management Systems (#BATT)

Automated, Connected, & Electric Vehicles (#ACES)



Distributed Energy Resources (#DER)

Laurel
DUNN

Probabilistic Model for Assessing Distribution Grid Performance under Hazard Scenarios

Victoria
CHENG

Virtual Inertia Frequency Regulation for Renewable Integration

Armando
DOMINGOS

Multi-Armed Bandits DER Control

Mathilde
BADOUAL

Optimal Bidding Strategy on the Energy Market - A Reinforcement Learning Approach

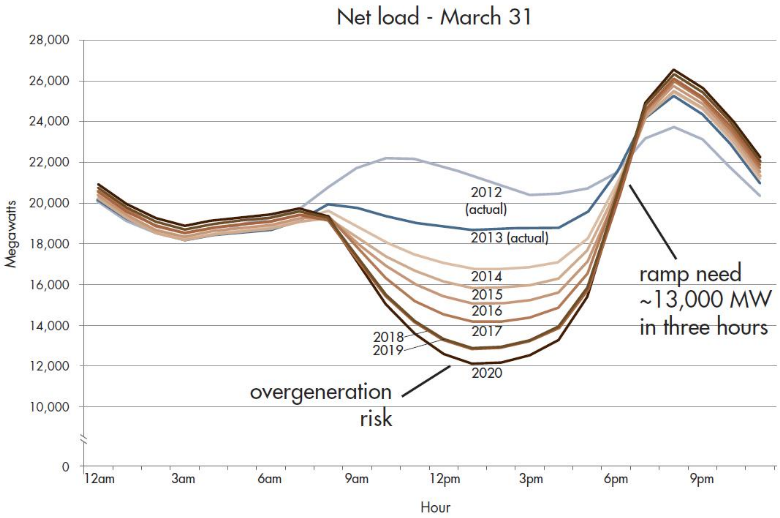
Teng
ZENG

Solving Overstay in PEV Charging Station Planning via Chance Constrained Optimization

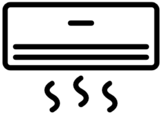
Bertrand
TRAVACCA

THIS TALK

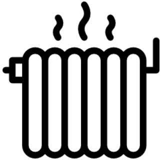
The duck curve shows steep ramping needs and overgeneration risk



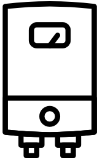
Aggregate Flexible Loads into Virtual Power Plant



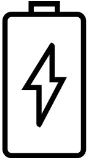
Air Conditioning



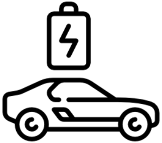
Heater



Water Heater



Energy Storage System



Plug-in Electric Vehicles

- Chen, Hashmi, Mathias, Busic, Meyn (2018)

★ PEV Charge Schedule Optimization is a MIP!

eMotorWerks' Juicebox

Control ∈ *on* (40 *A*)
or *off* (0 *A*)

UC Berkeley Smart EV Charger

Richmond Field Station

Control $\in \{0\ A\} \cup [12\ A, 30\ A]$

Only ≈ 10 % of papers on large-scale optimization of PEVs model DISCRETE charging rates!

★ Problem Statement

Consider a mixed integer nonlinear program (MINLP):

$$\begin{aligned}
\text{minimize} \quad & f(x) && (1)\\
\text{subject to:} \quad & g_j(x) \leq 0, \quad j = 1, \cdots, m && (2)\\
& x_i \in \{0, 1\}, \quad i = 1, \cdots, p < n && (3)\\
& 0 \leq x_i \leq 1, \quad i = p+1, \cdots, n && (4)
\end{aligned}$$

$x \in \mathbb{R}^n$ is the optimization variable
the first $p < n$ variables must be binary
$f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$ is quadratic and L_f – smooth
$g_j(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$ are quadratic and L_j – smooth

★ Problem Statement

Consider a mixed integer nonlinear program (MINLP):

$$\begin{aligned} \text{minimize} \quad & f(x) & (1)\\ \text{subject to:} \quad & g_j(x) \leq 0, \quad j = 1, \cdots, m & (2)\\ & x_i \in \{0,1\}, \quad i = 1, \cdots, p < n & (3)\\ & 0 \leq x_i \leq 1, \quad i = p+1, \cdots, n & (4) \end{aligned}$$

$x \in \mathbb{R}^n$ is the optimization variable
the first $p < n$ variables must be binary
$f(\cdot) : \mathbb{R}^n \to \mathbb{R}$ is quadratic and L_f – smooth
$g_j(\cdot) : \mathbb{R}^n \to \mathbb{R}$ are quadratic and L_j – smooth

Challenge

Solve **LARGE-SCALE** MINLPs, e.g. $n = 10^3, 10^4, 10^5, \cdots$

CLAY MATHEMATICS INSTITUTE

P vs NP – Millenium Prize Problem

Existing Convex Relaxation Methods

1. Binary relaxation
2. Lagrangian relaxation
3. Semi-definite relaxation
4. McCormick relaxations [McCormick ’76][Nagarajan ’16]
5. SoA Branch-and-Bound (linear relaxation) [Belotti ’08]
6. SoA Branch-and-Cut [Achterberg ’08]
7. Quadratic Convex relaxations [Hijazi ’17]
8. Polyhedral relaxations for MIMFs [Nagarajan ’18]

Existing Convex Relaxation Methods

1. Binary relaxation
2. Lagrangian relaxation
3. Semi-definite relaxation
4. McCormick relaxations [McCormick '76][Nagarajan '16]
5. SoA Branch-and-Bound (linear relaxation) [Belotti '08]
6. SoA Branch-and-Cut [Achterberg '08]
7. Quadratic Convex relaxations [Hijazi '17]
8. Polyhedral relaxations for MIMFs [Nagarajan '18]

Stochastic approach to recover integer constraint:

Let x^r be sol'n to binary relaxation. Feasible x can be drawn randomly from $\{0, 1\}$ following Bernoulli distribution $\mathcal{B}(x^r)$.

This can be sub-optimal.

Example

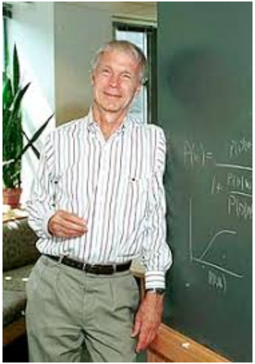
$$\text{minimize}_{x\in\{0,1\}} \left(x - \frac{1}{4}\right)^2 = \frac{1}{16} \quad (x^\star = 0 \text{ is opt. sol'n})$$

If we apply binary relaxation, we get $x^r = \frac{1}{4}$ and $\mathbb{E}_{x\sim\mathcal{B}(x^r)} \left(x - \frac{1}{4}\right)^2 = \frac{3}{16} > \frac{1}{16}$!

Outline

A short history of Hopfield Networks

- (1982) J. J. Hopfield used neural nets to model collaborative computations
- (1985) J. J. Hopfield showed that neural nets can be used to solve optimization problems
- (1990's – 2000's) Hopfield methods became very popular for solving MIQPs in power systems optimization
- In literature, power system researchers admit they didn't fully understand *why* Hopfield methods work well.



The Hopfield Method

Consider MINLP

$$
\begin{aligned}
\text{minimize} \quad & f(x) && (5)\\
\text{subject to:} \quad & x_i \in \{0, 1\}, \quad i = 1, \cdots, p < n && (6)\\
& 0 \leq x_i \leq 1, \quad i = p + 1, \cdots, n && (7)
\end{aligned}
$$

The Hopfield Method

Consider MINLP

$$\begin{aligned} \text{minimize} \quad & f(x) & (5) \\ \text{subject to:} \quad & x_i \in \{0,1\}, \quad i = 1, \cdots, p < n & (6) \\ & 0 \le x_i \le 1, \quad i = p+1, \cdots, n & (7) \end{aligned}$$

Hopfield method follows dynamics:

$$\frac{d}{dt} x_H(t) = -\nabla f(x(t)); \qquad x_H(0) = x(0) \in (0,1)^n \tag{8}$$

$$x(t) = \sigma(x_H(t)) \tag{9}$$

where $\sigma(\cdot) : \mathbb{R}^n \to [0,1]^n$ is an "activiation function" defined element-wise as:

$$\sigma(x) : x \mapsto [\sigma_1(x_1), \cdots, \sigma_n(x_n)]$$

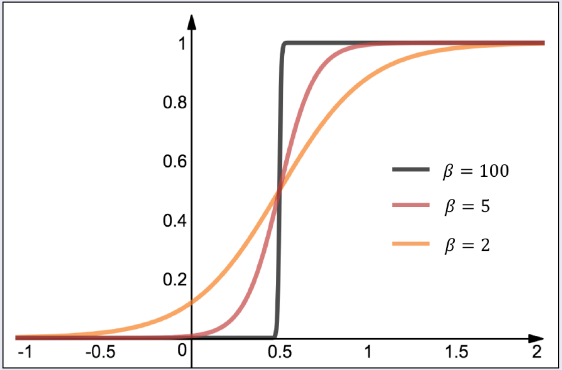
What is activation function $\sigma(x)$?

- strictly increasing
- $\sigma(\cdot) \in \mathbb{C}^1$ with Lipschitz constant L_{σ_i}

Example: tanh

$$\sigma_i(x) = \frac{1}{2}\tanh(\beta_i(x - \frac{1}{2})) + \frac{1}{2}; \qquad \beta_i > 0$$

"soft projection operator" from $\mathbb{R}$ to $\{0, 1\}$



★ ★ Hopfield Method $\Rightarrow$ Nonlinear Gradient Flow

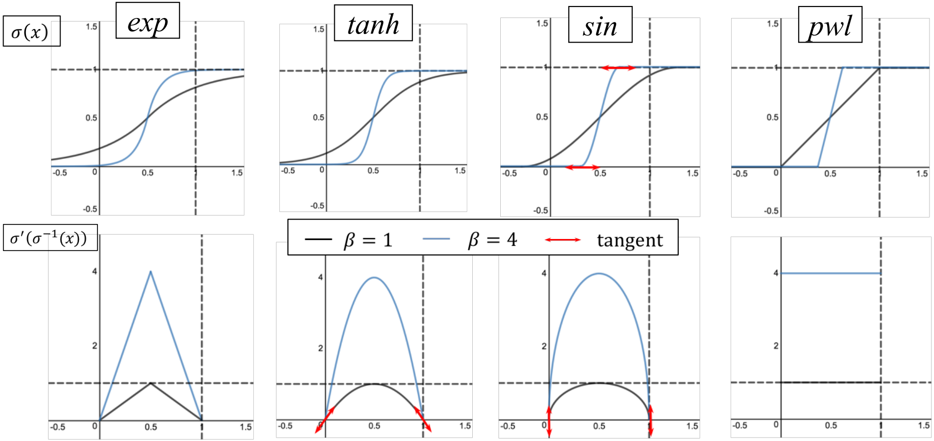
If $\sigma(\cdot)$ is a homeomorphism, then a **nonlinear gradient flow emerges**!

$$\frac{d}{dt}x(t) = -\sigma'(\sigma^{-1}(x(t))) \odot \nabla f(x(t)) \tag{10}$$

★ ★ Hopfield Method ⇒ Nonlinear Gradient Flow

If $\sigma(\cdot)$ is a homeomorphism, then a **nonlinear gradient flow emerges**!

$$\frac{d}{dt}x(t) = -\sigma'(\sigma^{-1}(x(t))) \odot \nabla f(x(t)) \tag{10}$$



Discretize time dynamics

Forward Euler time discretization of Hopfield dynamics:

$$x_H^{k+1} = x_H^k - \alpha^k \nabla f(x^k); \qquad x_H^0 = x^0 \in (0,1)^n \tag{11}$$

$$x^k = \sigma(x_H^k) \tag{12}$$

Discretize time dynamics

Forward Euler time discretization of Hopfield dynamics:

$$x_H^{k+1} = x_H^k - \alpha^k \nabla f(x^k); \qquad x_H^0 = x^0 \in (0,1)^n \tag{11}$$

$$x^k = \sigma(x_H^k) \tag{12}$$

For quadratic $f(x) = \frac{1}{2} x^T Q x$

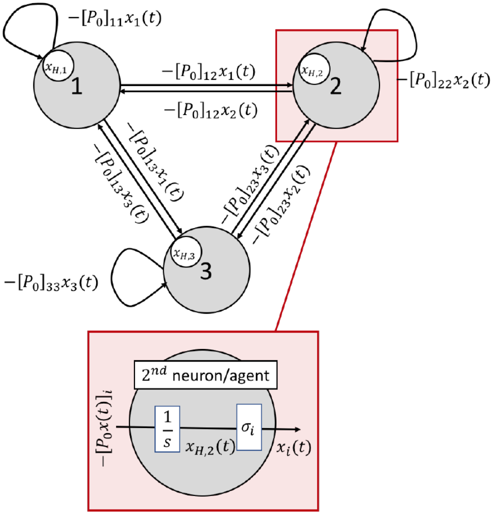
$$x_H^{k+1} = x_H^k - \alpha^k Q x^k; \qquad x_H^0 = x^0 \in (0,1)^n \tag{13}$$

$$x^k = \sigma(x_H^k) \tag{14}$$

Graphical Interpretation of Hopfield Method

Forward Simulation of Hopfield Neural Net!

- Undirected weighted graph
- n nodes, one for each x_i
- Each node has internal ($x_{H,i} \in \mathbb{R}$) and external ($x_i \in R$) states
- Weights $[P_0]_{ij}$ are elements of gradients of obj fcn



★ Hopfield vs Projected Gradient Descent

Hopfield

$$x_H^{k+1} = x_H^k - \alpha^k \nabla f(x^k) \qquad (15)$$

$$x^k = \sigma(x_H^k) \qquad (16)$$

Projected Gradient Descent

$$x_H^{k+1} = x^k - \alpha^k \nabla f(x^k) \qquad (17)$$

$$x^k = \text{Proj}_{[0,1]}(x_H^k) \qquad (18)$$

★ Hopfield vs Projected Gradient Descent

Hopfield

$$x_H^{k+1} = x_H^k - \alpha^k \nabla f(x^k) \quad (15)$$

$$x^k = \sigma(x_H^k) \quad (16)$$

Projected Gradient Descent

$$x^{k+1} = \text{Proj}_{[0,1]}(x^k - \alpha^k \nabla f(x^k)) \quad (17)$$

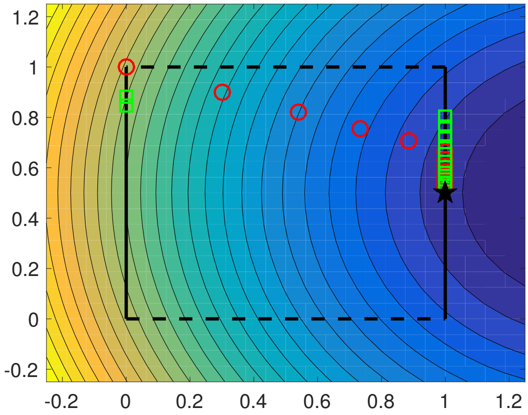
No dynamics!

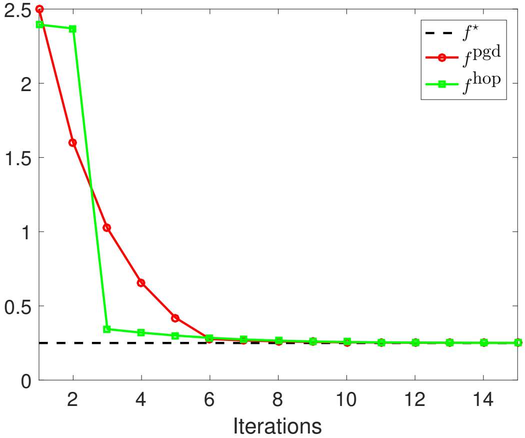
Simple Comparison

$$\text{minimize}_{x_1, x_2} \quad (x_1 - 1.5)^2 + (x_2 - 0.5)^2 \tag{18}$$

$$\text{subject to:} \quad x_1 \in \{0, 1\} \tag{19}$$

$$0 \leq x_2 \leq 1 \tag{20}$$





Outline

Continuous Improvement to a Fixed Point

Theorem 1: Continuous Improvement

The Hopfield method yields monotonically decreasing iterates, $f(x^{k+1}) \leq f(x^k), \forall\, k$ if ...

- activation fcn has Lipschitz continuous first derivative: $\sigma(\cdot) \in \mathbb{C}^1$ ($\text{exp}, \text{tanh}, \text{sin}$, pwl)
- step-size α^k follows an appropriately decreasing schedule

Specifically, the incremental improvement is bounded by:

$$0 \leq f(x^k) - f(x^{k+1}) \leq 0.5\alpha^k \cdot \nabla f(x^k)^T \Sigma^k \nabla f(x^k) \qquad \text{where} \quad \Sigma^k = \text{diag}(\sigma'(x_H^k))$$

Continuous Improvement to a Fixed Point

Theorem 1: Continuous Improvement

The Hopfield method yields monotonically decreasing iterates, $f(x^{k+1}) \leq f(x^k)$, $\forall\ k$ if ...

- activation fcn has Lipschitz continuous first derivative: $\sigma(\cdot) \in \mathbb{C}^1$ (exp, tanh, sin, pwl)
- step-size α^k follows an appropriately decreasing schedule

Specifically, the incremental improvement is bounded by:

$$0 \leq f(x^k) - f(x^{k+1}) \leq 0.5\alpha^k \cdot \nabla f(x^k)^T \Sigma^k \nabla f(x^k) \qquad \text{where} \quad \Sigma^k = \text{diag}(\sigma'(x_H^k))$$

Corollary: Convergence within a set

There exists a $f^\dagger$ such that $f(x^k) \to f^\dagger$ as $k \to \infty$, and x^k converges to the (non-empty) set

$$\mathcal{X} = \left\{ x \in [0,1]^n \mid x_i \in \{0,1\} \quad \text{OR} \quad \frac{\partial}{\partial x_i} f(x) = 0, \quad i = 1, \cdots, p \right\} \tag{21}$$

Remarks:

- Set $\mathcal{X}$ includes true minimizer $x^\star$, but $x^k \to x^\star$ not guaranteed

Convergence Rates

Theorem 2: Sub-linear convergence

If $f(x)$ is convex and $\sigma(\cdot)$ is smooth and verifies

$$\sigma'(\sigma^{-1}(x)) \geq \min\left\{|x|, |1-x|\right\}, \quad x \in [0,1]^n \tag{22}$$

then,

- $f(x^k) - f^\dagger = \mathcal{O}\left(\frac{1}{k^r}\right)$, with $0 < r < 1$
- To achieve precision ε, the worst case number of iterations is $2Mn/(\beta^2\varepsilon)$
  - M is upper-bound on Hessian: $\nabla^2 f(x) \preceq MI$
  - n is number of variables $x \in \mathbb{R}^n$
  - β is "hardness" of activation function

Remark: Slower than gradient descent, for which convergence is guaranteed at rate $\mathcal{O}\left(\frac{1}{k}\right)$

Outline

Dual Hopfield Method

So far, we have considered Hopfield methods to approximately solve

$$
\begin{aligned}
\text{minimize} \quad & f(x) && (23) \\
\text{subject to:} \quad & 0 \leq x_i \leq 1 \quad i = 1, \cdots, n && (24) \\
& x_i \in \{0, 1\} \quad i = 1, \cdots, p < n && (25)
\end{aligned}
$$

Dual Hopfield Method

So far, we have considered Hopfield methods to approximately solve

$$\begin{aligned} \text{minimize} \quad & f(x) && (23) \\ \text{subject to:} \quad & 0 \leq x_i \leq 1 \quad i = 1, \cdots, n && (24) \\ & x_i \in \{0, 1\} \quad i = 1, \cdots, p < n && (25) \end{aligned}$$

We now consider inequality constraints:

$$\begin{aligned} \text{minimize} \quad & f(x) && (26) \\ \text{subject to:} \quad & g_j(x) \leq 0, \quad j = 1, \cdots, m && (27) \\ & 0 \leq x_i \leq 1 \quad i = 1, \cdots, n && (28) \\ & x_i \in \{0, 1\} \quad i = 1, \cdots, p < n && (29) \end{aligned}$$

Idea: Instead of considering the "full" Lagrangian relaxation, consider

$$L(x, \mu) = f(x) + \sum_{j=1}^{m} \mu_j g_j(x) \tag{30}$$

Dual Hopfield Method

Apply Lagrangian relaxation

Idea: Instead of considering the "full" Lagrangian relaxation, consider

$$L(x, \mu) = f(x) + \sum_{j=1}^{m} \mu_j g_j(x) \tag{30}$$

Then the *dual function* is

$$D(\mu) = \min_{x} \quad L(x, \mu) = f(x) + \sum_{j=1}^{m} \mu_j g_j(x) \tag{31}$$

$$\text{subject to:} \quad 0 \leq x_i \leq 1 \quad i = 1, \cdots, n \tag{32}$$

$$x_i \in \{0, 1\} \quad i = 1, \cdots, p < n \tag{33}$$

which is amenable to Hopfield method, given μ.

Dual Ascent via Hopfield

Then solve the Dual Problem:

$$\max_{\mu \geq 0} D(\mu) \tag{34}$$

$$D(\mu) = \min_{x} L(x, \mu) = \min_{x} f(x) + \sum_{j=1}^{m} \mu_j g_j(x) \tag{35}$$

Dual Ascent via Hopfield

Then solve the Dual Problem:

$$\max_{\mu \geq 0} D(\mu) \tag{34}$$

$$D(\mu) = \min_{x} L(x, \mu) = \min_{x} f(x) + \sum_{j=1}^{m} \mu_j g_j(x) \tag{35}$$

Run Hopfield method to approximately solve $D(\mu) = \min_x L(x, \mu)$.

Suppose $x^\star(\mu) = \arg\min_x L(x, \mu)$.

The subgradient of $D(\mu)$ along dimension j: $g_j(x^\star(\mu)) \in \partial_j D(\mu)$

Algorithm 1 Dual (sub)-gradient Ascent via Hopfield Method

Initialize $\lambda^0 \geq 0$; Choose $\beta > 0$

for $k = 0, 1, \cdots, k_{\max}$

(1) use Hopfield method to approximately compute dual function

for $\ell = 0, \cdots, \ell_{\max}$

$x_H^{\ell+1} = x_H^{\ell} - \alpha^{\ell} \nabla_x L(x^{\ell}, \mu^k)$

$x^{\ell} = \sigma(x_H^{\ell+1})$

$x_{\text{hop}}^k \leftarrow x^{\ell}$

until stopping criterion is met

(2) update dual variable μ via (sub)-gradient ascent

$\mu^{k+1} = \mu^k + \beta^k \sum_{j=1}^{m} g_j(x_{\text{hop}}^k(\mu^k))$

end for

Outline

Examples: Random MIQPs

Consider solving MIQP w.r.t. $x \in \mathbb{R}^n$

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}x^T Q x + R^T x && (36) \\ \text{subject to:} \quad & Ax \leq b && (37) \\ & A_{eq}x = b_{eq} && (38) \\ & lb \leq x \leq ub && (39) \\ & x_i \in \{0, 1\}, \quad i = 1, \cdots, p && (40) \end{aligned}$$

- Randomly generated parameters $Q, R, A, b, A_{eq}, b_{eq}, lb, ub$ for each n
- Number of constraints also randomized

Comparative Analysis

All problems solved on Matlab:

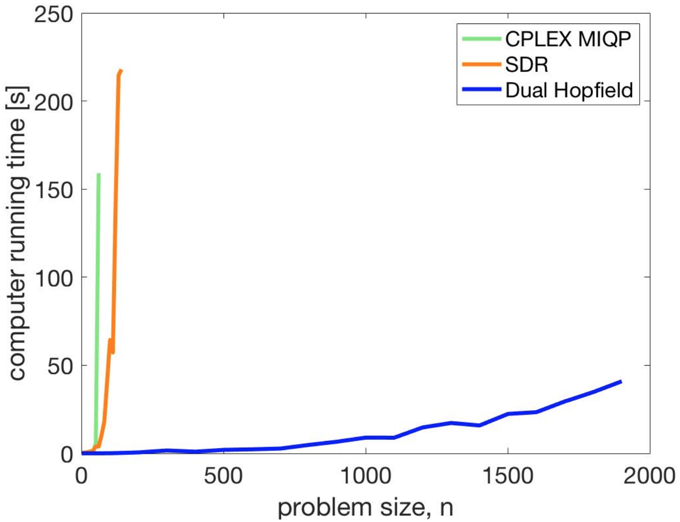
- CPLEX MIQP: using function *cplexmiqp* developed by IBM
- Binary Relaxation via CPLEX QP : using function *cplexqp*
- Semi-definite relaxation (SDR): corresponding SDP solved using *CVX*
- Hopfield: Dual Ascent Hopfield Method uses dual variables from *cplexqp*

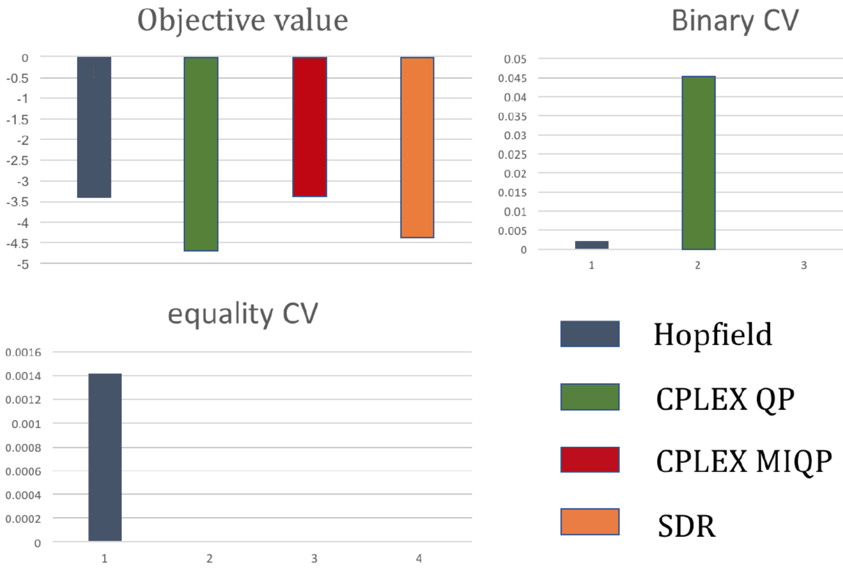
For each method, we compute:

- computer running time [sec]
- constraint violations (CV):
  - binary CV: $\frac{1}{p}\sum_{i=1}^{p} d(x_i, \{0,1\})$
  - inequality CV: $\frac{1}{m}\sum_{j=1}^{m} |[Ax - b]_j|$
  - equality CV: $\frac{1}{\ell}\sum_{k=1}^{\ell} |[A_{eq}x - b_{eq}]_k|$
- objective function value

Comparative Analysis

Computer running time




Objective value
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
-4
-4.5
-5
Binary CV
0.05
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
1
2
3
4
equality CV
0.0016
0.0014
0.0012
0.001
0.0008
0.0006
0.0004
0.0002
0
1
2
3
4
Hopfield
CPLEX QP
CPLEX MIQP
SDR

Application: Optimal Economic Dispatch of DERs

Consider n generators with cost: $f_i(x_i) = c_i x_i^2 + b_i x_i + a_i$, with $a_i, b_i, c_i \geq 0$.

The first p generators can only make binary decisions. That is:

- $\forall\, i \in \{1, p\}$ we have $x_i \in \{P_{i,\min}, P_{i,\max}\}$
- $\forall\, i \in \{p+1, n\}$ we have $x_i \in [P_{i,\min}, P_{i,\max}]$

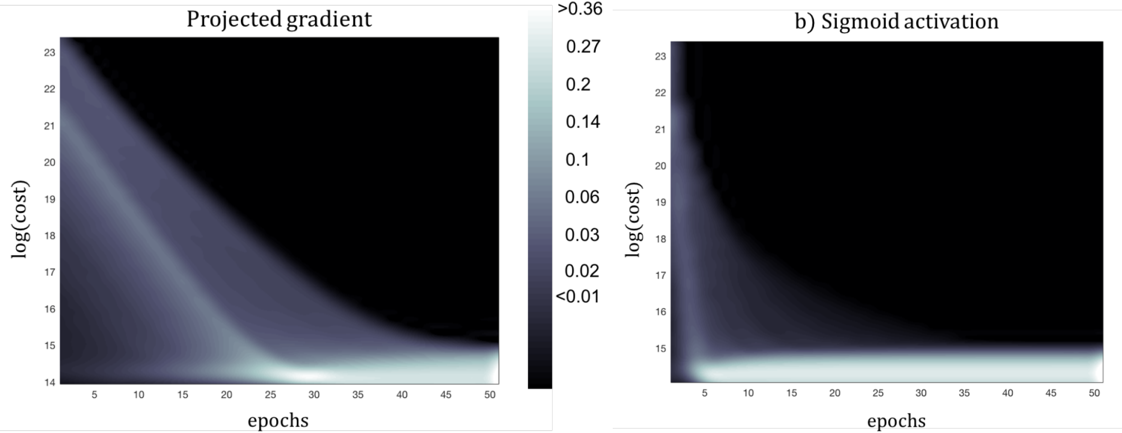
Problem Statement

Find the optimal dispatch for generators to minimize cost and meet demand:

$$\begin{aligned} \text{minimize} \quad & \sum_i^n f_i(x_i) \\ \text{subject to:} \quad & \sum_i^n x_i = D \\ & \text{(constraints above)} \end{aligned}$$

Simulation parameters: $n = 1000$ generators. Other parameters randomly generated.
We perform 5000 Monte-Carlo simulations.

Monte Caroline Simulation Results



SUMMARY

- Hopfield Methods for large-scale MINLPs – *An old heuristic with new analysis!*

EXTENSIONS

- Alternative descent direction
- Nesterov acceleration
- Chance constraints
- Distributed algorithms via dual decomposition
- ...

ON-GOING / FUTURE

- Application to Large-Scale PEV Charge Scheduling
- More comprehensive comparative analysis
- Open source codes! `hmip`

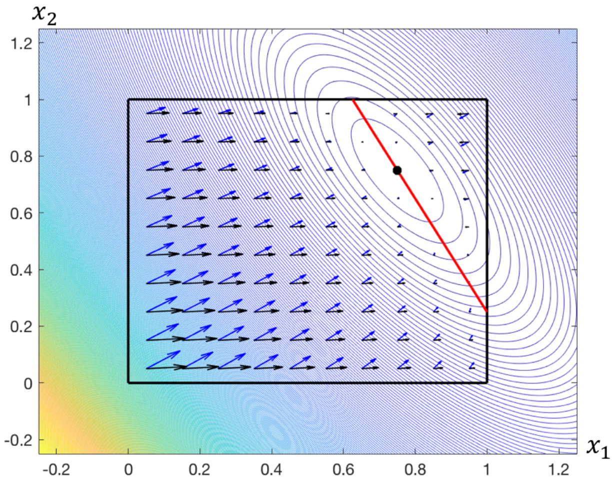
VISIT US!

Energy, Controls, and Applications Lab (eCAL)

ecal.berkeley.edu

smoura@berkeley.edu

APPENDIX SLIDES



$\nabla f_0(x)$ $\qquad$ $\nabla f_0(x)_1 = 0$

$\sigma'(\sigma^{-1}(x)) \odot \nabla f_0(x)$ $\qquad$ • $\nabla f_0(x) = 0$

Stochastic approach to recover integer constraint:

Let x^r be solution to binary relaxation. Feasible x can be drawn randomly from $\{0, 1\}$ following Bernoulli distribution $\mathcal{B}(x^r)$.

This can be sub-optimal.

Stochastic approach to recover integer constraint:

Let x^r be solution to binary relaxation. Feasible x can be drawn randomly from $\{0, 1\}$ following Bernoulli distribution $\mathcal{B}(x^r)$.

This can be sub-optimal.

Example

$$\text{minimize}_{x \in \{0,1\}} \left(x - \frac{1}{4} \right)^2 = \frac{1}{16} \quad (x^\star = 0 \;\text{ is the optimal solution})$$

If we apply binary relaxation, we get $x^r = \frac{1}{4}$ and $\mathbb{E}_{x \sim \mathcal{B}(x^r)} \left(x - \frac{1}{4} \right)^2 = \frac{3}{16} > \frac{1}{16}$!

Other ideas:

- Branch & Bound, Branch & Cut

Existing Methods

Convex Relaxation #2: Lagrangian Relaxation

Notice that $x_i \in \{0, 1\}$ is equivalent to satisfying $x_i(1 - x_i) = 0$

$$\text{minimize} \quad f(x) \tag{41}$$

$$\text{subject to:} \quad g_j(x) \leq 0, \quad j = 1, \cdots, m \tag{42}$$

$$0 \leq x \leq 1 \tag{43}$$

$$x_i(1 - x_i) = 0, \quad i = 1, \cdots, p < n \tag{44}$$

Notice that $x_i \in \{0, 1\}$ is equivalent to satisfying $x_i(1 - x_i) = 0$

$$\begin{aligned} \text{minimize} \quad & f(x) && (41) \\ \text{subject to:} \quad & g_j(x) \leq 0, \quad j = 1, \cdots, m && (42) \\ & 0 \leq x \leq 1 && (43) \\ & x_i(1 - x_i) = 0, \quad i = 1, \cdots, p < n && (44) \end{aligned}$$

Form the *Lagrangian*:

$$L(x, \mu, \underline{\mu}, \overline{\mu}, \lambda) = f(x) + \sum_{j=1}^{m} \left[\mu_j g_j(x) + \underline{\mu}_j x_i + \overline{\mu}_j (1 - x_i) \right] + \sum_{i=1}^{p} \lambda_i x_i (1 - x_i) \quad (45)$$

Notice that $x_i \in \{0, 1\}$ is equivalent to satisfying $x_i(1 - x_i) = 0$

$$\begin{aligned} \text{minimize} \quad & f(x) && (41) \\ \text{subject to:} \quad & g_j(x) \leq 0, \quad j = 1, \cdots, m && (42) \\ & 0 \leq x \leq 1 && (43) \\ & x_i(1 - x_i) = 0, \quad i = 1, \cdots, p < n && (44) \end{aligned}$$

Form the *Lagrangian*:

$$L(x, \mu, \underline{\mu}, \overline{\mu}, \lambda) = f(x) + \sum_{j=1}^{m} \left[\mu_j g_j(x) + \underline{\mu}_j x_i + \overline{\mu}_j (1 - x_i) \right] + \sum_{i=1}^{p} \lambda_i x_i (1 - x_i) \tag{45}$$

Define the (concave) *dual function* of $\Lambda = [\mu, \underline{\mu}, \overline{\mu}, \lambda]$

$$D(\Lambda) = \min_{x \in \mathbb{R}^n} L(x, \mu, \underline{\mu}, \overline{\mu}, \lambda) \tag{46}$$

Weak duality approach: Solve convex program $\max_{\Lambda} D(\Lambda)$

Existing Methods

Convex Relaxation #3: Semi-definite Relaxation

Introduce new variable $X = xx^T$. This is called "lifting". Can re-write MIQCQP

Existing Methods

Convex Relaxation #3: Semi-definite Relaxation

Introduce new variable $X = xx^T$. This is called "lifting". Can re-write MIQCQP

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\text{Tr}(QX) + R^T x + S && (47) \\ \text{subject to:} \quad & \frac{1}{2}\text{Tr}(Q_j X) + R_j^T x + S_j \leq 0, \quad j = 1, \cdots, m && (48) \\ & 0 \leq x \leq 1 && (49) \\ & X_{ii} = x_i, \quad i = 1, \cdots, p < n && (50) \\ & X = xx^T && (51) \end{aligned}$$

If Q, Q_i are positive semi-definite, then only $X = xx^T$ makes this non-convex.

Existing Methods

Convex Relaxation #3: Semi-definite Relaxation

Introduce new variable $X = xx^T$. This is called "lifting". Can re-write MIQCQP

$$\text{minimize} \quad \frac{1}{2}\text{Tr}(QX) + R^T x + S \tag{47}$$

$$\text{subject to:} \quad \frac{1}{2}\text{Tr}(Q_j X) + R_j^T x + S_j \leq 0, \quad j = 1, \cdots, m \tag{48}$$

$$0 \leq x \leq 1 \tag{49}$$

$$X_{ii} = x_i, \quad i = 1, \cdots, p < n \tag{50}$$

$$X = xx^T \tag{51}$$

If Q, Q_i are positive semi-definite, then only $X = xx^T$ makes this non-convex. Relax into convex inequality $X \succeq xx^T$. Using Schur complement:

$$X \succeq xx^T \Leftrightarrow \begin{bmatrix} X & x \\ x & 1 \end{bmatrix} \succeq 0 \tag{52}$$

This can be cast as a semi-definite program (SDP).