Lagrange Relaxation: Decomposition Algorithms

Operations Research

Anthony Papavasiliou

Contents

1. Context
2. Dual Function Optimization Algorithms
   - Subgradient Method
   - Cutting Plane Algorithm
   - Bundle Methods
   - Level Method
   - Numerical Comparison
3. Alternating Direction Method of Multipliers

Table of Contents

1. Context
2. Dual Function Optimization Algorithms
   - Subgradient Method
   - Cutting Plane Algorithm
   - Bundle Methods
   - Level Method
   - Numerical Comparison
3. Alternating Direction Method of Multipliers

When to Use Lagrange Relaxation

Consider the following optimization problem:

$$
\begin{aligned}
p^\star \quad = \quad & \max f_0(x) \\
& f(x) \leq 0 \\
& h(x) = 0
\end{aligned}
$$

with $x \in \mathcal{D} \subset \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}^m$, $h : \mathbb{R}^n \to \mathbb{R}^l$

Context for Lagrange relaxation:

1. *Complicating constraints* $f(x) \leq 0$ and $h(x) = 0$ make the problem difficult
2. Dual function is relatively easy to evaluate

$$g(u, v) = \sup_{x \in \mathcal{D}} (f_0(x) - u^T f(x) - v^T h(x)) \tag{1}$$

Idea of Dual Decomposition

- Dual function $g(u, v)$ is convex *regardless* of primal problem
- Computation of $g(u, v)$, $\pi \in \partial g(u, v)$ is relatively easy
- But... $g(u, v)$ may be non-differentiable

Idea: minimize $g(u, v)$ using algorithms that rely on linear approximation of $g(u, v)$:

1. Subgradient method
2. Cutting plane methods
3. Bundle methods
4. Level methods

and a closely related method: alternating direction of multipliers method (ADMM)

Dual Function Properties

Proposition: $g(u, v)$ is convex lower-semicontinous[1]. If (u, v) is such that (1) has optimal solution $x_{u,v}$, then $\begin{bmatrix} -f(x_{u,v}) \\ -h(x_{u,v}) \end{bmatrix}$ is a subgradient of g

[1] A function is lower-semicontinuous when its epigraph is a closed subset of $\mathbb{R}^m \times \mathbb{R}^l \times \mathbb{R}$.

Table of Contents

1 Context

2 Dual Function Optimization Algorithms

- Subgradient Method
- Cutting Plane Algorithm
- Bundle Methods
- Level Method
- Numerical Comparison

3 Alternating Direction Method of Multipliers

Subgradient Method

Subgradient method is simple algorithm to minimize non-differentiable convex function g

$$u_{k+1} = u_k - \alpha_k \pi_k$$

- u_k is the k-th iterate
- π_k is *any* subgradient of g at u_k
- $\alpha_k > 0$ is the k-th step size

Not a descent method, so we keep track of the best point so far

$$g_k^{\text{best}} = \min_{i=1,\ldots,k} g(u_i)$$

Step Size Rules

Step sizes are fixed ahead of time

- Constant step size: $\alpha_k = \alpha$ (constant)
- Constant step length: $\alpha_k = \gamma / \|\pi_k\|_2$ (so $\|u_{k+1} - u_k\|_2 = \gamma$)
- Square summable but not summable: step sizes satisfy

$$\sum_{k=1}^{\infty} \alpha_k^2 < \infty, \sum_{k=1}^{\infty} \alpha_k = \infty$$

- Non-summable diminishing: step sizes satisfy

$$\lim_{k \to \infty} \alpha_k = 0, \sum_{k=1}^{\infty} \alpha_k = \infty$$

Assumptions

- $d^\star = \inf_u g(u) > \infty$, with $g(u^\star) = d^\star$
- $\|\pi\|_2 \leq G$ for all $\pi \in \partial g$ (equivalent to Lipschitz condition on g)
- $R \geq \|u_1 - u^\star\|_2$

These assumptions are stronger than needed, just to simplify proofs

Convergence Results

Define $g_\infty = \lim_{k\to\infty} g_k^{\text{best}}$

- Constant step size: $g_\infty - d^\star \leq G\alpha^2/2$, i.e. converges to $G^2\alpha/2$-suboptimal (converges to $d^\star$ if g differentiable, α small enough)
- Constant step length: $g_\infty - d^\star \leq G\gamma/2$, i.e. converges to $G\gamma/2$-suboptimal
- Diminishing step size rule: $g_\infty = d^\star$, i.e. converges

Convergence Proof

Key quantity: *Euclidean distance to the optimal set*, not function value

Let $u^\star$ be any minimizer of g

$$
\begin{aligned}
\|u_{k+1} - u^\star\|_2^2 &= \|u_k - \alpha_k \pi_k - u^\star\|_2^2 \\
&= \|u_k - u^\star\|_2^2 - 2\alpha_k \pi_k^T (u_k - u^\star) + \alpha_k^2 \|\pi_k\|_2^2 \\
&\leq \|u_k - u^\star\|_2^2 - 2\alpha_k (g(u_k) - d^\star) + \alpha_k^2 \|\pi_k\|_2^2
\end{aligned}
$$

Using $d^\star = g(u^\star) \geq g(u_k) + \pi_k^T (u^\star - u_k)$

Apply recursively to get

$$\begin{aligned}
&\|u_{k+1} - u^\star\|_2^2 \\
&\leq \|u_1 - u^\star\|_2^2 - 2\sum_{i=1}^{k} \alpha_i (g(u_k) - d^\star) + \sum_{i=1}^{k} \alpha_i^2 \|\pi_i\|_2^2 \\
&\leq R^2 - 2\sum_{i=1}^{k} \alpha_i (g(u_i) - d^\star) + G^2 \sum_{i=1}^{k} \alpha_i^2
\end{aligned}$$

Now we use

$$\sum_{i=1}^{k} \alpha_i (g(u_i) - d^\star) \geq (g_k^{\text{best}} - d^\star)(\sum_{i=1}^{k} \alpha_i)$$

to get

$$g_k^{\text{best}} - d^\star \leq \frac{R^2 + G^2 \sum_{i=1}^{k} \alpha_i^2}{2\sum_{i=1}^{k} \alpha_i}$$

Constant step size: For $\alpha_k = \alpha$ we get

$$g_k^{\text{best}} - d^\star \leq \frac{R^2 + G^2 k \alpha^2}{2k\alpha}$$

Right hand side converges to $G^2\alpha/2$ as $k \to \infty$

Constant step length: for $\alpha_k = \gamma/\|\pi_k\|_2$ we get

$$g_k^{\text{best}} - d^\star \leq \frac{R^2 + G^2 \sum_{i=1}^k \alpha_i^2}{2\sum_{i=1}^k \alpha_i} \leq \frac{R^2 + \gamma^2 k}{2\gamma k/G}$$

Right hand side converges to $G\gamma/2$ as $k \to \infty$

Square summable but not summable step sizes: Suppose step sizes satisfy

$$\sum_{k=1}^{\infty} \alpha_k^2 < \infty, \sum_{k=1}^{\infty} \alpha_k = \infty$$

then

$$g_k^{\text{best}} - d^\star \leq \frac{R^2 + G^2 \sum_{i=1}^{k} \alpha_i^2}{2 \sum_{i=1}^{k} \alpha_i}$$

as $k \to \infty$, numerator converges to a finite number, denominator converges to ∞, so $g_k^{\text{best}} \to d^\star$

Polyak Step Size

- Choice due to Polyak:

$$\alpha_k = \frac{g(u_k) - d^\star}{\|\pi^{(k)}\|_2^2}$$

  (can also use when optimal value is estimated)

- Motivation: start with basic inequality

$$\|u_{k+1} - u^\star\|_2^2 \le \|u_k - u^\star\|_2^2 - 2\alpha_k(g(u_k) - d^\star) + \alpha_k^2\|\pi_k\|_2^2$$

  and choose α_k to minimize right hand side

- Yields

$$\|u_{k+1} - u^\star\|_2^2 \le \|u_k - u^\star\|_2^2 - \frac{(g(u_k) - d^\star)^2}{\|\pi_k\|_2^2}$$

(in particular $\|u_k - u^\star\|_2$ decreases at each step)

- Applying recursively,

$$\sum_{i=1}^{k} \frac{(g(u_i) - d^\star)^2}{\|\pi_i\|_2^2} \le R^2$$

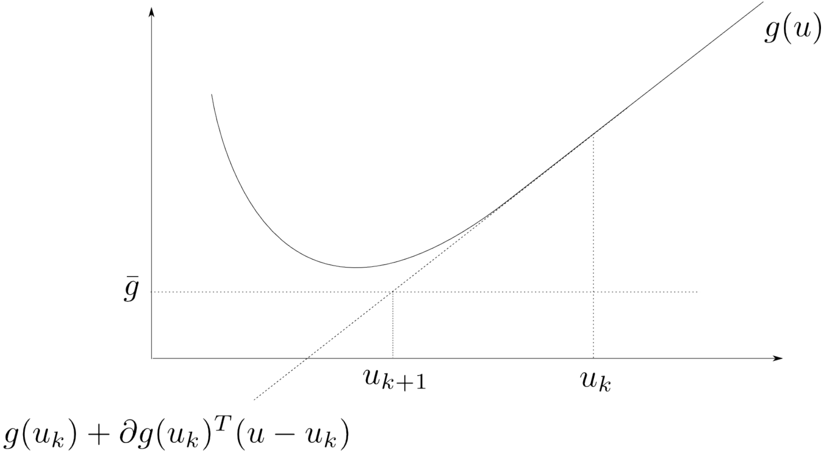
and so

$$\sum_{i=1}^{k} (g(u_i) - d^\star)^2 \le R^2 G^2$$

which proves $g(u_k) \to d^\star$

Graphical Illustration of Polyak Rule

$\bar{g}$ is an estimate of $d^\star$



Projected Subgradient Method

Solves constrained optimization problem

$$\min g(u)$$
$$\text{s.t. } u \in \mathcal{C}$$

where $g : \mathbb{R}^n \to \mathbb{R}, \mathcal{C} \subset \mathbb{R}^n$ are convex

Projected subgradient method is given by

$$u_{k+1} = P(u_k - \alpha_k \pi_k)$$

P is (Euclidean) projection on $\mathcal{C}$ and $\pi_k \in \partial g(u_k)$

Same convergence results:

- For constant step size, converges to neighborhood of optimal (for g differentiable and α small enough, converges)
- For diminishing summable step sizes, converges

Key idea: projection does not increase distance to $u^\star$

Motivation for Cutting Plane Algorithm

The subgradient algorithm uses subgradient information locally

Motivation for cutting plane algorithm: use subgradient information globally

Cutting plane algorithm, also known as **Kelley, Cheney, Goldstein** method, uses *bundle* of information $(g(u_k), \pi_k), k = 1, \ldots K$, where $\pi_k \in \partial g(u_k)$

Cutting Plane Algorithm

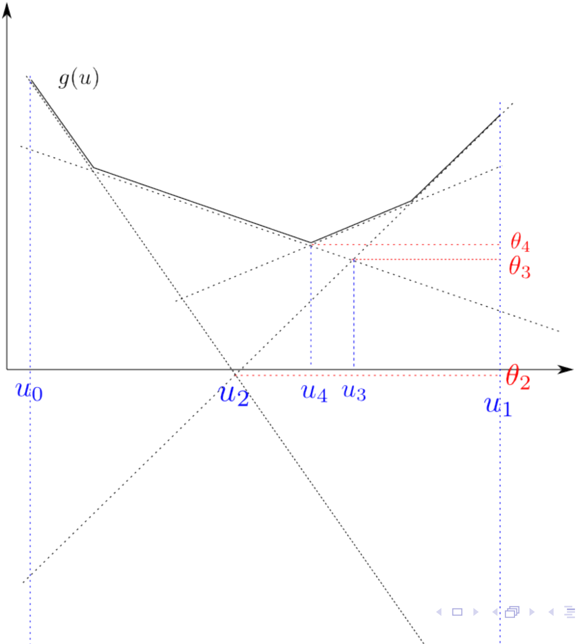
Define $\hat{g}(u) \leq g(u)$:

$$\hat{g}(u) = \min \theta$$
$$\text{s.t. } \theta \geq g(u_k) + \pi_k^T(u - u_k), k = 1, \ldots K$$

Given *bundle* of information $(g(u_k), \pi_k)$, $k = 1, \ldots, K$:

1. Solve $\min \hat{g}(u)$, denote u_{K+1} as optimal solution
2. Add u_{K+1}, $\pi_{K+1} \in \partial g(u_{K+1})$ to bundle
3. Return to step 1

Graphical Illustration



Observations

- θ_k is increasing
- $g(u_k)$ is not necessarily increasing
- Initialization requires restricting u within a confidence region
- Cutting plane algorithm is generally unstable
- L-shaped method is the cutting plane algorithm applied to two-stage stochastic linear programs

Analytic Center Cutting Plane Method (ACCPM)

Suppose dom $g \subseteq \mathbb{R}^m$ and consider the polyhedron in $\mathbb{R}^{m+1}$:

$$\begin{aligned} \mathcal{P}_K &= \{(u,\theta) : \hat{g}(u) \leq \theta \leq g_K^{\text{best}}\} \\ &= \{(u,\theta) : g(u_k) + \pi_k^T(u - u_k) \leq \theta \leq g_K^{\text{best}} \text{ for } k = 1,\ldots,K\} \end{aligned}$$

$\mathcal{P}_K$ is a polyhedron in $\mathbb{R}^{m+1}$

- Cutting plane method takes (u_{K+1}, θ_{K+1}) as point with lowest θ in $\mathcal{P}_K$, ...
- ... but this is unstable
- Instead, analytic center cutting plane method takes a 'central' point in $\mathcal{P}_K$

Analytic Center

Analytic center of polyhedron $\mathcal{P} = \{u : a_i^T u \leq b_i, i = 1, \ldots, m\}$:

$$AC(\mathcal{P}) = \text{argmin}_u - \sum_{i=1}^{m} \log(b_i - a_i^T u)$$

ACCPM Algorithm

Given an initial polyhedron $\mathcal{P}_0$

$k := 0$

Repeat

Compute $u_{k+1} = AC(\mathcal{P}_k)$

Query cutting-plane oracle at u_{k+1}

If u_{k+1} optimal, quit

Else, let $g_{k+1}^{\text{best}} \leftarrow \min(g_k^{\text{best}}, g(u_{k+1}))$ and add returned cutting-plane inequality to $\mathcal{P}_k$:

$$\mathcal{P}_{k+1} := \mathcal{P}_k \cap \{\theta \geq g(u_{k+1}) + \pi_{k+1}^T (u - u_{k+1})\}$$

If $\mathcal{P}_{k+1} = \emptyset$, quit

$k := k + 1$

Stopping Criterion

Since ACCPM is not a descent method, we keep track of best point found, and best lower bound

- Best function value so far: $g_k^{\text{best}} = \min_{i=1,\ldots,k} g(u_i)$
- Best lower bound so far: $\theta_k^{best} = \max_{i=1,\ldots,k} \theta_i^\star$
- Can stop when $g_k^{\text{best}} - \theta_k^{best} \leq \epsilon$
- Guaranteed to be ϵ-suboptimal

Bundle Methods

Rationale of bundle methods:

- Choose a *stability center* $\hat{u}$, that we believe is near-optimal
- Because $\hat{g}$ may be highly inaccurate ($\hat{g} \ll g$), minimizing $\hat{g}$ may result in u_{K+1} very far from $\hat{u}$
- Idea: add quadratic stabilizing term $\|u - \hat{u}\|^2$

Define

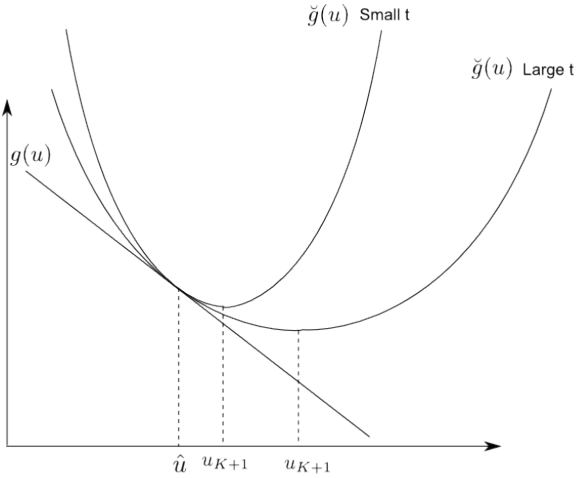
$$\breve{g}(u) = \hat{g}(u) + \frac{1}{2t}\|u - \hat{u}\|^2$$

and solve

$$(BP): \quad \min_{(u,\theta)\in\mathbb{R}^{m+1}} \theta + \frac{1}{2t}\|u - \hat{u}\|^2$$
$$\theta \geq g(u_k) + \pi_k^T(u - u_k), k = 1, \ldots, K$$

Denote (u_{K+1}, θ_{K+1}) as *unique* optimal solution

Graphical Illustration



Small t $\Rightarrow$ small steps, large t $\Rightarrow$ large steps

Key quantities:

$$\begin{aligned} \delta &:= g(\hat{u}) - \hat{g}(u_{K+1}) \\ \breve{\delta} &:= g(\hat{u}) - \breve{g}(u_{K+1}) = \delta - \frac{1}{2t}\|u_{K+1} - \hat{u}\|^2 \end{aligned}$$

Both are predictions of $g(\hat{u}) - g(u_{K+1})$

Stability Center Update

Consider the following condition:

$$g(u_{K+1}) \leq g(u_K) - \kappa\delta \tag{2}$$

where κ is a fixed tolerance

Two possibilities:

- If (2) is true, *descent step*: set $\hat{u} := u_{K+1}$
- If (2) is not true, *null step*: do not change $\hat{u}$ and update bundle with $(g(u_{K+1}), \pi_{K+1})$

Termination

Note:

$$0 \in \partial \hat{g}(u_{K+1}) + \frac{1}{t}(u_{K+1} - \hat{u})$$

so $\hat{\pi} \in \partial \hat{g}(u_{K+1})$ is computable as

$$\hat{\pi} = (\hat{u} - u_{K+1})/t$$

The following inequality is obtained, for any $u \in \mathbb{R}^m$:

$$\begin{aligned} g(u) \geq \hat{g}(u) &\geq \hat{g}(u_{K+1}) + \hat{\pi}^T(u - u_{K+1}) \\ &= g(\hat{u}) - \delta + \hat{\pi}^T(u - u_{K+1}) \end{aligned}$$

Terminate when both δ and $\hat{\pi}$ are small

Bundle Method Algorithm

$k := 0$

Repeat

Compute u_{K+1} solving (BP)

If δ and $\hat{\pi}$ are sufficiently small, quit

If equation (2) is true, perform *descent step*, else perform *null step*

$k := k + 1$

Motivation of Level Method

Consider a level L_k, then the **level set** of $\hat{g}$ is $\{u \in \mathbb{R}^m : \hat{g}(u) \leq L_k\}$

Idea of level method: project current iterate u_k on $\{u : \hat{g}(u) \leq L_k\}$

Justification:

- minimizer of $\hat{g}$ can be highly unstable, but level set of $\hat{g}$ is relatively stable
- projections are computationally "cheap"

Choosing Level Sets

Recall the following definitions:

$$\begin{aligned} g_k^{\text{best}} &= \min_{i=1,\dots k} g(u_i) \\ \theta_k^{\text{best}} &= \max_{i=1,\dots,k} \theta_i^\star \end{aligned}$$

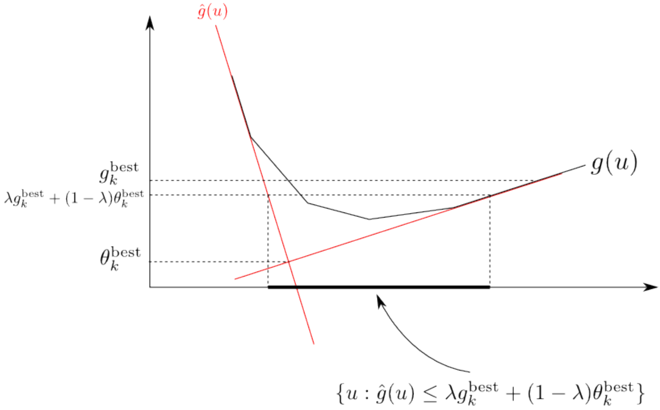
and consider the following level set of $\hat{g}$, parametrized on λ:

$$L_k = \lambda g_k^{\text{best}} + (1-\lambda)\theta_k^{\text{best}}$$

Consider two extremes:

- For $\lambda = 0$, algorithm makes no progress
- For $\lambda = 1$, algorithm reduces to cutting plane method

Graphical Interpretation



Level Set Algorithm

$k := 0$

Repeat

Compute u_{k+1} by solving

$$\min \|u - u_k\|_2^2$$
$$g(u_i) + \pi_i^T (u - u_i) \geq L_k,\ i = 1, \ldots, k$$

Add $(g(u_{k+1}), \pi_{k+1})$ to bundle, where $\pi_{k+1} \in \partial g(u_{k+1})$

Update $\theta^{\text{best}}_{k+1}, g^{\text{best}}_{k+1}$

If $g^{\text{best}}_{k+1} - \theta^{\text{best}}_{k+1} < \epsilon$, quit

$k := k + 1$

Convergence Result

Denote

- L: Lipschitz constant of g
- R: diameter of domain of g
- c: a constant that depends only on λ of level method

To obtain a gap smaller than ϵ, it suffices to perform

$$M(\epsilon) \leq c(\frac{LD}{\epsilon})$$

iterations

Case Study

Unit commitment on Belgian power system:

- 62 generators (nuclear, gas, biomass, oil)
- Demand (2014) net of wind, solar, hydro

Three cases:

- Case 1: high demand
- Case 2: medium demand
- Case 3: low demand

Unit commitment problem

$$\min \sum_{i \in I} C_i(x_i)$$

$$x_i \in \mathcal{D}_i, i \in I$$

$$(u^t): \quad \sum_{i \in I} c_i^t(x_i^t) \leq 0, t = 1, \ldots, T$$

Relax *complicating constraints* to obtain the following Lagrangian:

$$L(x, u) = \sum_{i \in I} (C_i(x_i) + \sum_{t=1}^{T} u^t c_i^t(x_i^t))$$

What have we gained? We can solve one problem per plant:

$$\min_{x_i \in \mathcal{D}_i} (C_i(x_i) + \sum_{t=1}^{T} u^t c_i^t(x_i^t))$$

Termination Criterion

	$\|u - u^\star\|_2$	$\|u - u^\star\|_\infty$	iter
	$\epsilon = 0.01$		
Level	10.0	4.8	19
ACCPM	20.7	6.1	38
	$\epsilon = 0.001$		
Level	8.3	4.7	33
ACCPM	8.8	3.7	192
	$\epsilon = 0.0005$		
Level	9.7	4.9	48
ACCPM	7.7	4.6	249

Table: Case 1

	$\|u - u^\star\|_2$	$\|u - u^\star\|_\infty$	iter
	$\epsilon = 0.01$		
Level	6.8	3.4	22
ACCPM	16.9	6.7	52
	$\epsilon = 0.001$		
Level	3.2	1.2	49
ACCPM	6.4	2.2	211
	$\epsilon = 0.0005$		
Level	3.1	1.4	36
ACCPM	5.8	1.9	253

Table: Case 2

	$\|u - u^\star\|_2$	$\|u - u^\star\|_\infty$	iter
	$\epsilon = 0.01$		
Level	7.5	3.2	19
ACCPM	17.7	6.7	54
	$\epsilon = 0.001$		
Level	1.7	0.8	45
ACCPM	5.4	2.1	240
	$\epsilon = 0.0005$		
Level	1.9	1.0	57
ACCPM	3.8	1.3	284

Table: Case 3

Prices

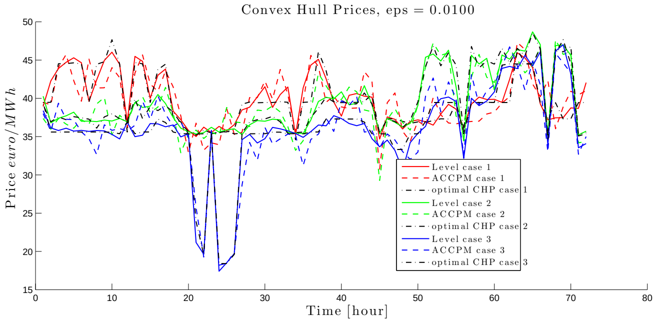


Figure: Prices for $\epsilon = 0.01$

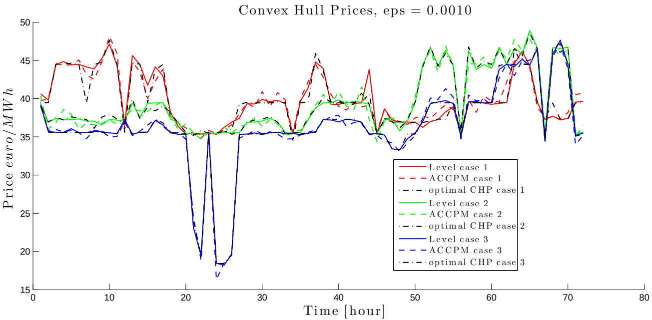


Figure: Prices for $\epsilon = 0.001$

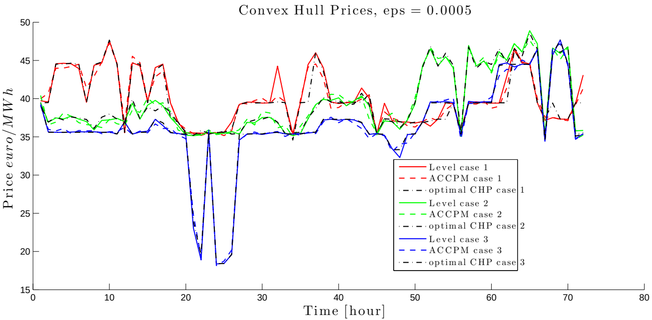


Figure: Prices for $\epsilon = 0.0005$

Observations

Conclusions:

- Level method converges in fewer iterations
- Dual multipliers that achieve target ϵ are too unstable for $\epsilon = 0.01$, very stable for $\epsilon = 0.0005$

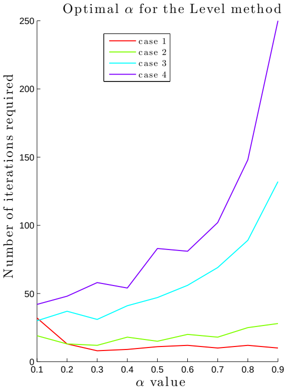
Parameter Tuning for the Level Method

Recall the trade-off in tuning λ for the level method:

- For $\lambda = 0$, algorithm makes no progress
- For $\lambda = 1$, algorithm reduces to cutting plane method

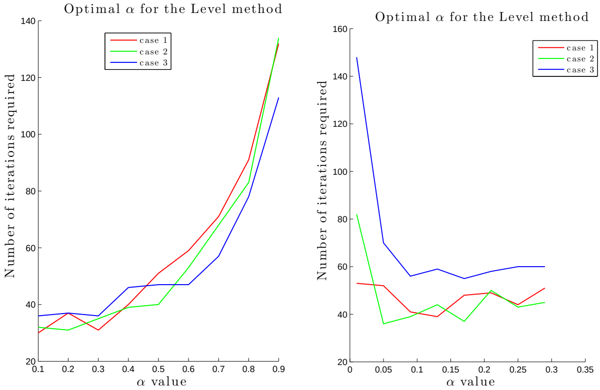
We want to find a suitable intermediate value

Figure: Required iterations for horizon of 2, 5, 24 and 72 periods. Note $\alpha = 1 - \lambda$.



Intuitive result: Cutting plane method works well only in low dimensions

Figure: Level method performance for two different shapes of demand curves for 72 period horizon



Conclusion: pick $\alpha = 1 - \lambda = 0.2$

Convergence Behavior

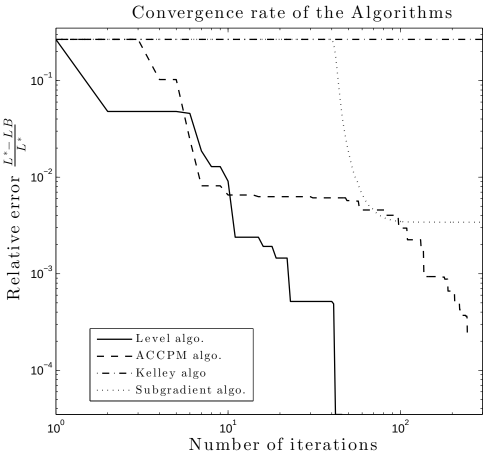


Figure: Convergence on 72-period instance

Volatility of the Iterate Sequence

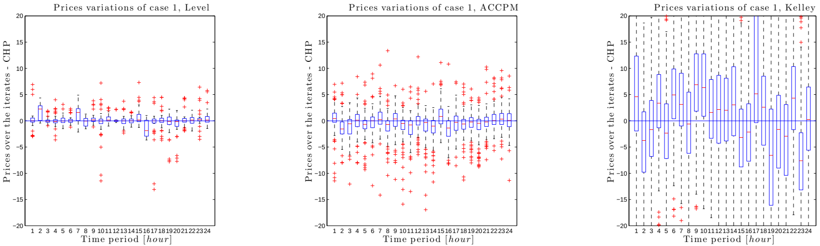


Figure: Box plots of iterates on 72-period instance, low demand (case 1)

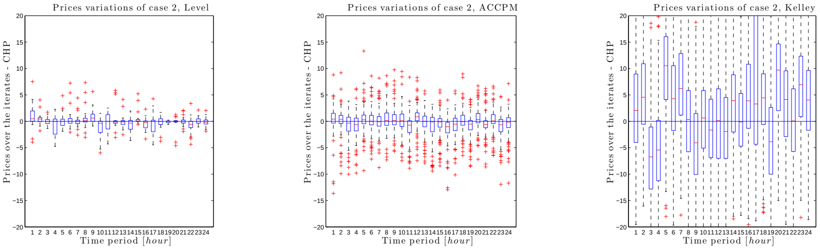


Figure: Box plots of iterates on 72-period instance, medium demand (case 2)

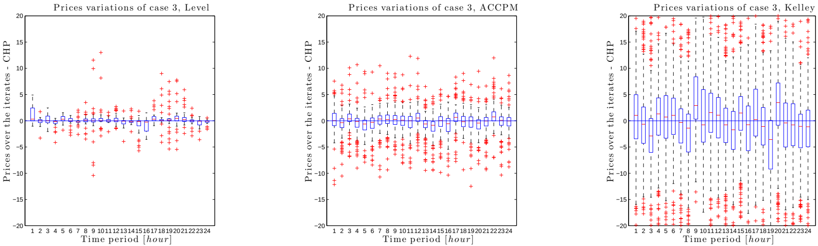


Figure: Box plots of iterates on 72-period instance, high demand (case 3)

Conclusions of Numerical Analysis

Level method and ACCPM dominate subgradient and cutting plane method in terms of

- convergence rate
- volatility of iterates

in large-scale problems

Table of Contents

1. Context

2. Dual Function Optimization Algorithms
   - Subgradient Method
   - Cutting Plane Algorithm
   - Bundle Methods
   - Level Method
   - Numerical Comparison

3. Alternating Direction Method of Multipliers

Alternating Direction Method of Multipliers

ADMM problem form (with f, ϕ convex)

$$\min f(x) + \phi(z)$$
$$\text{s.t. } Ax + Bz = c$$

Two sets of variables, with separable objective

Augmented Lagrangian:

$$L_\rho(x, z, \nu) = f(x) + \phi(z) + \nu^T(Ax + Bz - c) + (\rho/2)\|Ax + Bz - c\|_2^2$$

ADMM:

- x-minimization: $x_{k+1} = \text{argmin}_x L_\rho(x, z_k, \nu_k)$
- z-minimization: $z_{k+1} = \text{argmin}_z L_\rho(x_{k+1}, z, \nu_k)$
- Dual update: $\nu_{k+1} = \nu_k + \rho(Ax_{k+1} + Bz_{k+1} - c)$

ADMM and Optimality Conditions

Optimality conditions (for differentiable case):

- Primal feasibility: $Ax + Bz - c = 0$
- Dual feasibility: $\nabla f(x) + A^T \nu = 0, \quad \nabla g(z) + B^T \nu = 0$

Since z_{k+1} minimizes $L_\rho(x_{k+1}, z, \nu_k)$ we have

$$\begin{aligned} 0 &= \nabla g(z_{k+1}) + B^T \nu_k + \rho B^T (A x_{k+1} + B z_{k+1} - c) \\ &= \nabla g(z_{k+1}) + B^T \nu_{k+1} \end{aligned}$$

So with ADMM dual variable update, $(x_{k+1}, z_{k+1}, y_{k+1})$ satisfies second dual feasibility condition

Primal and first dual feasibility condition are achieved as $k \to \infty$

Convergence

Assume (very little):

- f, g convex, closed, proper
- L_0 has a saddle point

Then ADMM converges:

- iterates approach feasibility: $Ax_k + Bz_k - c \to 0$
- Objective approaches optimal value: $f(x_k) + \phi(x_k) \to p^\star$