# An Alternating Direction Method for Dual MAP LP Relaxation

Ofer Meshi and Amir Globerson

The Hebrew University of Jerusalem, Israel

# The MAP Problem

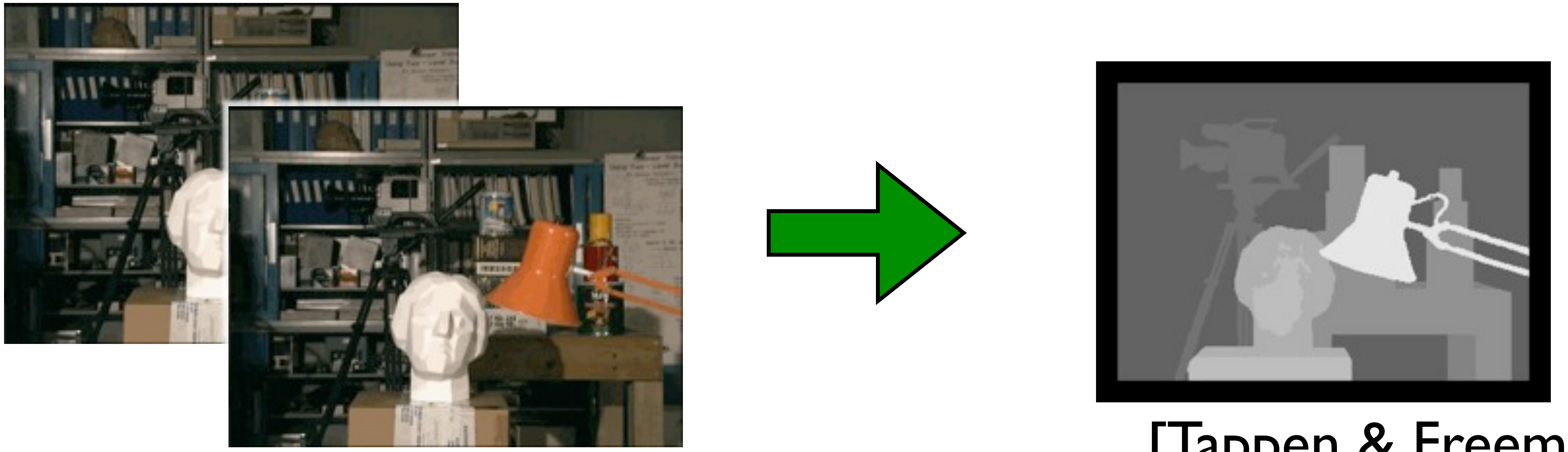
- Many machine learning tasks can be cast as finding the most probable configuration in a probabilistic graphical model.
- Applications
  - Language processing
  - Computer vision
  - Computational biology
  - Many others

# The MAP Problem

- **POS tagging**: given a sentence find the part-of-speech for each word

| | | | |
|---|---|---|---|
| John | hit | the | ball |
| Noun | Verb | Det | Noun |

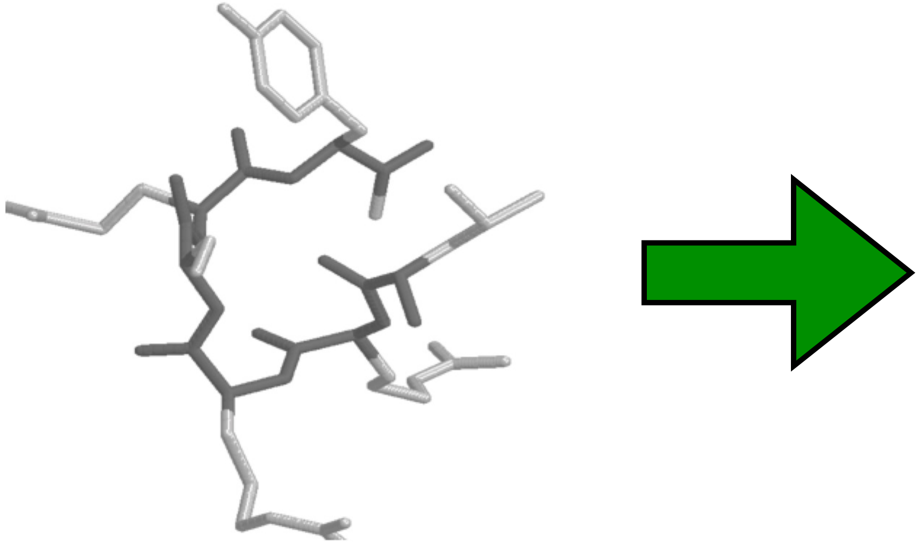
- **Stereo vision**: given left and right images, find the disparity of each pixel



[Tappen & Freeman 2003]

# MAP for Protein Design

- Given 3D shape, find amino-acids with most stable structure

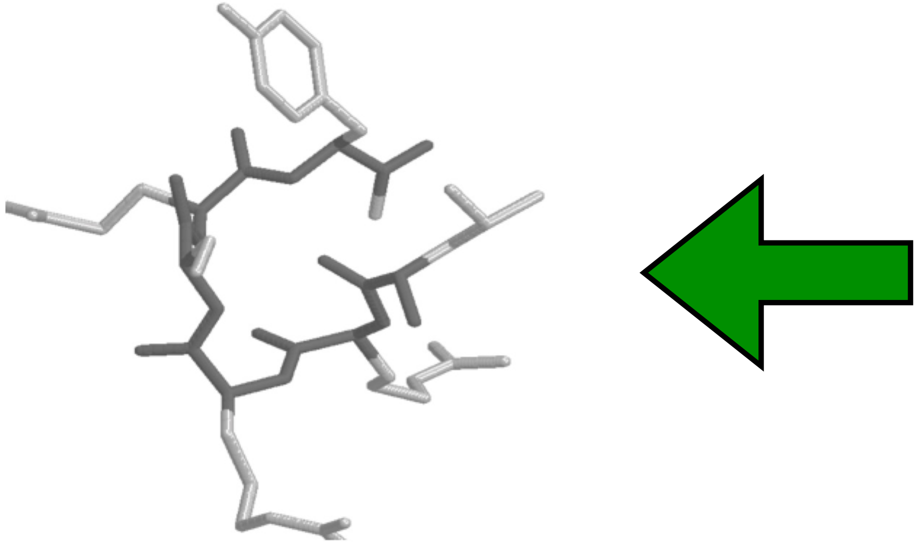


KDLMHNKCYHFFM

[Yanover et al. 2006]

# MAP for Protein Design

- Given 3D shape, find amino-acids with most stable structure



KDLMHNKCYHFFM

- Side-chain prediction (the inverse problem): given a sequence of amino-acids find most stable 3D configuration

[Yanover et al. 2006]

# The MAP Problem

- NP hard for general models
- LP relaxations often provide good approximations
- Off-the-shelf LP solvers are slow in practice due to problem size [Yanover et al. 2006]
- Several algorithms suggested, but many are either slow or not globally convergent

# Our Work

- A novel algorithm for MAP with LP relaxation
- Guaranteed to globally converge
- Scalable - efficient closed-form iterative local updates
- Based on the dual linear programming relaxation
- Uses the alternating direction method of multipliers (ADMM)

# The MAP Problem

- Formally:
  - Discrete variables $X_1, \ldots, X_n$
  - Markov random field (MRF)

$$p(x) \propto \exp\left(\sum_i \underset{\uparrow}{\theta_i(x_i)} + \sum_c \underset{\uparrow}{\theta_c(x_c)}\right)$$

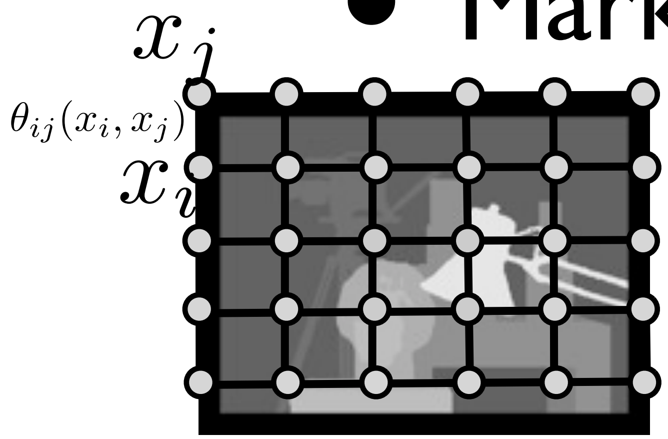
Singleton factors

High-order factors

Example: pairwise factor $\theta_{ij}(x_i, x_j)$

# The MAP Problem

- Formally:
  - Discrete variables $X_1, \ldots, X_n$
  - Markov random field (MRF)



$$p(x) \propto \exp\left(\sum_i \theta_i(x_i) + \sum_c \theta_c(x_c)\right)$$

  - Find the most probable configuration (maximum a-posteriori):

$$\arg\max_x \sum_i \theta_i(x_i) + \sum_c \theta_c(x_c)$$

# MAP and LP Relaxation

- General MAP is NP-hard!
- LP relaxation

$$
\begin{aligned}
&\max_x \sum_i \theta_i(x_i) + \sum_c \theta_c(x_c) \\
&= \max_{\mu \in M(G)} \sum_i \sum_{x_i} \mu_i(x_i)\theta_i(x_i) + \sum_c \sum_{x_c} \mu_c(x_c)\theta_c(x_c) \\
&= \max_{\mu \in M(G)} \mu \cdot \theta
\end{aligned}
$$

↑

Realizable marginal distributions

$$\mu_i(x_i) = p(x_i)$$

$$\mu_c(x_c) = p(x_c)$$

$$\text{for some } p$$

[Wainwright & Jordan 2010]

# MAP and LP Relaxation

- General MAP is NP-hard!
- LP relaxation

$$\max_x \sum_i \theta_i(x_i) + \sum_c \theta_c(x_c)$$

$$= \max_{\mu \in M(G)} \sum_i \sum_{x_i} \mu_i(x_i)\theta_i(x_i) + \sum_c \sum_{x_c} \mu_c(x_c)\theta_c(x_c)$$

$$= \max_{\mu \in \cancel{M(G)}} \mu \cdot \theta$$

$$L(G) = \left\{ \mu \geq 0 \;\middle|\; \begin{array}{ll} \sum_{x_{c\setminus i}} \mu_c(x_{c\setminus i}, x_i) = \mu_i(x_i) & \forall c, i : i \in c, x_i \\ \sum_{x_i} \mu_i(x_i) = 1 & \forall i \end{array} \right\}$$

[Wainwright & Jordan 2010]

# MAP and LP Relaxation

- General MAP is NP-hard!
- LP relaxation

$$\max_x \sum_i \theta_i(x_i) + \sum_c \theta_c(x_c)$$

$$= \max_{\mu \in M(G)} \sum_i \sum_{x_i} \mu_i(x_i)\theta_i(x_i) + \sum_c \sum_{x_c} \mu_c(x_c)\theta_c(x_c)$$

$$\leq \max_{\mu \in \cancel{M(G)}} \mu \cdot \theta$$

$$L(G) = \left\{ \mu \geq 0 \;\middle|\; \begin{array}{ll} \sum_{x_{c\setminus i}} \mu_c(x_{c\setminus i}, x_i) = \mu_i(x_i) & \forall c, i : i \in c, x_i \\ \sum_{x_i} \mu_i(x_i) = 1 & \forall i \end{array} \right\}$$

[Wainwright & Jordan 2010]

# MAP and LP Relaxation

- General MAP is NP-hard!
- LP relaxation

$$\max_x \sum_i \theta_i(x_i) + \sum_c \theta_c(x_c)$$

$$= \max_{\mu \in M(G)} \sum_i \sum_{x_i} \mu_i(x_i)\theta_i(x_i) + \sum_c \sum_{x_c} \mu_c(x_c)\theta_c(x_c)$$

$$\leq \max_{\mu \in \cancel{M(G)}} \mu \cdot \theta$$

$$L(G) = \left\{ \mu \geq 0 \;\middle|\; \begin{array}{ll} \sum_{x_{c \setminus i}} \mu_c(x_{c \setminus i}, x_i) = \mu_i(x_i) & \forall c, i : i \in c, x_i \\ \sum_{x_i} \mu_i(x_i) = 1 & \forall i \end{array} \right\}$$

- Tractable, BUT off-the-shelf LP solvers do poorly [Yanover et al. 2006]

[Wainwright & Jordan 2010]

# The Dual LP

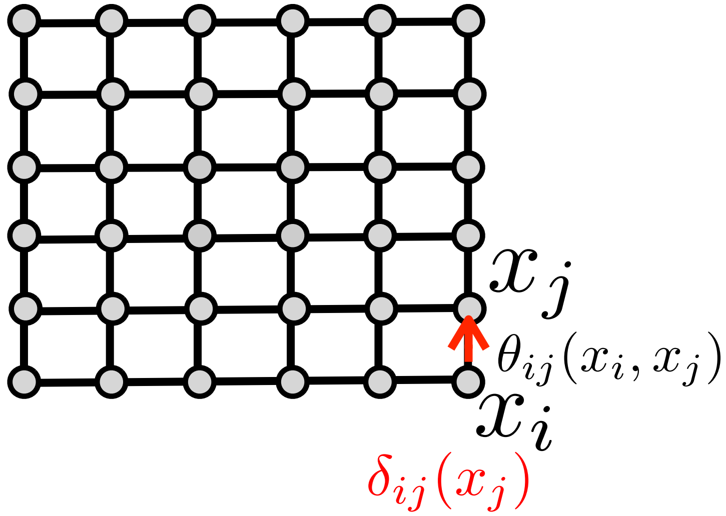
- Focus on the pairwise case.
- Dual variables $\delta_{ij}(x_j)$ for each directed edge
- Define: $\bar{\theta}_{ij}(x_i,x_j) = \theta_{ij}(x_i,x_j) - \delta_{ij}(x_j) - \delta_{ji}(x_i)$

$$\bar{\theta}_i(x_i) = \theta_i(x_i) + \sum_{k \in N(i)} \delta_{ki}(x_i)$$

- Dual Objective:

$$\min_{\delta} \sum_i \max_{x_i} \bar{\theta}_i(x_i) + \sum_{ij} \max_{x_i,x_j} \bar{\theta}_{ij}(x_i,x_j)$$

- Maximum is over small elements



# Dual MAP-LP

- The Lagrangian dual problem

$$\min_{\delta} \sum_{i} \max_{x_i} \underbrace{\left(\theta_i(x_i) + \sum_{c:i\in c} \delta_{ci}(x_i)\right)}_{\bar{\theta}_i(x_i)} + \sum_{c} \max_{x_c} \underbrace{\left(\theta_c(x_c) - \sum_{i:i\in c} \delta_{ci}(x_i)\right)}_{\bar{\theta}_c(x_c)}$$

- Two common algorithms:

[Werner 2007; Sontag et al. 2010]

# Dual MAP-LP

- The Lagrangian dual problem

$$\min_{\delta} \sum_{i} \max_{x_i} \underbrace{\left(\theta_i(x_i) + \sum_{c:i\in c} \delta_{ci}(x_i)\right)}_{\bar{\theta}_i(x_i)} + \sum_{c} \max_{x_c} \underbrace{\left(\theta_c(x_c) - \sum_{i:i\in c} \delta_{ci}(x_i)\right)}_{\bar{\theta}_c(x_c)}$$

- Two common algorithms:
  - Coordinate descent - fast but might get stuck
    - [Schlesinger 1976; Werner 2007; Globerson & Jaakkola 2007]
  - Subgradient - slow convergence
    - [Komodakis et al. 2007]

# Globally Optimal Methods

- Subgradient descent [Komodakis et al. 2007]
- Alternative solution: smooth the objective by adding local entropy terms
  - Can then apply gradient descent [Johnson 2008; Hazan & Shashua 2010]
  - Or accelerated gradient-based method [Jojic et al. 2010]. Temperature parameter might affect accuracy of the solution or cause numerical issues
  - Our approach: Augmented Lagrangians

# Augmented Lagrangians

- Optimization problem:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{s.t.} & Ax = b \end{array}$$

- Equivalent (augmented) problem:

$$\begin{array}{ll} \text{minimize} & f(x) + \frac{\rho}{2}\|Ax - b\|^2 \\ \text{s.t.} & Ax = b \end{array}$$

Quadratic smoothing

- Augmented Lagrangian:

$$\mathcal{L}_\rho(x, \nu) = f(x) + \nu^\top (Ax - b) + \frac{\rho}{2}\|Ax - b\|^2$$

[Gabay & Mercier 1976; Glowinski & Marroco 1975; Boyd et al. 2011]

# Augmented Lagrangians

- Augmented Lagrangian:

$$\mathcal{L}_\rho(x,\nu) = f(x) + \nu^\top(Ax-b) + \frac{\rho}{2}\|Ax-b\|^2$$

- Dual problem:

$$\max_\nu \min_x \mathcal{L}_\rho(x,\nu)$$

- Dual Subgradient ascent:

$$\nu_{t+1} = \nu_t + \epsilon(Ax_{t+1} - b)$$

$$x_{t+1} = \arg\max_x \mathcal{L}_\rho(x,\nu_t)$$

# Augmented Lagrangians

- Augmented Lagrangian:

$$\mathcal{L}_\rho(x,\nu) = f(x) + \nu^\top (Ax - b) + \frac{\rho}{2}\|Ax - b\|^2$$

- Dual problem:

$$\max_{\nu} \min_{x} \mathcal{L}_\rho(x,\nu)$$

- Dual Subgradient ascent:

$$\nu_{t+1} = \nu_t + \epsilon(Ax_{t+1} - b)$$

*Natural step size*

$$x_{t+1} = \arg\max_{x} \mathcal{L}_\rho(x,\nu_t)$$

# Augmented Lagrangians

- Augmented Lagrangian:

$$\mathcal{L}_\rho(x,\nu) = f(x) + \nu^\top(Ax-b) + \frac{\rho}{2}\|Ax-b\|^2$$

- Dual problem:

$$\max_\nu \min_x \mathcal{L}_\rho(x,\nu)$$

- Dual Subgradient ascent:

*Natural step size*

$$x_{t+1} = \arg\max_x \mathcal{L}_\rho(x,\nu_t)$$

# Augmented Lagrangians

- Augmented Lagrangian:

$$\mathcal{L}_\rho(x,\nu) = f(x) + \nu^\top (Ax - b) + \frac{\rho}{2}\|Ax - b\|^2$$

- Dual problem:

$$\max_\nu \min_x \mathcal{L}_\rho(x,\nu)$$

- Dual Subgradient ascent:

$$\nu_{t+1} = \nu_t + \rho(Ax_{t+1} - b)$$

*Natural step size*

$$x_{t+1} = \arg\max_x \mathcal{L}_\rho(x,\nu_t)$$

# Augmented Lagrangians

- Augmented Lagrangian:

$$\mathcal{L}_\rho(x,\nu) = f(x) + \nu^\top (Ax - b) + \frac{\rho}{2}\|Ax - b\|^2$$

- Dual problem:

$$\max_\nu \min_x \mathcal{L}_\rho(x,\nu)$$

- Dual Subgradient ascent:

$$\nu_{t+1} = \nu_t + \rho(Ax_{t+1} - b)$$

*Natural step size*

$$x_{t+1} = \underset{\uparrow}{\arg}\max_x \mathcal{L}_\rho(x,\nu_t)$$

Costly

# Decomposable Objectives

- Optimization problem

$$\text{minimize } f(x) + g(z) \underbrace{\;\;\;}_{+\frac{\rho}{2}\|Ax - z\|^2} \text{ s.t. } Ax = z$$

Quadratic smoothing

- Augmented Lagrangian

$$\mathcal{L}_\rho(x, z, \nu) = f(x) + g(z) + \nu^\top (Ax - z) + \frac{\rho}{2}\|Ax - z\|^2$$

- ADMM updates

$$\begin{aligned} \nu^{t+1} &= \nu^t + \rho(Ax^{t+1} - z^{t+1}) \\ x^{t+1}, z^{t+1} &= \operatorname*{arg\,min}_{x,z} \mathcal{L}_\rho(x, z, \nu^t) \end{aligned}$$

[Gabay & Mercier 1976; Glowinski & Marroco 1975; Boyd et al. 2011]

# Decomposable Objectives

- Optimization problem

$$\text{minimize } f(x) + g(z) \underbrace{\phantom{x}}_{+\frac{\rho}{2}\|Ax - z\|^2} \text{ s.t. } Ax = z$$

Quadratic smoothing

- Augmented Lagrangian

$$\mathcal{L}_\rho(x, z, \nu) = f(x) + g(z) + \nu^\top (Ax - z) + \frac{\rho}{2}\|Ax - z\|^2$$

- ADMM updates

$$\begin{aligned} \nu^{t+1} &= \nu^t + \rho(Ax^{t+1} - z^{t+1}) \\ x^{t+1}, z^{t+1} &= \operatorname*{arg\,min}_{x,z} \mathcal{L}_\rho(x, z, \nu^t) \end{aligned}$$

*Don't want to optimize jointly!*

[Gabay & Mercier 1976; Glowinski & Marroco 1975; Boyd et al. 2011]

# Decomposable Objectives

- Optimization problem

$$\text{minimize } f(x) + g(z) \underbrace{\qquad}_{+\frac{\rho}{2}\|Ax - z\|^2} \text{s.t. } Ax = z$$

Quadratic smoothing

- Augmented Lagrangian

$$\mathcal{L}_\rho(x, z, \nu) = f(x) + g(z) + \nu^\top (Ax - z) + \frac{\rho}{2}\|Ax - z\|^2$$

- ADMM updates

$$\begin{aligned} \nu^{t+1} &= \nu^t + \rho(Ax^{t+1} - z^{t+1}) \\ x^{t+1} &= \operatorname*{arg\,min}_x \mathcal{L}_\rho(x, z^t, \nu^t) \\ z^{t+1} &= \operatorname*{arg\,min}_z \mathcal{L}_\rho(x^{t+1}, z, \nu^t) \end{aligned}$$

[Gabay & Mercier 1976; Glowinski & Marroco 1975; Boyd et al. 2011]

# ADMM for MAP-LP

- Martins et al. [2011] apply ADMM to the *primal* MAP-LP

$$\max_{\mu \in L(G)} \mu \cdot \theta$$

  - Does not handle non-pairwise non-binary models (only through binarization)

- Our approach: apply ADMM to the *dual* MAP-LP

  - Handles non-pairwise non-binary models directly

# The Augmented Dual LP Algorithm

$$\min_{\delta} \sum_i \max_{x_i} \left( \theta_i(x_i) + \sum_{c:i \in c} \delta_{ci}(x_i) \right) + \sum_c \max_{x_c} \left( \theta_c(x_c) - \sum_{i:i \in c} \delta_{ci}(x_i) \right)$$

- Can now apply ADMM!
- Properties:
  - Efficient updates
  - Convergence guarantee
    - O(1/$\varepsilon^2$) time in worst case, but fast in practice

# The Augmented Dual LP Algorithm

$$\min_{\delta} \sum_i \max_{x_i} \left( \theta_i(x_i) + \sum_{c:i\in c} \delta_{ci}(x_i) \right) + \sum_c \max_{x_c} \left( \theta_c(x_c) - \quad \lambda_c(x_c) \quad \right)$$

- Can now apply ADMM!
- Properties:
  - Efficient updates
  - Convergence guarantee
    - O(1/$\varepsilon^2$) time in worst case, but fast in practice

# The Augmented Dual LP Algorithm

$$\min_{\delta} \sum_i \max_{x_i} \left( \theta_i(x_i) + \sum_{c:i\in c} \delta_{ci}(x_i) \right) + \sum_c \max_{x_c} \left( \theta_c(x_c) - \quad \lambda_c(x_c) \quad \right)$$

$$\text{s.t. } \lambda_c(x_c) = \sum_{i:i\in c} \delta_{ci}(x_i) \quad \forall c, x_c$$

- Can now apply ADMM!
- Properties:
    - Efficient updates
    - Convergence guarantee
        - O(1/$\varepsilon^2$) time in worst case, but fast in practice

# The Augmented Dual LP Algorithm

$$\min_{\delta} \boxed{\sum_i \max_{x_i}\left(\theta_i(x_i)+\sum_{c:i\in c}\delta_{ci}(x_i)\right)+\sum_c \max_{x_c}\left(\theta_c(x_c)-\lambda_c(x_c)\right)}$$

$$0+g(\delta,\lambda)$$

$$\left.\begin{aligned}\text{s.t. } \lambda_c(x_c) &= \sum_{i:i\in c}\bar{\delta}_{ci}(x_i) \quad \forall c, x_c\\ \text{s.t. } \delta_{ci}(x_i) &= \bar{\delta}_{ci}(x_i) \quad \forall c, i, x_i\end{aligned}\right\} A\bar{\delta}=\begin{pmatrix}\delta\\ \lambda\end{pmatrix}$$

$$\min f(x)+g(z) \quad \text{s.t.} \quad Ax=z$$

$$x=\bar{\delta} \quad , \quad z=\begin{bmatrix}\delta\\ \lambda\end{bmatrix}$$

- Can now apply ADMM!
- Properties:
  - Efficient updates
  - Convergence guarantee
    - O(1/$\varepsilon^2$) time in worst case, but fast in practice

# Updates

# Updates

- Example: the λ update

$$\min_{\lambda_c} \max_{x_c} \left(\theta_c(x_c) - \lambda_c(x_c)\right) + \sum_{x_c} \mu_c(x_c)\lambda_c(x_c) + \frac{\rho}{2} \sum_{x_c} \left( \lambda_c(x_c) - \sum_{i:i\in c} \bar{\delta}_{ci}(x_i) \right)^2$$

# Updates

- Example: the λ update

$$\min_{\lambda_c} \max_{x_c} \left(\theta_c(x_c) - \lambda_c(x_c)\right) + \sum_{x_c} \mu_c(x_c)\lambda_c(x_c) + \frac{\rho}{2} \sum_{x_c} \left( \lambda_c(x_c) - \sum_{i:i \in c} \bar{\delta}_{ci}(x_i) \right)^2$$

- Requires solving a QP

$$\min_x \frac{1}{2}\|x\|^2 - v^\top x + \frac{1}{\rho} \max_i(x_i)$$

# Updates

- Example: the λ update

$$\min_{\lambda_c} \max_{x_c} \left(\theta_c(x_c) - \lambda_c(x_c)\right) + \sum_{x_c} \mu_c(x_c)\lambda_c(x_c) + \frac{\rho}{2} \sum_{x_c} \left( \lambda_c(x_c) - \sum_{i:i \in c} \bar{\delta}_{ci}(x_i) \right)^2$$

- Requires solving a QP

$$\min_x \frac{1}{2} \|x\|^2 - v^\top x + \frac{1}{\rho} \max_i (x_i)$$

- Solution: sort v then reduce a total of 1/ρ from the largest elements

# Updates

- Example: the λ update

$$\min_{\lambda_c} \max_{x_c} \left(\theta_c(x_c) - \lambda_c(x_c)\right) + \sum_{x_c} \mu_c(x_c)\lambda_c(x_c) + \frac{\rho}{2} \sum_{x_c} \left(\lambda_c(x_c) - \sum_{i:i \in c} \bar{\delta}_{ci}(x_i)\right)^2$$

- Requires solving a QP

$$\min_x \frac{1}{2}\|x\|^2 - v^\top x + \frac{1}{\rho} \max_i(x_i)$$

- Solution: sort v then reduce a total of 1/ρ from the largest elements

- Can be done efficiently via partitioning (O(n) in expectation)

# Updates

- Example: the λ update

$$\min_{\lambda_c} \max_{x_c} \left(\theta_c(x_c) - \lambda_c(x_c)\right) + \sum_{x_c} \mu_c(x_c)\lambda_c(x_c) + \frac{\rho}{2} \sum_{x_c} \left( \lambda_c(x_c) - \sum_{i:i \in c} \bar{\delta}_{ci}(x_i) \right)^2$$

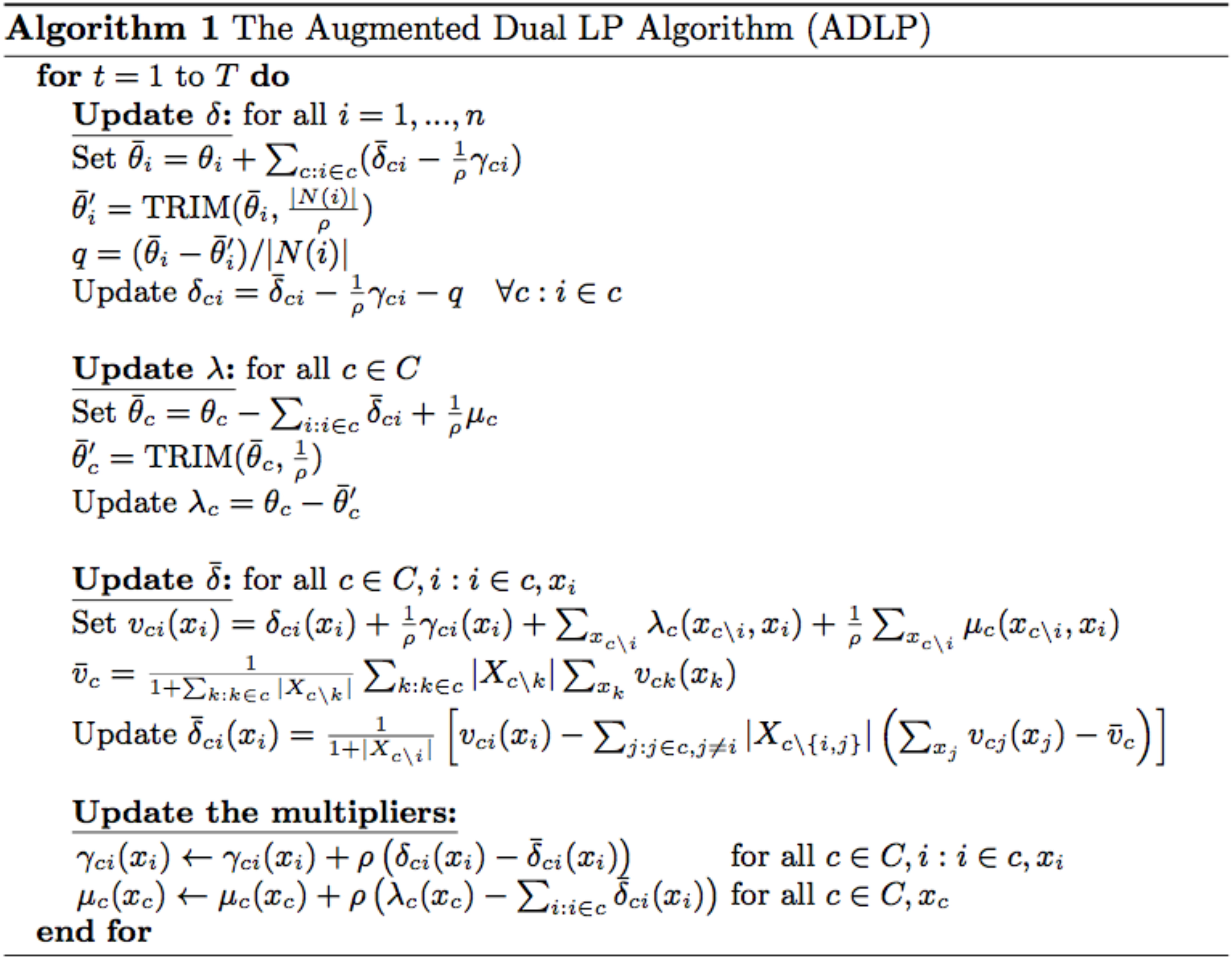
- Requires solving a QP

$$\min_x \frac{1}{2}\|x\|^2 - v^\top x + \frac{1}{\rho} \max_i(x_i)$$

- Solution: sort v then reduce a total of 1/ρ from the largest elements

- Can be done efficiently via partitioning (O(n) in expectation)

# ADLP Algorithm



**Algorithm 1** The Augmented Dual LP Algorithm (ADLP)

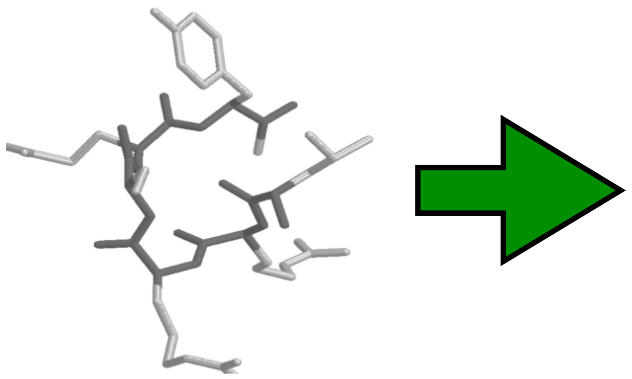
```
for $t = 1$ to $T$ do
    Update $\delta$: for all $i = 1, ..., n$
    Set $\bar{\theta}_i = \theta_i + \sum_{c:i \in c}(\bar{\delta}_{ci} - \frac{1}{\rho}\gamma_{ci})$
    $\bar{\theta}'_i = \text{TRIM}(\bar{\theta}_i, \frac{|N(i)|}{\rho})$
    $q = (\bar{\theta}_i - \bar{\theta}'_i)/|N(i)|$
    Update $\delta_{ci} = \bar{\delta}_{ci} - \frac{1}{\rho}\gamma_{ci} - q \quad \forall c : i \in c$

    Update $\lambda$: for all $c \in C$
    Set $\bar{\theta}_c = \theta_c - \sum_{i:i \in c} \bar{\delta}_{ci} + \frac{1}{\rho}\mu_c$
    $\bar{\theta}'_c = \text{TRIM}(\bar{\theta}_c, \frac{1}{\rho})$
    Update $\lambda_c = \theta_c - \bar{\theta}'_c$

    Update $\bar{\delta}$: for all $c \in C, i : i \in c, x_i$
    Set $v_{ci}(x_i) = \delta_{ci}(x_i) + \frac{1}{\rho}\gamma_{ci}(x_i) + \sum_{x_{c \setminus i}} \lambda_c(x_{c \setminus i}, x_i) + \frac{1}{\rho} \sum_{x_{c \setminus i}} \mu_c(x_{c \setminus i}, x_i)$
    $\bar{v}_c = \frac{1}{1 + \sum_{k:k \in c} |X_{c \setminus k}|} \sum_{k:k \in c} |X_{c \setminus k}| \sum_{x_k} v_{ck}(x_k)$
    Update $\bar{\delta}_{ci}(x_i) = \frac{1}{1 + |X_{c \setminus i}|} \left[ v_{ci}(x_i) - \sum_{j:j \in c, j \neq i} |X_{c \setminus \{i,j\}}| \left( \sum_{x_j} v_{cj}(x_j) - \bar{v}_c \right) \right]$

    Update the multipliers:
    $\gamma_{ci}(x_i) \leftarrow \gamma_{ci}(x_i) + \rho\left(\delta_{ci}(x_i) - \bar{\delta}_{ci}(x_i)\right)$    for all $c \in C, i : i \in c, x_i$
    $\mu_c(x_c) \leftarrow \mu_c(x_c) + \rho\left(\lambda_c(x_c) - \sum_{i:i \in c} \bar{\delta}_{ci}(x_i)\right)$ for all $c \in C, x_c$
end for
```

# Experiments

- Compared algorithms
  - Block coordinate descent (MPLP) [Globerson & Jaakkola 2007]
  - Accelerated Dual Decomposition (ADD) [Jojic et al. 2010]
  - Our Augmented Dual LP algorithm (ADLP)

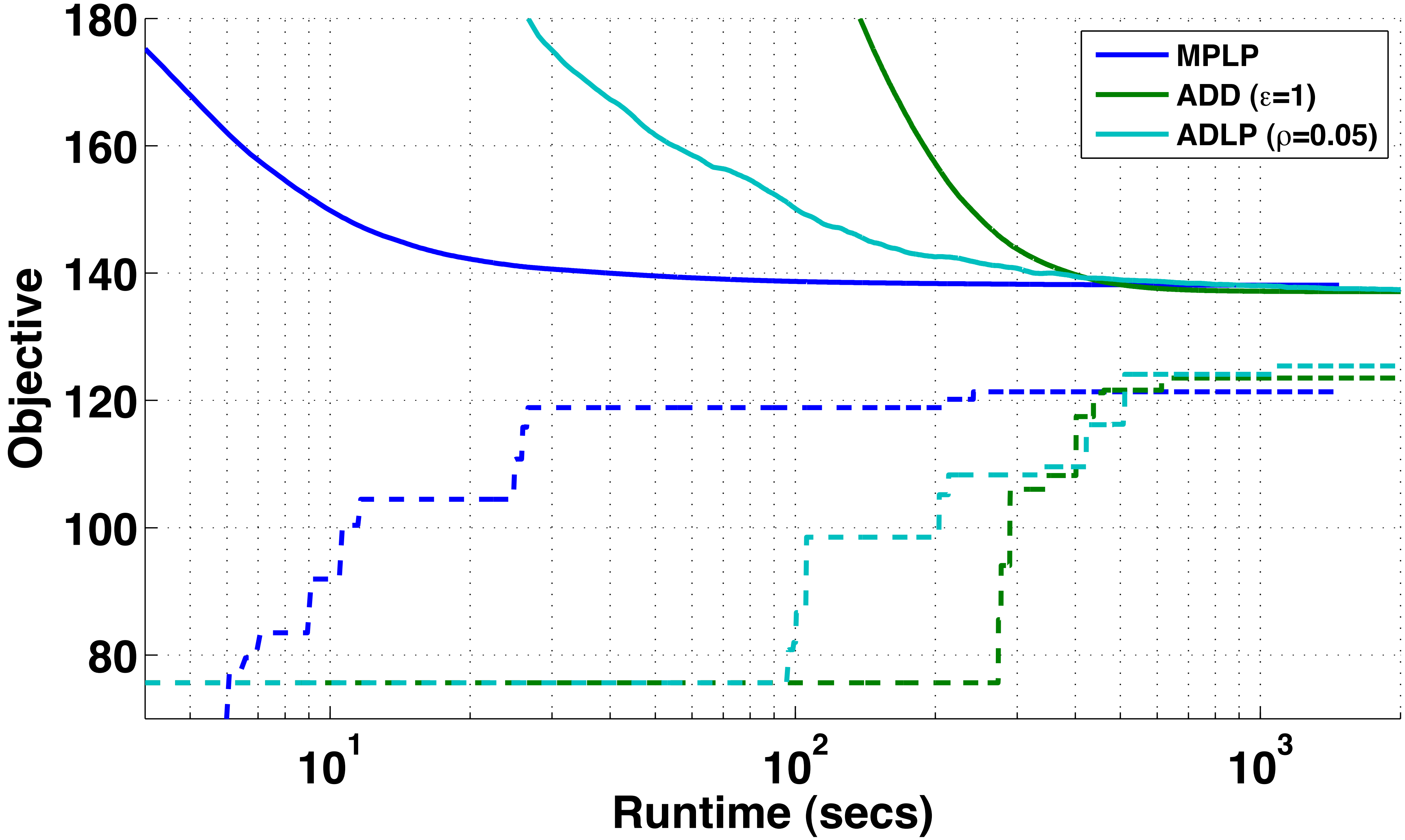
# Experiments

- Protein design problem



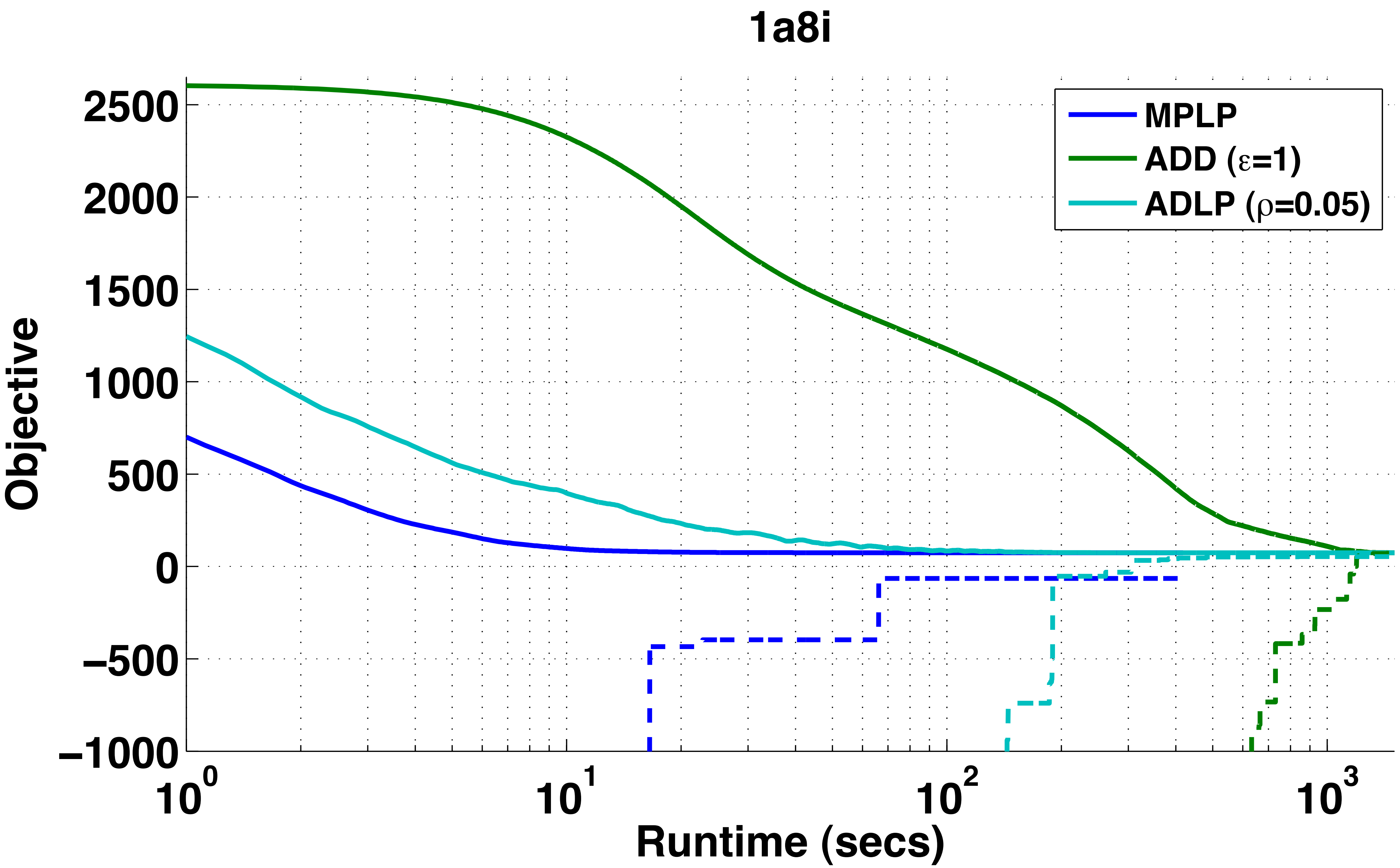
KDLMHNKCYHFFM

1jo8



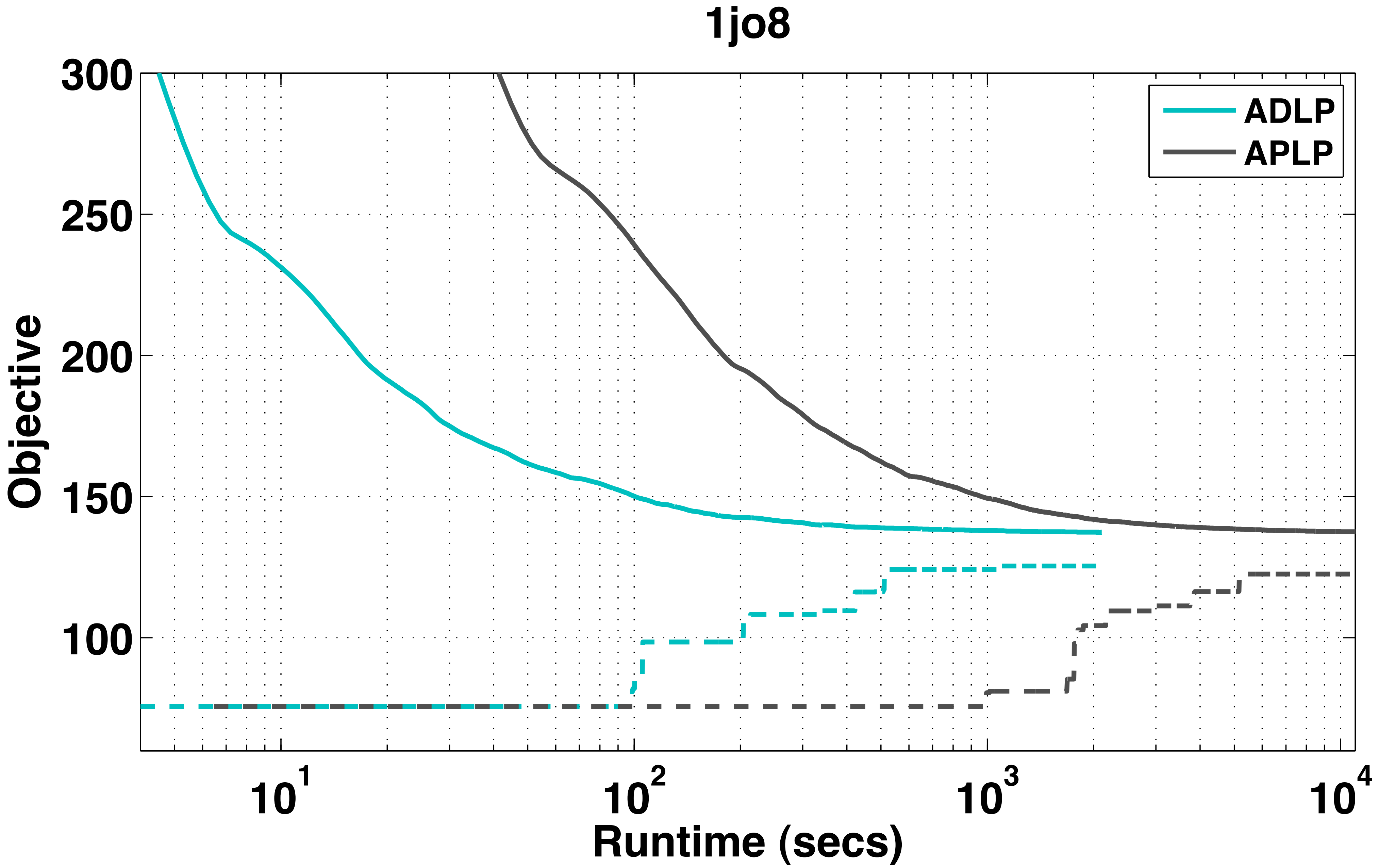
# Experiments

- Side chain prediction (inverse problem)



# Experiments

- Protein design: primal vs. dual ADMM
  (our primal ADMM variant handles high-order non-binary factors directly)



# Summary

- A novel algorithm for MAP-LP based on applying ADMM to the dual LP
- Efficient closed-form updates
- Provably converges to the approximate optimum
- Empirically effective

# Thank you!

# The Augmented Lagrangian

$$\mathcal{L}_\rho(\delta, \lambda, \bar{\delta}, \gamma, \mu) =$$

$$\sum_i \max_{x_i} \left( \theta_i(x_i) + \sum_{c:i\in c} \delta_{ci}(x_i) \right) + \sum_c \max_{x_c} \left( \theta_c(x_c) - \lambda_c(x_c) \right)$$

$$+ \sum_c \sum_{x_c} \mu_c(x_c) \left( \lambda_c(x_c) - \sum_{i:i\in c} \bar{\delta}_{ci}(x_i) \right) + \frac{\rho}{2} \sum_c \sum_{x_c} \left( \lambda_c(x_c) - \sum_{i:i\in c} \bar{\delta}_{ci}(x_i) \right)^2$$

$$+ \sum_c \sum_{i:i\in c} \sum_{x_i} \gamma_{ci}(x_i) \left( \delta_{ci}(x_i) - \bar{\delta}_{ci}(x_i) \right) + \frac{\rho}{2} \sum_c \sum_{i:i\in c} \sum_{x_i} \left( \delta_{ci}(x_i) - \bar{\delta}_{ci}(x_i) \right)$$

# Updates

- The λ update

$$\min_{\lambda_c} \max_{x_c} \left(\theta_c(x_c) - \lambda_c(x_c)\right) + \sum_{x_c} \mu_c(x_c)\lambda_c(x_c) + \frac{\rho}{2} \sum_{x_c} \left( \lambda_c(x_c) - \sum_{i:i \in c} \bar{\delta}_{ci}(x_i) \right)^2$$

- The δ update

$$\min_{\delta_i} \max_{x_i} \left( \theta_i(x_i) + \sum_{c:i \in c} \delta_{ci}(x_i) \right) + \sum_{c:i \in c} \sum_{x_i} \gamma_{ci}(x_i)\delta_{ci}(x_i) + \frac{\rho}{2} \sum_{c:i \in c} \sum_{x_i} \left( \delta_{ci}(x_i) - \bar{\delta}_{ci}(x_i) \right)^2$$

- The δ̄ update

$$- \sum_{i:i \in c} \sum_{x_i} \gamma_{ci}(x_i)\bar{\delta}_{ci}(x_i) + \frac{\rho}{2} \sum_{i:i \in c} \sum_{x_i} \left( \delta_{ci}(x_i) - \bar{\delta}_{ci}(x_i) \right)^2$$

$$- \sum_{x_c} \mu_c(x_c) \sum_{i:i \in c} \bar{\delta}_{ci}(x_i) + \frac{\rho}{2} \sum_{x_c} \left( \lambda_c(x_c) - \sum_{i:i \in c} \bar{\delta}_{ci}(x_i) \right)^2$$