# An alternating direction method of multipliers for solving user equilibrium problem

Zhiyuan Liu, Xinyuan Chen, Jintao Hu, Shuaian Wang, Kai Zhang, Honggang Zhang. (2023).

*European Journal of Operational Research*, 310(3), 1072-1084

理論談話会#5 (2024/05/13)
M1 Furuhashi Fumihito

# Summary

## Main Challenge

Applying **parallel computing** approach to solve the **UE problem**

## Contribution

1. **Origin-base** formulation
2. The algorithm grouping links into **Blocks**

## Validation

- 4 numerical experiments
- The performance of ADMM is **superior** to some existing algorithms

*"This study presented an initial step on the aspect of using ADMM for parallel computing of UE."*

$$\min Z_2 = \sum_{a\in A}\int_0^{\sum_{o\in O} v_a^o} t_a(w)dw \quad (7)$$

$$\text{s.t.} \sum_{\substack{a\in A \\ i(a)=n}} v_a^o - \sum_{\substack{a\in A \\ h(a)=n}} v_a^o = g_n^o, \qquad \forall o \in O, n \in N \quad (8)$$

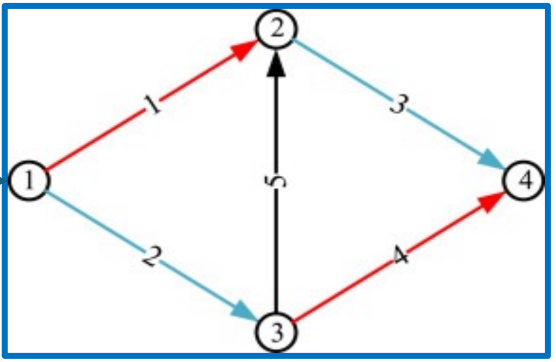
$$v_a \geq 0, \forall o \in O, a \in A \quad (9)$$

$$g_n^o = \begin{cases} \sum_{od\in W^o} q^{od}, n = o \\ -q^{od}, n = d \\ 0, \text{otherwise} \end{cases}, \forall o \in O, n \in N \quad (10)$$

$$L_0(\mathbf{v}, \boldsymbol{\lambda}) = \sum_{a\in A}\int_0^{\sum_{0\in O} v_a^o} t_a(w)dw + \sum_{o\in O}\sum_{n\in N} \lambda_n^o H_n^o(v) \quad (12)$$

$$\max_{\boldsymbol{\lambda}} \inf_{\mathbf{v}\geq 0} L_0(\mathbf{v}, \boldsymbol{\lambda}) \quad (13)$$

$$\mathbf{v}^* = \arg\min_{\mathbf{v}\geq 0} L_0(\mathbf{v}, \boldsymbol{\lambda}^*) \quad (14)$$



# Novelty, Utility, Reliability

## Novelty

1. Applying **parallel** computing approach to solve the **UE** problem by adopting **ADMM**
2. Proposing a novel algorithm grouping network links into **block**

## Utility

1. On a big size network, the ADMM algorithm can solve UE problems tremendously **faster**
2. The ADMM algorithms can be more **accelerate** and **extend** its use to other types of traffic assignment problems

## Reliability

4 types of numerical tests

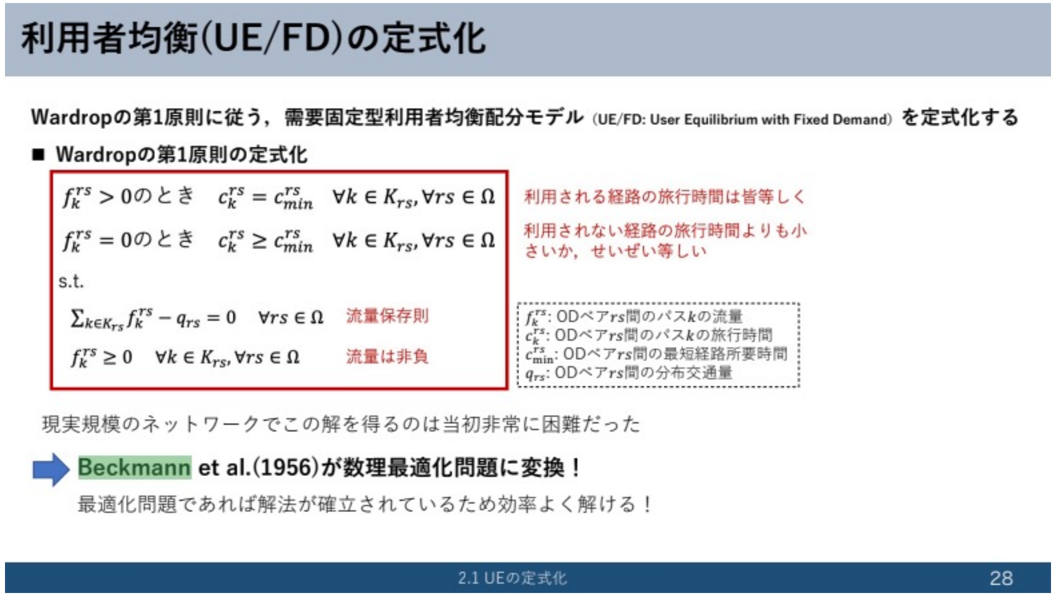
# Contents

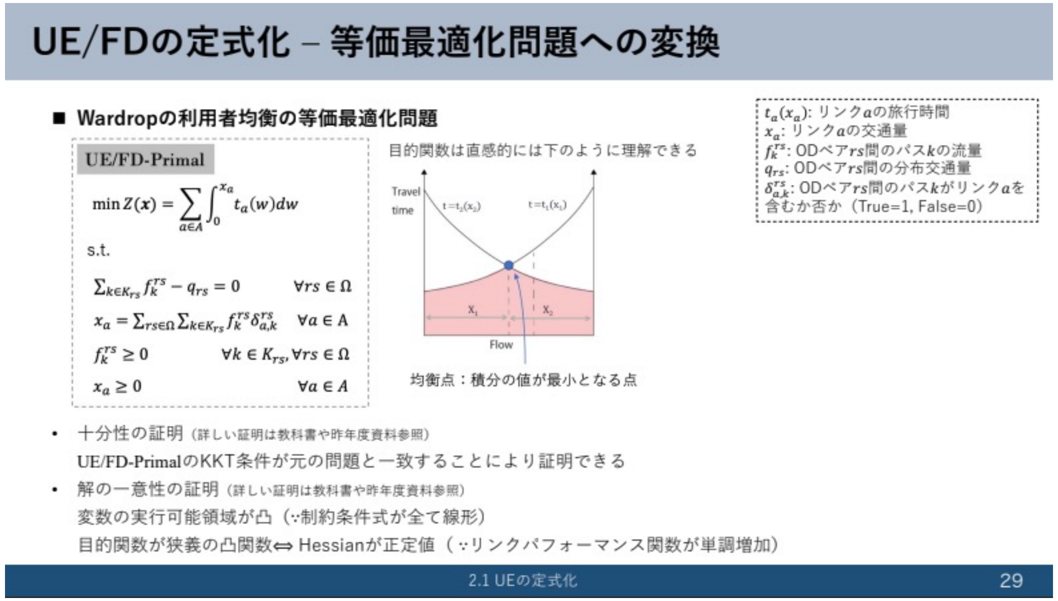


*Key Words*: Traffic assignment, User equilibrium, Parallel computing, Alternating direction method of multipliers, Edge-coloring problem

# Contents

# 1.1 Literature review



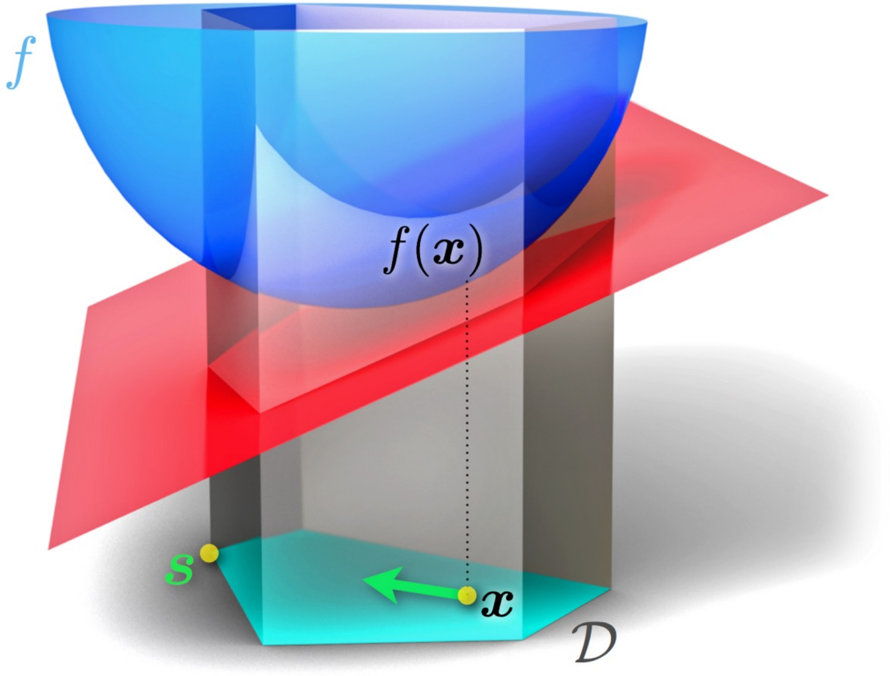




増田さん Start-upゼミ#2資料 pp. 28-29

# Solution

***Algorithms***

- Link-base: Frank-Wolfe algorithm, Gauss-Seidel iteration method
- Path-base
- Origin-base (**This paper**)



# Contents

# Definition

| | | |
|---|---|---|
| Transport Network | $G$ | $G = (N, A)$ |
| Set of nodes | $N$ | $n \in N$ |
| Set of links | $A$ | $a \in A$ |
| Set of OD pairs | $W$ | $(o, d) \in W$ |
| Travel demand for $(o, d) \in W$ | $q^{od}$ | |
| Set of path between OD pair $(o, d)$ | $K^{od}$ | $k \in K^{od}$ |
| Flow on path $k \in K^{od}$ | $f_k^{od}$ | |
| Flow on link $a \in A$ | $v_a$ | |
| Link-path incidence relationship | $\delta_{a,k}^{od}$ | $\delta_{a,k}^{od} = \begin{cases} 1, path\ k \in K^{od}\ use\ link\ a \\ 0, otherwise \end{cases}$ |
| Travel time function of link $a \in A$ | $t_a(v_a)$ | Strictly increasing and continuously differentiable |
| Lagrange multipliers | $\varphi^{od}$ | Shortest travel time between OD pair $(o, d)$ |
| Travel cost of the path $k$ betweenOD pair $(o, d)$ | $c_k^{od}$ | |

# Widely used mathematical formulation of UE [MP-UE]

$$\min Z_1 = \sum_{a \in A} \int_0^{v_a} t_a(w)dw \qquad (1)$$

$$\text{s.t.} \sum_{k \in K^{od}} f_k^{od} = q^{od}, \qquad \forall od \in W \qquad (2)$$

$$f_k^{od} \geq 0, \forall k \in K^{od}, od \in W \qquad (3)$$

$$v_a = \sum_{od \in W} \sum_{k \in K^{od}} f_k^{od} \delta_{a,k}^{od}, \forall a \in A. \qquad (4)$$

# Widely used mathematical formulation of UE [MP-UE]

$$\min Z_1 = \sum_{a \in A} \int_0^{v_a} t_a(w) dw \quad (1)$$

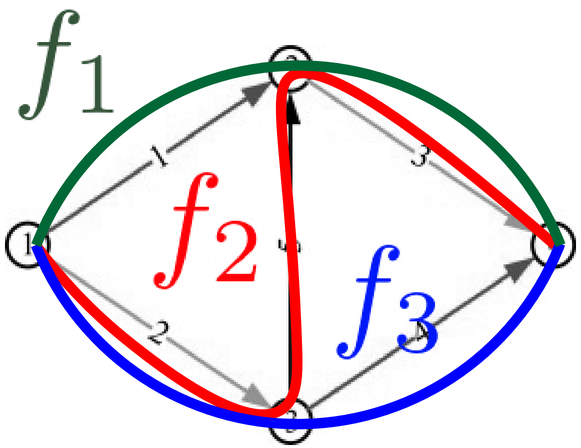
Flow conservation

$$\text{s.t.} \sum_{k \in K^{od}} f_k^{od} = q^{od}, \qquad \forall od \in W \quad (2)$$

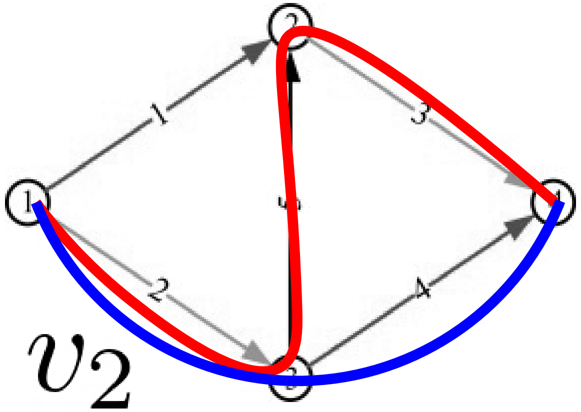
$$f_k^{od} \geq 0, \forall k \in K^{od}, od \in W \quad (3)$$

Travel cost of link $a$

$$v_a = \sum_{od \in W} \sum_{k \in K^{od}} f_k^{od} \delta_{a,k}^{od}, \forall a \in A. \quad (4)$$




Flow on path




Flow on link

# Widely used mathematical formulation of UE [MP-UE]

$$\min Z_1 = \sum_{a \in A} \int_0^{v_a} t_a(w)dw \quad (1)$$

Flow conservation

$$s.t. \sum_{k \in K^{od}} f_k^{od} = q^{od}, \quad \forall od \in W \quad (2)$$

$$f_k^{od} \geq 0, \forall k \in K^{od}, od \in W \quad (3)$$

Travel cost of link a

$$v_a = \sum_{od \in W} \sum_{k \in K^{od}} f_k^{od} \delta_{a,k}^{od}, \forall a \in A. \quad (4)$$

## PROBLEMS

**Separable** w.r.t link flows.

$\int_0^{v_{a_i}} t_{a_i}(w)dw$ and $\int_0^{v_{a_j}} t_{a_j}(w)dw$ is independent each other (= **parallelizable**!)

**NOT separable** w.r.t link or path flows

And the number of paths $N(K)$ is tremendously large in an urban transport network...

**Lagrangian duality theoly**

(2) and (4) could be converted into the objective function. But it is difficult to be directly handled by Lagrangian duality.

## SOLUTION

**Origin-base Model** (Beckmann et al. 1956)

# Origin-based model [OB-UE]

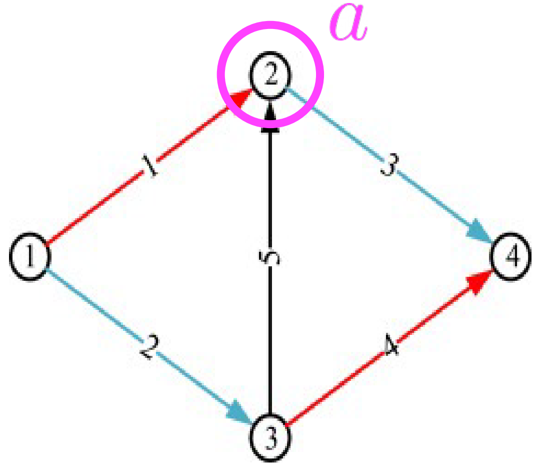
$$\min Z_2 = \sum_{a \in A} \int_0^{\sum_{o \in O} v_a^o} t_a(w)dw \tag{7}$$

$$\text{s.t.} \sum_{\substack{a \in A \\ i(a)=n}} v_a^o - \sum_{\substack{a \in A \\ h(a)=n}} v_a^o = g_n^o, \quad \forall o \in O, n \in N \tag{8}$$

$$v_a \geq 0, \forall o \in O, a \in A \tag{9}$$

$$g_n^o = \begin{cases} \sum_{od \in W^o} q^{od}, n = o \\ -q^{od}, n = d \\ 0, \text{otherwise} \end{cases}, \forall o \in O, n \in N \tag{10}$$

| | | |
|---|---|---|
| Traffic flow on link $a$ originating from $o$ | $v_a^o$ | $v_a = \sum_{o \in O} v_a^o$ |
| Tail node of link $a$ | $i(a)$ | |
| Head node of link $a$ | $h(a)$ | |




$A' = \{1\}$ when $i(a) = 2$
$A' = \{3\}$ when $h(a) = 2$

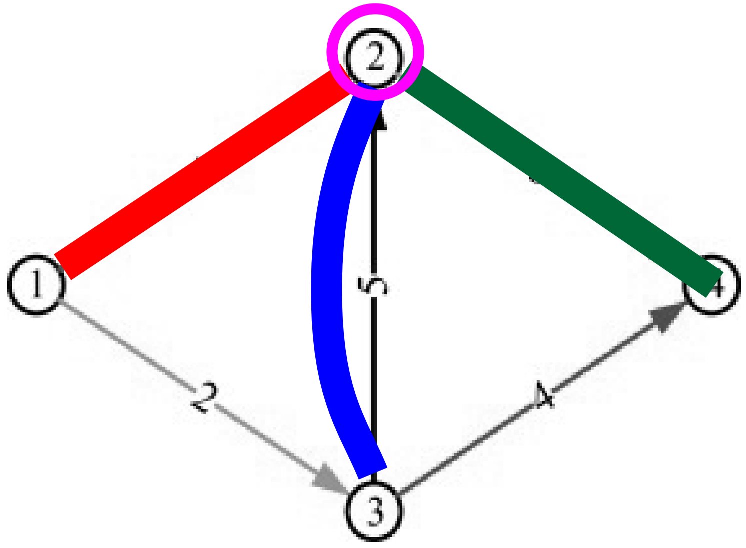
# Origin-based model [OB-UE]

$$\min Z_2 = \sum_{a\in A} \int_0^{\sum_{o\in O} v_a^o} t_a(w)dw \quad (7)$$

$$\text{s.t.} \sum_{\substack{a\in A \\ i(a)=n}} v_a^o - \sum_{\substack{a\in A \\ h(a)=n}} v_a^o = g_n^o, \qquad \forall o \in O, n \in N \quad (8)$$

$$v_a \geq 0, \forall o \in O, a \in A \quad (9)$$

$$g_n^o = \begin{cases} \sum_{od\in W^o} q^{od}, n = o \\ -q^{od}, n = d \\ 0, \text{otherwise} \end{cases}, \forall o \in O, n \in N \quad (10)$$




$$(v_1^1 + v_5^1) - (v_3^1) = g_3^1$$

# Origin-based model [OB-UE]

$$\min Z_2 = \sum_{a \in A} \int_0^{\sum_{o \in O} v_a^o} t_a(w)dw \qquad (7)$$

$$\text{s.t.} \sum_{\substack{a \in A \\ i(a)=n}} v_a^o - \sum_{\substack{a \in A \\ h(a)=n}} v_a^o = g_n^o, \qquad \forall o \in O, n \in N \qquad (8)$$

$$v_a \geq 0, \forall o \in O, a \in A \qquad (9)$$

$$g_n^o = \begin{cases} \sum_{od \in W^o} q^{od}, n = o \\ -q^{od}, n = d \\ 0, \text{otherwise} \end{cases}, \forall o \in O, n \in N \qquad (10)$$

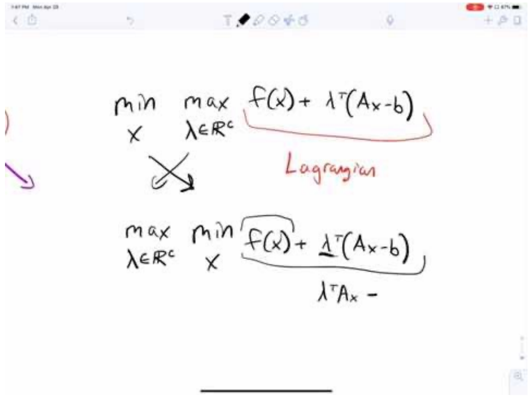
## HOW TO SOLVE?

***Conventional Approach*** (Nie, 2010)

- Decompose it into a series of restricted origin-based subproblems
- Solve them sequentially by the Gauss-Seidel iteration sheme

***This paper***

- Employ **Lagrange relaxation**



# Lagrange Relaxation

https://www.bunkyo.ac.jp/~nemoto/lecture/opt-model/2008/duality1-2007.pdf

[P]

$$\max 6x_1 + 4x_2$$
$$\text{s.t.}\ 2x_1 + x_2 \le 70$$
$$3x_1 + 4x_2 \le 180$$
$$x_1, x_2 \ge 0$$

# Lagrange Relaxation

https://www.bunkyo.ac.jp/~nemoto/lecture/opt-model/2008/duality1-2007.pdf

[P]

$$\max 6x_1 + 4x_2$$
$$\text{s.t. } 2x_1 + x_2 \leq 70$$
$$3x_1 + 4x_2 \leq 180$$
$$x_1, x_2 \geq 0$$

$\leq$

[P1]

Negative Negative

$$\max 6x_1 + 4x_2 + y_1\left(\mathbf{70 - (2x_1 + x_2)}\right) + y_2(\mathbf{180 - (3x_1 + 4x_2)})$$
$$\text{s.t. } 2x_1 + x_2 \leq 70$$
$$3x_1 + 4x_2 \leq 180$$
$$x_1, x_2, y_1, y_2 \geq 0$$

# Lagrange Relaxation

https://www.bunkyo.ac.jp/~nemoto/lecture/opt-model/2008/duality1-2007.pdf

[P]

$$\max 6x_1 + 4x_2$$
$$\text{s.t. } 2x_1 + x_2 \leq 70$$
$$3x_1 + 4x_2 \leq 180$$
$$x_1, x_2 \geq 0$$

$\leq$

[P1]

$$\max 6x_1 + 4x_2 + y_1\big(70 - (2x_1 + x_2)\big) + y_2(180 - (3x_1 + 4x_2))$$
$$\text{s.t. } 2x_1 + x_2 \leq 70$$
$$3x_1 + 4x_2 \leq 180$$
$$x_1, x_2, y_1, y_2 \geq 0$$

$\leq$

[P2-1]

$$\max 6x_1 + 4x_2 + y_1\big(70 - (2x_1 + x_2)\big) + y_2(180 - (3x_1 + 4x_2))$$
$$\text{s.t. } \boldsymbol{x_1, x_2, y_1, y_2 \geq 0}$$

**Lagrange Relaxation**

# Lagrange Relaxation

https://www.bunkyo.ac.jp/~nemoto/lecture/opt-model/2008/duality1-2007.pdf

[P]

$$\max 6x_1 + 4x_2$$
$$\text{s.t. } 2x_1 + x_2 \leq 70$$
$$3x_1 + 4x_2 \leq 180$$
$$x_1, x_2 \geq 0$$

$\leq$

[P1]

$$\max 6x_1 + 4x_2 + y_1\big(70 - (2x_1 + x_2)\big) + y_2(180 - (3x_1 + 4x_2))$$
$$\text{s.t. } 2x_1 + x_2 \leq 70$$
$$3x_1 + 4x_2 \leq 180$$
$$x_1, x_2, y_1, y_2 \geq 0$$

$\leq$

[P2-1]

$$\max 6x_1 + 4x_2 + y_1\big(70 - (2x_1 + x_2)\big) + y_2(180 - (3x_1 + 4x_2))$$
$$\text{s.t. } x_1, x_2, y_1, y_2 \geq 0$$

$=$

[P2-2]

Negative — Negative

$$\max(6 - 2y_1 - 3y_2)x_1 + (4 - y_1 - 4y_2)x_2 + 70y_1 + 180y_2$$
$$\text{s.t. } x_1, x_2, y_1, y_2 \geq 0$$

# Lagrange Relaxation

https://www.bunkyo.ac.jp/~nemoto/lecture/opt-model/2008/duality1-2007.pdf

[P]

$$\max 6x_1 + 4x_2$$
$$\text{s.t. } 2x_1 + x_2 \leq 70$$
$$3x_1 + 4x_2 \leq 180$$
$$x_1, x_2 \geq 0$$

$\leq$

[P1]

$$\max 6x_1 + 4x_2 + y_1(70 - (2x_1 + x_2)) + y_2(180 - (3x_1 + 4x_2))$$

**Dual problem**

[D]

$$\min 70y_1 + 180y_2$$
$$\text{s.t. } 2y_1 + 3y_2 \geq 6$$
$$y_1 + 4y_2 \geq 4$$
$$y_1, y_2 \geq 0$$

$\leq$

[P2-1]

$$\max 6x_1 + 4x_2 \ldots - (3x_1 + 4x_2))$$
$$\text{s.t. } x_1, x_2, y_1, y_2 \geq 0$$

$=$

[P2-2]

$$\max (6 - 2y_1 - 3y_2)x_1 + (4 - y_1 - 4y_2)x_2 + 70y_1 + 180y_2$$
$$\text{s.t. } x_1, x_2, y_1, y_2 \geq 0$$

# Lagrangian Function [OB-UE]

$$\min Z_2 = \sum_{a \in A} \int_0^{\sum_{o \in O} v_a^o} t_a(w)dw \quad (7)$$

$$\text{s.t.} \sum_{\substack{a \in A \\ i(a)=n}} v_a^o - \sum_{\substack{a \in A \\ h(a)=n}} v_a^o = g_n^o, \qquad \forall o \in O, n \in N \quad (8)$$

$$v_a \geq 0, \forall o \in O, a \in A \quad (9)$$

$$g_n^o = \begin{cases} \sum_{od \in W^o} q^{od}, n = o \\ -q^{od}, n = d \\ 0, \text{otherwise} \end{cases}, \forall o \in O, n \in N \quad (10)$$

$$L_0(\mathbf{v}, \boldsymbol{\lambda}) = \sum_{a \in A} \int_0^{\sum_{0 \in O} v_a^o} t_a(w)dw + \sum_{o \in O} \sum_{n \in N} \lambda_n^o H_n^o(v) \quad (12)$$

$$\max_{\boldsymbol{\lambda}} \inf_{\mathbf{v} \geq 0} L_0(\mathbf{v}, \boldsymbol{\lambda}) \quad (13)$$

$$\mathbf{v}^* = \arg\min_{\mathbf{v} \geq 0} L_0(\mathbf{v}, \boldsymbol{\lambda}^*) \quad (14)$$

# Lagrangian Function [OB-UE]

$$\min Z_2 = \sum_{a \in A} \int_0^{\sum_{o \in O} v_a^o} t_a(w)dw \quad (7)$$

$$\text{s.t.} \sum_{\substack{a \in A \\ i(a)=n}} v_a^o - \sum_{\substack{a \in A \\ h(a)=n}} v_a^o = g_n^o, \qquad \forall o \in O, n \in N \quad (8)$$

Affine functions $v_a \geq 0, \forall o \in O, a \in A$ (9)

$$g_n^o = \begin{cases} \sum_{od \in W^o} q^{od}, n = o \\ -q^{od}, n = d \\ 0, \text{otherwise} \end{cases}, \forall o \in O, n \in N \quad (10)$$

$$L_0(\mathbf{v}, \boldsymbol{\lambda}) = \sum_{a \in A} \int_0^{\sum_{o \in O} v_a^o} t_a(w)dw + \sum_{o \in O} \sum_{n \in N} \lambda_n^o H_n^o(v) \quad (12)$$

**Dual problem** $\max_{\boldsymbol{\lambda}} \inf_{\mathbf{v} \geq 0} L_0(\mathbf{v}, \boldsymbol{\lambda})$ (13)

$$\mathbf{v}^* = \arg\min_{\mathbf{v} \geq 0} L_0(\mathbf{v}, \boldsymbol{\lambda}^*) \quad (14)$$

# Lagrangian Function [OB-UE]

$$\min Z_2 = \sum_{a \in A} \int_0^{\sum_{o \in O} v_a^o} t_a(w)dw \tag{7}$$

$$\text{s.t.} \sum_{\substack{a \in A \\ i(a)=n}} v_a^o - \sum_{\substack{a \in A \\ h(a)=n}} v_a^o = g_n^o, \quad \forall o \in O, n \in N \tag{8}$$

$$v_a \geq 0, \forall o \in O, a \in A \tag{9}$$

$$g_n^o = \begin{cases} \sum_{od \in W^o} q^{od}, n = o \\ -q^{od}, n = d \\ 0, \text{otherwise} \end{cases}, \forall o \in O, n \in N \tag{10}$$

$$L_0(\mathbf{v}, \boldsymbol{\lambda}) = \sum_{a \in A} \int_0^{\sum_{o \in O} v_a^o} t_a(w)dw + \sum_{o \in O} \sum_{n \in N} \lambda_n^o H_n^o(v) \tag{12}$$

$$\max_{\boldsymbol{\lambda}} \inf_{\mathbf{v} \geq 0} L_0(\mathbf{v}, \boldsymbol{\lambda}) \tag{13}$$

$$\mathbf{v}^* = \arg\min_{\mathbf{v} \geq 0} L_0(\mathbf{v}, \boldsymbol{\lambda}^*) \tag{14}$$

## PROBLEM

**NOT STRICT CONVEX** w.r.t $\mathbf{v}$

## HOW TO SOLVE?

***Conventional Approach*** (Nie, 2010)

- Gradient-based algorithms, e.g., the dual ascent method
- Requiring **strict convexity** of the primal model (Boyed et al., 2011)

***This paper***

- Adopt **Augumented Lagrangian**

# Lagrangian Function [OB-UE]

$$\min Z_2 = \sum_{a \in A} \int_0^{\sum_{o \in O} v_a^o} t_a(w)\,dw \qquad (7)$$

s. t.

It can be easily proven that the objective function (7) is convex but not strict convex w.r.t. the origin-based link flows $\mathbf{v}$. Considering that the constraints (9) are affine functions, the strong duality holds (Boyd & Vandenberghe, 2004). Thus, the optimal value of the primal equals the optimal value of the dual problem. We can obtain a primal optimal point $\mathbf{v}^*$ from a dual optimal point $\boldsymbol{\lambda}^*$ (Boyd et al., 2011):

$$\min_{\mathbf{v} \geq 0} L_0\left(\mathbf{v}, \boldsymbol{\lambda}^*\right). \qquad (14)$$

$L_0(\mathbf{v}, \boldsymbol{\lambda})$

$$\max_{\boldsymbol{\lambda}} \inf_{\mathbf{v} \geq 0} L_0(\mathbf{v}, \boldsymbol{\lambda}) \qquad (13)$$

$$\mathbf{v}^* = \arg\min_{\mathbf{v} \geq 0} L_0(\mathbf{v}, \boldsymbol{\lambda}^*) \qquad (14)$$

## PROBLEM

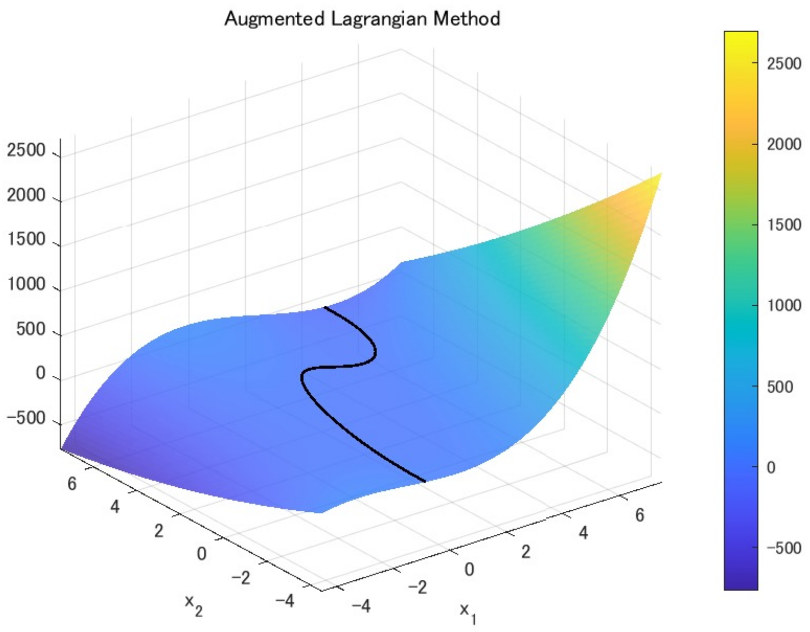
- Adopt **Augumented Lagrangian**

# Augumented Lagransian

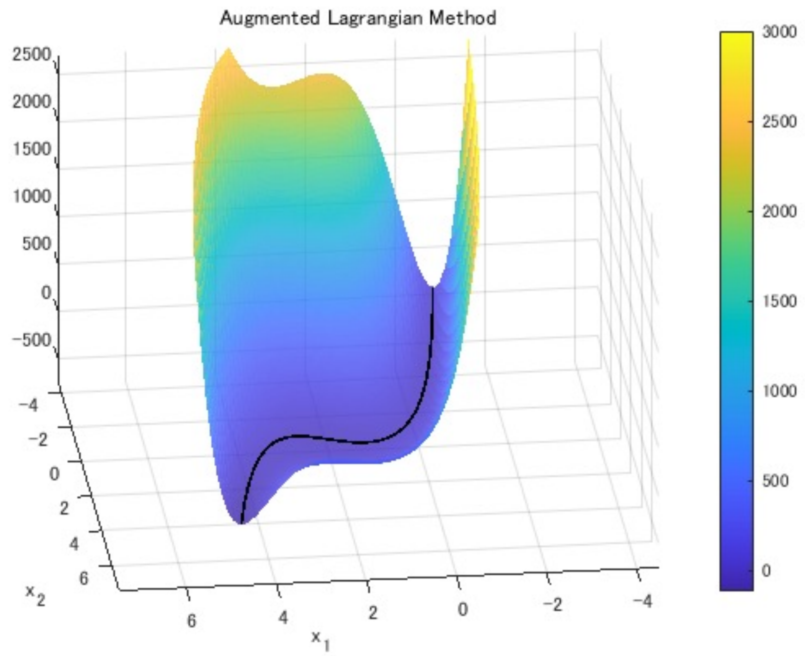
https://qiita.com/hibs_MATLAB_Amb/items/1b92be3f1c1de446d64d

Penalty parameter

$$L_\rho(\mathbf{v},\boldsymbol{\lambda}) = \sum_{a\in A}\int_0^{\sum_{o\in O} v_a^o} t_a(w)dw + \sum_{o\in O}\sum_{n\in N}\lambda_n^o H_n^o(v) + \frac{\rho}{2}\sum_{o\in O}\sum_{n\in N}\left(H_n^o(v)\right)^2 \tag{15}$$




$$+\frac{\rho}{2}\sum_{o\in O}\sum_{n\in N}\left(H_n^o(v)\right)^2$$



# New model [ROB-UE]

$$\max_{\boldsymbol{\lambda}} \inf_{\mathbf{v} \geq 0} L_\rho(\mathbf{v}, \boldsymbol{\lambda})$$

$$\mathbf{v}^{(i+1)} := \arg\min_{\mathbf{v} \geq 0} L_\rho\left(\mathbf{v}, \boldsymbol{\lambda}^{(i)}\right) \quad (17)$$

$$\boldsymbol{\lambda}^{(i+1)} := \boldsymbol{\lambda}^{(i+1)} + \rho\left(C \cdot \mathbf{v}^{(i+1)} - \mathbf{g}\right) \quad (18)$$

$$L_\rho(\mathbf{v}, \boldsymbol{\lambda}) = \sum_{a \in A} \int_0^{\sum_{o \in O} v_a^o} t_a(w)dw + \sum_{o \in O} \sum_{n \in N} \lambda_n^o H_n^o(v) + \frac{\rho}{2} \sum_{o \in O} \sum_{n \in N} \left(H_n^o(v)\right)^2 \quad (15)$$

$$H_n^o(v) := \sum_{\substack{a \in A \\ i(a)=n}} v_a^o - \sum_{\substack{a \in A \\ h(a)=n}} v_a^o - g_n^o, \forall o \in O, n \in N$$

# New model [ROB-UE]

$$\max_{\boldsymbol{\lambda}} \inf_{\mathbf{v}\geq 0} L_\rho(\mathbf{v}, \boldsymbol{\lambda})$$

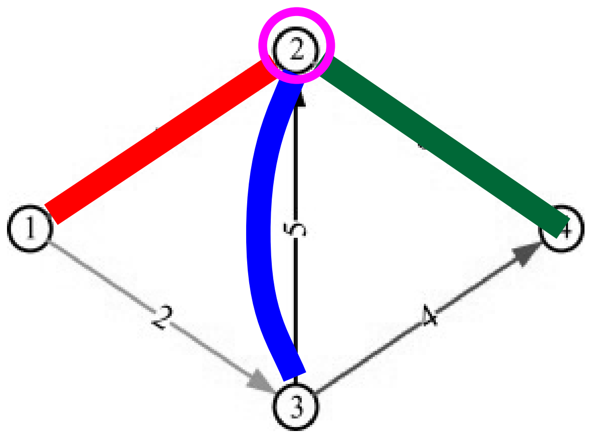
$$\mathbf{v}^{(i+1)} := \arg\min_{\mathbf{v}\geq 0} L_\rho\left(\mathbf{v}, \boldsymbol{\lambda}^{(i)}\right) \quad (17)$$

$$\boldsymbol{\lambda}^{(i+1)} := \boldsymbol{\lambda}^{(i+1)} + \rho\left(C \cdot \mathbf{v}^{(i+1)} - \mathbf{g}\right) \quad (18)$$

$$L_\rho(\mathbf{v}, \boldsymbol{\lambda}) = \sum_{a\in A} \int_0^{\sum_{o\in O} v_a^o} t_a(w)dw + \sum_{o\in O}\sum_{n\in N} \lambda_n^o H_n^o(v) + \frac{\rho}{2}\sum_{o\in O}\sum_{n\in N}\left(H_n^o(v)\right)^2 \quad (15)$$

$$H_n^o(v) := \sum_{\substack{a\in A \\ i(a)=n}} v_a^o - \sum_{\substack{a\in A \\ h(a)=n}} v_a^o - g_n^o, \forall o \in O, n \in N$$

**INDEPENDENT**




$$H_2^1 = \left(v_1^1 + v_5^1\right) - \left(v_3^1\right) - g_3^1$$

# New model [ROB-UE]

$$\max_{\boldsymbol{\lambda}} \inf_{\mathbf{v} \geq 0} L_\rho(\mathbf{v}, \boldsymbol{\lambda})$$

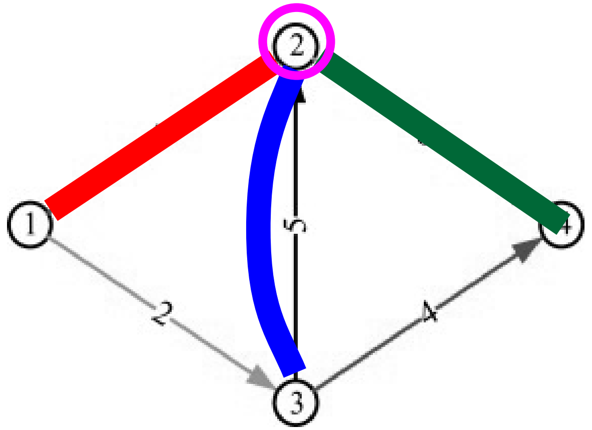
$$\mathbf{v}^{(i+1)} := \arg\min_{\mathbf{v} \geq 0} L_\rho\left(\mathbf{v}, \boldsymbol{\lambda}^{(i)}\right) \quad (17)$$

$$\boldsymbol{\lambda}^{(i+1)} := \boldsymbol{\lambda}^{(i+1)} + \rho\left(C \cdot \mathbf{v}^{(i+1)} - \mathbf{g}\right) \quad (18)$$

$$L_\rho(\mathbf{v}, \boldsymbol{\lambda}) = \sum_{a \in A} \int_0^{\sum_{o \in O} v_a^o} t_a(w)dw + \sum_{o \in O} \sum_{n \in N} \lambda_n^o H_n^o(v) + \frac{\rho}{2} \sum_{o \in O} \sum_{n \in N} \left(H_n^o(v)\right)^2 \quad (15)$$

$$H_n^o(v) := \sum_{\substack{a \in A \\ i(a)=n}} v_a^o - \sum_{\substack{a \in A \\ h(a)=n}} v_a^o - g_n^o, \forall o \in O, n \in N$$

**INDEPENDENT**




$$H_2^1 = \left(v_1^1 + v_5^1\right) - \left(v_3^1\right) - g_3^1$$

## SOLUTION

**The alternating direction method of multipliers (ADMM)**

# Contents

# The alternating direction method of multipliers (ADMM)

**[P1]** $$\min_{x} f(x) + g(x)$$

**[P2]** $$\min_{x,y} f(x) + g(y), \quad \text{s.t.}\ x = y$$

https://en.wikipedia.org/wiki/Augmented_Lagrangian_method

# The alternating direction method of multipliers (ADMM)

**[P1]** $$\min_x f(x) + g(x)$$

Requiring solving a proximity function in $x$ and $y$ at **the same time**

**[P2]** $$\min_{x,y} f(x) + g(y), \quad \text{s.t.}\ x = y$$

**[Augumented Lagransian]**

$$L_\lambda(x, y, \lambda) = f(x) + g(y) + \langle \lambda, x - y \rangle + \frac{\rho}{2} \|x - y\|^2$$

**[Algorithms]**

$$x^{(i+1)} \coloneqq \arg\min_x L_\rho\left(x, y^{(i)}, \lambda^{(i)}\right)$$ **FIX** $y$

$$y^{(i+1)} \coloneqq \arg\min_y L_\rho\left(x^{(i+1)}, y, \lambda^{(i)}\right)$$ **FIX** $x$

$$\lambda^{(i+1)} \coloneqq \lambda^{(i)} + \lambda\left(x^{(i+1)} - y^{(i+1)}\right)$$

**ADMM** allows this problem to be solved **step by step**

https://en.wikipedia.org/wiki/Augmented_Lagrangian_method

# The alternating direction method of multipliers (ADMM)

[P1] $$\min_x f(x) + g(x)$$

[P2] $$\min_{x,y} f(x) + g(y),\quad \text{s.t.}\ x = y$$

[Augumented Lagransian]

$$L_\lambda(x, y, \lambda) = f(x) + g(y) + \langle \lambda, x - y \rangle + \frac{\rho}{2}\|x - y\|^2$$

[Algorithms]

$$x^{(i+1)} \coloneqq \arg\min_x L_\rho\left(x, y^{(i)}, \lambda^{(i)}\right)$$
$$y^{(i+1)} \coloneqq \arg\min_y L_\rho(x^{(i+1)}, y, \lambda^{(i)})$$
$$\lambda^{(i+1)} \coloneqq \lambda^{(i)} + \lambda(x^{(i+1)} - y^{(i+1)})$$

*This paper*

$$L_\rho(\mathbf{v}, \boldsymbol{\lambda}) = \sum_{a\in A} \int_0^{\sum_{o\in O} v_a^o} t_a(w)dw + \sum_{o\in O}\sum_{n\in N} \lambda_n^o H_n^o(v) + \frac{\rho}{2}\sum_{o\in O}\sum_{n\in N}\left(H_n^o(v)\right)^2$$

***Adopt new concept***

***Block*** : Group of links **independent** each other

# Block

***DEFINITION***

- Links in each block are **independent.** No overlapped tail/head nodes into the same block/set.
- For each block, the augmented Lagrangian function can then be solved/updated in **parallel.**

$$C \cdot \mathbf{v} - \mathbf{g} = \mathbf{0} \quad (11)$$

$$\mathbf{v}^{(i+1)} \coloneqq \underset{\mathbf{v} \geq 0}{\arg\min}\, L_\rho\left(\mathbf{v}, \boldsymbol{\lambda}^{(i)}\right) \quad (17)$$

$$\boldsymbol{\lambda}^{(i+1)} \coloneqq \boldsymbol{\lambda}^{(i+1)} + \rho\left(C \cdot \mathbf{v}^{(i+1)} - \mathbf{g}\right) \quad (18)$$

$$C_1 \cdot \mathbf{v}_{B_1} - C_2 \cdot \mathbf{v}_{B_2} - \mathbf{g} = \mathbf{0} \quad (19)$$

$$\mathbf{v}_{B_1}^{(i+1)} \coloneqq \underset{\mathbf{v}_{B_1} \geq 0}{\arg\min}\, L_\rho\left(\mathbf{v}_{B_1}, \mathbf{v}_{B_1}^{(i)}, \boldsymbol{\lambda}^{(i)}\right) \quad (20)$$

$$\mathbf{v}_{B_2}^{(i+1)} \coloneqq \underset{\mathbf{v}_{B_1} \geq 0}{\arg\min}\, L_\rho\left(\mathbf{v}_{B_1}, \mathbf{v}_{B_1}^{(i)}, \boldsymbol{\lambda}^{(i)}\right) \quad (21)$$

$$\boldsymbol{\lambda}^{(i+1)} \coloneqq \boldsymbol{\lambda}^{(i)} + \rho\left(C_1 \cdot \mathbf{v}_{B_1}^{(i+1)} + C_2 \cdot \mathbf{v}_{B_2}^{(i+1)} - \mathbf{g}\right) \quad (22)$$

$$L_\rho^{B_p}\left(\mathbf{v}_{B_1}^{(i+1)}, \ldots, \mathbf{v}_{B_{p-1}}^{(i+1)}, \mathbf{v}_{B_p}, \mathbf{v}_{B_{p+1}}^{(i)}, \ldots, \mathbf{v}_{B_P}^{(i)}, \boldsymbol{\lambda}^{(i)}\right) = \sum_{a \in B_p} L_\rho^{B_p,a}\left(\mathbf{v}_a\right)$$

# Block

***DEFINITION***

- Links in each block are **independent.** No overlapped tail/head nodes into the same block/set.
- For each block, the augmented Lagrangian function can then be solved/updated in **parallel.**

$$C \cdot \mathbf{v} - \mathbf{g} = \mathbf{0} \quad (11)$$

$$\mathbf{v}^{(i+1)} := \underset{\mathbf{v} \geq 0}{\arg\min}\, L_\rho\left(\mathbf{v}, \boldsymbol{\lambda}^{(i)}\right) \quad (17)$$

$$\boldsymbol{\lambda}^{(i+1)} := \boldsymbol{\lambda}^{(i+1)} + \rho\left(C \cdot \mathbf{v}^{(i+1)} - \mathbf{g}\right) \quad (18)$$

→

$$C_1 \cdot \mathbf{v}_{B_1} - C_2 \cdot \mathbf{v}_{B_2} - \mathbf{g} = \mathbf{0} \quad (19)$$

$$\mathbf{v}_{B_1}^{(i+1)} := \underset{\mathbf{v}_{B_1} \geq 0}{\arg\min}\, L_\rho\left(\mathbf{v}_{B_1}, \mathbf{v}_{B_1}^{(i)}, \boldsymbol{\lambda}^{(i)}\right) \quad (20)$$

$$\mathbf{v}_{B_2}^{(i+1)} := \underset{\mathbf{v}_{B_1} \geq 0}{\arg\min}\, L_\rho\left(\mathbf{v}_{B_1}, \mathbf{v}_{B_1}^{(i)}, \boldsymbol{\lambda}^{(i)}\right) \quad (21)$$

$$\boldsymbol{\lambda}^{(i+1)} := \boldsymbol{\lambda}^{(i)} + \rho\left(C_1 \cdot \mathbf{v}_{B_1}^{(i+1)} + C_2 \cdot \mathbf{v}_{B_2}^{(i+1)} - \mathbf{g}\right) \quad (22)$$

$$L_\rho^{B_p}\left(\mathbf{v}_{B_1}^{(i+1)}, \dots, \mathbf{v}_{B_{p-1}}^{(i+1)}, \mathbf{v}_{B_p}, \mathbf{v}_{B_{p+1}}^{(i)}, \dots, \mathbf{v}_{B_P}^{(i)}, \boldsymbol{\lambda}^{(i)}\right) = \sum_{a \in B_p} L_\rho^{B_p, a}(\mathbf{v}_a)$$

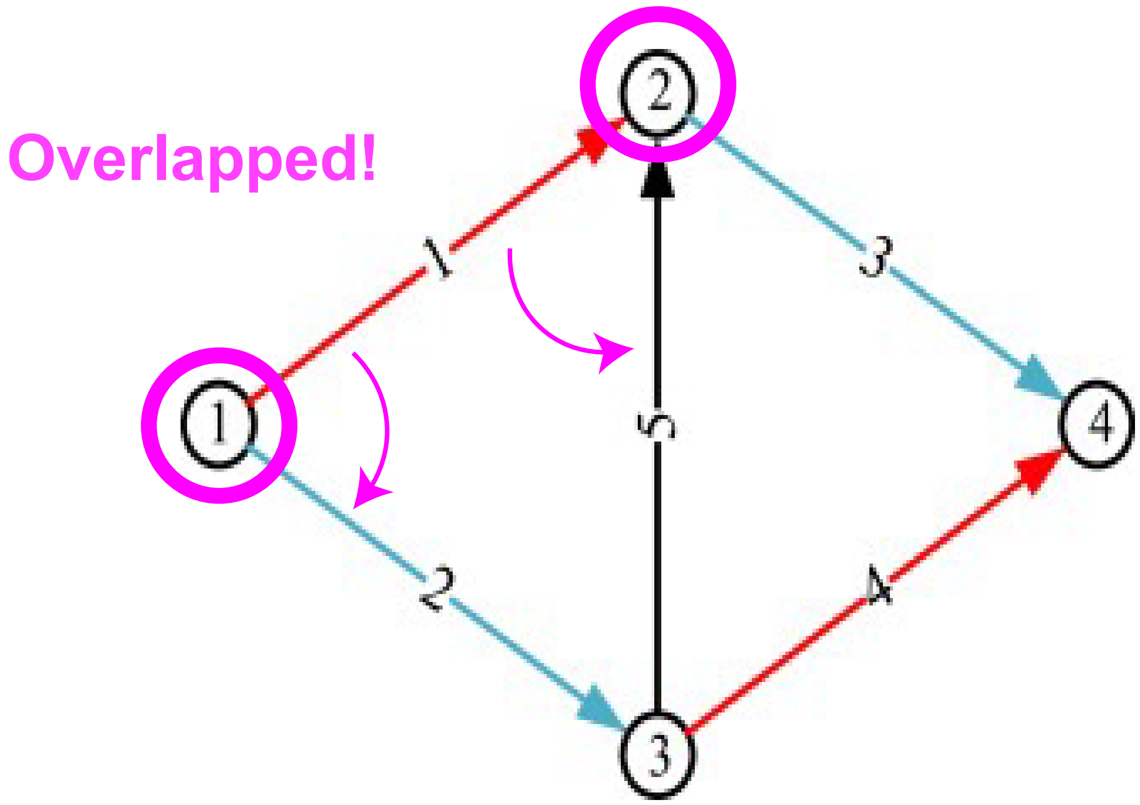
After update | Target | Before update

**Independent** link-based subproblems
= Suitable for **parallel** computing!

# What is “Block”

***DEFINITION***

- Links in each block are **independent.** No overlapped tail/head nodes into the same block/set.
- For each block, the augmented Lagrangian function can then be solved/updated in **parallel.**



# Algorithm

Iteration $i$

$$\mathbf{v}_{B_1}^{(i+1)} \coloneqq \arg\min_{\mathbf{v}_{B_1}\geq 0} L_\rho\left(\mathbf{v}_{B_1}, \mathbf{v}_{B_2}^{(i)}, \mathbf{v}_{B_3}^{(i)}, \boldsymbol{\lambda}^{(i)}\right)$$

$$\mathbf{v}_{B_2}^{(i+1)} \coloneqq \arg\min_{\mathbf{v}_{B_2}\geq 0} L_\rho\left(\mathbf{v}_{B_1}^{(i+1)}, \mathbf{v}_{B_2}, \mathbf{v}_{B_3}^{(i)}, \boldsymbol{\lambda}^{(i)}\right)$$

$$\mathbf{v}_{B_3}^{(i+1)} \coloneqq \arg\min_{\mathbf{v}_{B_3}\geq 0} L_\rho\left(\mathbf{v}_{B_1}^{(i+1)}, \mathbf{v}_{B_2}^{(i+1)}, \mathbf{v}_{B_3}, \boldsymbol{\lambda}^{(i)}\right)$$

$$\boldsymbol{\lambda}^{(i+1)} \coloneqq \boldsymbol{\lambda}^{(i)} + \rho\left(C \cdot \mathbf{v}^{(i+1)} - \mathbf{g}\right)$$

***Criterion***

$$|RG| = \left|1 - \frac{\sum_{od\in W} \varphi^{od} q^{od}}{\sum_{a\in A} v_a t_a(v_a)}\right|$$

# 3.2 アルゴリズム

Iteration $i$

$$\mathbf{v}_{B_1}^{(i+1)} := \arg\min_{\mathbf{v}_{B_1}\geq 0} L_\rho\left(\mathbf{v}_{B_1}, \mathbf{v}_{B_2}^{(i)}, \mathbf{v}_{B_3}^{(i)}, \boldsymbol{\lambda}^{(i)}\right)$$

$$\mathbf{v}_{B_2}^{(i+1)} := \arg\min_{\mathbf{v}_{B_2}\geq 0} L_\rho\left(\mathbf{v}_{B_1}^{(i+1)}, \mathbf{v}_{B_2}, \mathbf{v}_{B_3}^{(i)}, \boldsymbol{\lambda}^{(i)}\right)$$

$$\mathbf{v}_{B_3}^{(i+1)} := \arg\min_{\mathbf{v}_{B_3}\geq 0} L_\rho\left(\mathbf{v}_{B_1}^{(i+1)}, \mathbf{v}_{B_2}^{(i+1)}, \mathbf{v}_{B_3}, \boldsymbol{\lambda}^{(i)}\right)$$

$$\boldsymbol{\lambda}^{(i+1)} := \boldsymbol{\lambda}^{(i)} + \rho\left(C \cdot \mathbf{v}^{(i+1)} - \mathbf{g}\right)$$

***Criterion***

$$|RG| = \left|1 - \frac{\sum_{od\in W} \varphi^{od} q^{od}}{\sum_{a\in A} v_a t_a(v_a)}\right|$$

# Algorithm

Iteration $i$

$$\mathbf{v}_{B_1}^{(i+1)} := \underset{\mathbf{v}_{B_1}\geq 0}{\arg\min}\, L_\rho\left(\mathbf{v}_{B_1}, \mathbf{v}_{B_2}^{(i)}, \mathbf{v}_{B_3}^{(i)}, \boldsymbol{\lambda}^{(i)}\right)$$

$$\mathbf{v}_{B_2}^{(i+1)} := \underset{\mathbf{v}_{B_2}\geq 0}{\arg\min}\, L_\rho\left(\mathbf{v}_{B_1}^{(i+1)}, \mathbf{v}_{B_2}, \mathbf{v}_{B_3}^{(i)}, \boldsymbol{\lambda}^{(i)}\right)$$

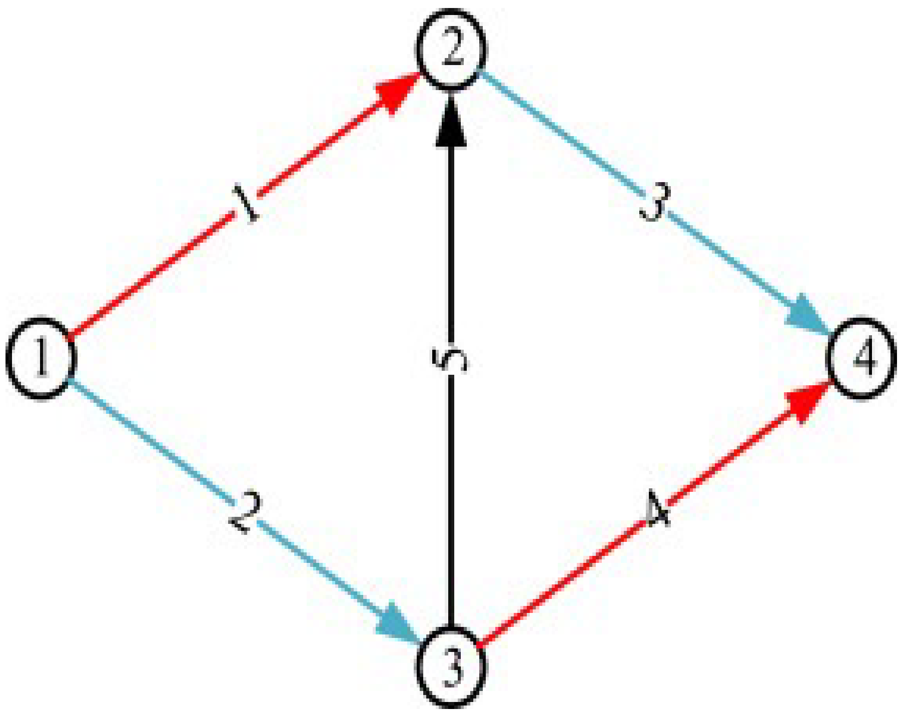
$$\mathbf{v}_{B_3}^{(i+1)} := \underset{\mathbf{v}_{B_3}\geq 0}{\arg\min}\, L_\rho\left(\mathbf{v}_{B_1}^{(i+1)}, \mathbf{v}_{B_2}^{(i+1)}, \mathbf{v}_{B_3}, \boldsymbol{\lambda}^{(i)}\right)$$

$$\boldsymbol{\lambda}^{(i+1)} := \boldsymbol{\lambda}^{(i)} + \rho\left(C\cdot\mathbf{v}^{(i+1)} - \mathbf{g}\right)$$

***Criterion***

$$|RG| = \left|1 - \frac{\sum_{od\in W}\varphi^{od}q^{od}}{\sum_{a\in A}v_a t_a(v_a)}\right|$$

**Q1. How to group links into Block?**



# Algorithm

Iteration $i$

$$\mathbf{v}_{B_1}^{(i+1)} := \underset{\mathbf{v}_{B_1} \geq 0}{\arg\min}\, L_\rho\left(\mathbf{v}_{B_1}, \mathbf{v}_{B_2}^{(i)}, \mathbf{v}_{B_3}^{(i)}, \boldsymbol{\lambda}^{(i)}\right)$$

$$\mathbf{v}_{B_2}^{(i+1)} := \underset{\mathbf{v}_{B_2} \geq 0}{\arg\min}\, L_\rho\left(\mathbf{v}_{B_1}^{(i+1)}, \mathbf{v}_{B_2}, \mathbf{v}_{B_3}^{(i)}, \boldsymbol{\lambda}^{(i)}\right)$$

$$\mathbf{v}_{B_3}^{(i+1)} := \underset{\mathbf{v}_{B_3} \geq 0}{\arg\min}\, L_\rho\left(\mathbf{v}_{B_1}^{(i+1)}, \mathbf{v}_{B_2}^{(i+1)}, \mathbf{v}_{B_3}, \boldsymbol{\lambda}^{(i)}\right)$$

$$\boldsymbol{\lambda}^{(i+1)} := \boldsymbol{\lambda}^{(i)} + \rho\left(C \cdot \mathbf{v}^{(i+1)} - \mathbf{g}\right)$$

***Criterion***

$$|RG| = \left| 1 - \frac{\sum_{od \in W} \varphi^{od} q^{od}}{\sum_{a \in A} v_a t_a(v_a)} \right|$$

**Q1. How to group links into Block?**

**Q2. How to solve link-based subproblem?**

$$L_\rho^{B_p}\left(\mathbf{v}_{B_1}^{(i+1)}, \ldots, \mathbf{v}_{B_{p-1}}^{(i+1)}, \mathbf{v}_{B_p}, \mathbf{v}_{B_{p+1}}^{(i)}, \ldots, \mathbf{v}_{B_P}^{(i)}, \boldsymbol{\lambda}^{(i)}\right)$$

$$= \sum_{a \in B_p} L_\rho^{B_p, a}(\mathbf{v}_a)$$

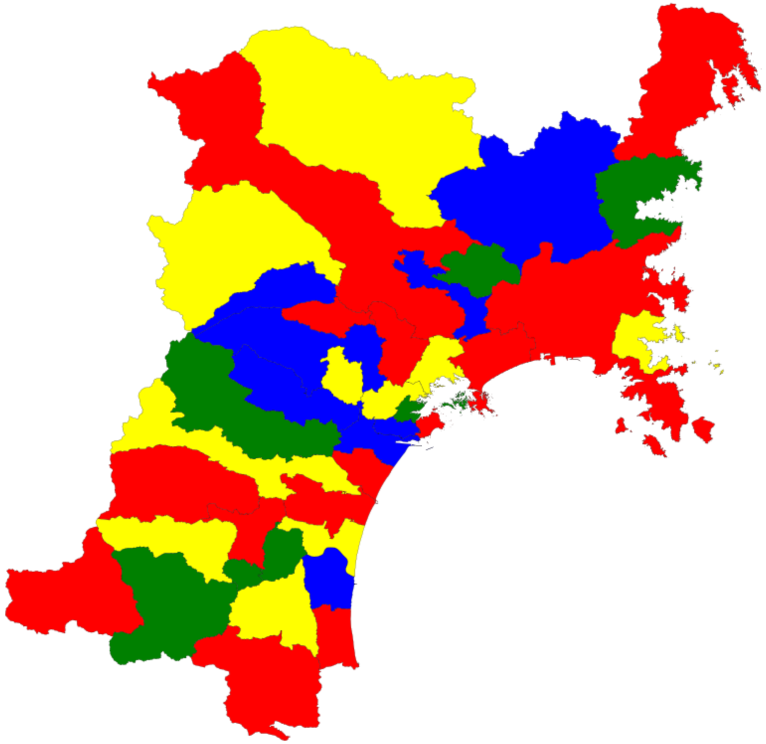
# Contents

# Link blocking



**Q1. How to group links into Block?**

*Edge coloring problem in graph theory* (Bollobas, 2013)



# Definition

**Definition 1**

(Degree): The degree (or valency) of a vertex/node of a graph is the number of edges that are incident to the vertex, denoted by$\deg(n), \forall n \in N$. The maximum degree of a graph $G$, denoted by $\Delta(G)$, is the largest degree of the vertices in the graph.

**Definition 2**

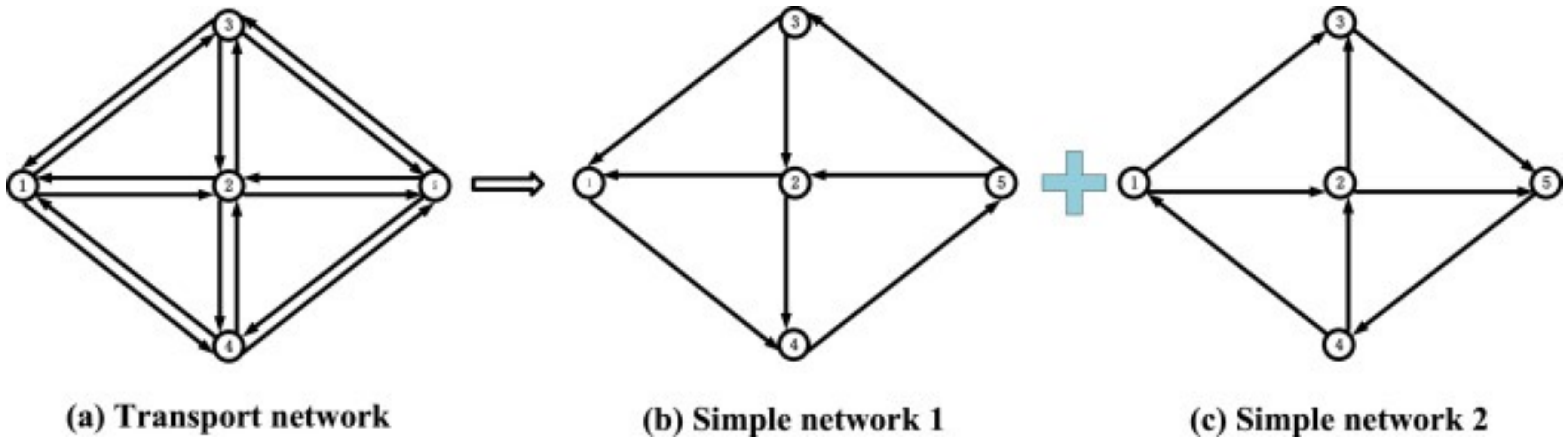
(Chromatic index): Given a graph, the minimum required number of colors for the edges is named the chromatic index of the graph, denoted by $\chi(G)$.

**Definition 3**

*(Multiplicity)*: The maximum number of edges in any bundle of parallel edges of a graph is called the multiplicity, denoted by $u(G)$. For a transport network, due to the exist of bi-directional road links, usually $u(G) = 2$.

For any multigraph, $\chi(G) \leq \Delta(G) + u(G)$ (Bollobás, 2013).

# Proposed method



# Contents

# 5.1 The link-based subproblem

**Q2. How to solve link-based subproblem?**

*"In this section, we proceed to solve the link-based minimization problem, which is the most **important**/time-consuming subproblem of ADMM"*

Iteration $i$

$$\mathbf{v}_{B_1}^{(i+1)} \coloneqq \underset{\mathbf{v}_{B_1}\geq 0}{\arg\min}\, L_\rho\left(\mathbf{v}_{B_1}, \mathbf{v}_{B_2}^{(i)}, \mathbf{v}_{B_3}^{(i)}, \boldsymbol{\lambda}^{(i)}\right)$$

$$\mathbf{v}_{B_2}^{(i+1)} \coloneqq \underset{\mathbf{v}_{B_2}\geq 0}{\arg\min}\, L_\rho\left(\mathbf{v}_{B_1}^{(i+1)}, \mathbf{v}_{B_2}, \mathbf{v}_{B_3}^{(i)}, \boldsymbol{\lambda}^{(i)}\right)$$

$$\mathbf{v}_{B_3}^{(i+1)} \coloneqq \underset{\mathbf{v}_{B_3}\geq 0}{\arg\min}\, L_\rho\left(\mathbf{v}_{B_1}^{(i+1)}, \mathbf{v}_{B_2}^{(i+1)}, \mathbf{v}_{B_3}, \boldsymbol{\lambda}^{(i)}\right)$$

$$\boldsymbol{\lambda}^{(i+1)} \coloneqq \boldsymbol{\lambda}^{(i)} + \rho\left(C \cdot \mathbf{v}^{(i+1)} - \mathbf{g}\right)$$

$$\min_{\substack{v_a \geq 0,\\ a\in B_p}} L_\rho^{B_p,a}(\mathbf{v}_a) = \int_0^{\sum_{o\in O} v_a^o} t_a(w)dw + \sum_{o\in O}\left[\lambda_{i(a)}^o\left(H_{i(a)}^o(v)\right) + \lambda_{h(a)}^o\left(H_{h(a)}^o(v)\right) + \frac{\rho}{2}\left(H_{i(a)}^o(v)\right)^2 + \frac{\rho}{2}\left(H_{h(a)}^o(v)\right)^2\right]$$

# Gradient projection method

$$e_1^1 = \left(\sum_{\substack{a\in B_2,\\ i(a)=1}} (v')_a^1 + \sum_{\substack{a\in B_3,\\ i(a)=1}} (v')_a^1\right) - \left(\sum_{\substack{a\in B_2,\\ h(a)=1}} (v')_a^1 + \sum_{\substack{a\in B_3,\\ h(a)=1}} (v')_a^1\right) - g_1^1$$

Iteration *j*

$$v_a^{o(j+1)} = \max\left[0, v_a^{o(j)} - \alpha\left(s_a^{o(j)}\right)^{-1}\left(d_a^{o(j)}\right)\right], \forall o \in O, a \in A$$

$$d_a^o = \frac{\partial}{\partial v_a^o} L_\rho^{B_p,a}(\mathbf{v}_a) = t_a\left(\sum_{o\in O} v_a^o\right) + 2\rho v_a^o + \left(\rho e_{i(a)}^o - \rho e_{h(a)}^o + \lambda_{i(a)}^o - \lambda_{h(a)}^o\right)$$

$$s_a^o = \frac{\partial^2}{\partial (v_a^o)^2} L_\rho^{B_p,a}(\mathbf{v}_a) = t'_a\left(\sum_{o\in O} v_a^o\right) + 2\rho$$

***Criterion***

$$AG^{(j)} = \sum_{o\in O}\left|d_a^{o(j)} \cdot v_a^{o(j)}\right|$$

$$EAG^{(j)} = \left|AG^{(j)} - AG^{(j-1)}\right|$$

# Contents

# Numerical examples

***Criterion***

$$|RG| = \left|1 - \frac{\sum_{od\in W} \varphi^{od} q^{od}}{\sum_{a\in A} v_a t_a(v_a)}\right|$$

$$AG^{(j)} = \sum_{o\in O} \left|d_a^{o(j)} \cdot v_a^{o(j)}\right|$$

$$EAG^{(j)} = \left|AG^{(j)} - AG^{(j-1)}\right|$$

***BPR function***

$$t_a = t_a^0 \left(1 + \alpha \left(\frac{v_a}{C_a}\right)^{\beta}\right)$$

***Environment***

Windows10 64 bit

AMD 4800 H 2.9GHz CPU

16G RAM

C++ **How many cores?**

| | AMD Ryzen 7 4800H | Apple M3 Pro 12 Core | Apple M1 8 Core 3200 MHz |
|---|---|---|---|
| Price | Search Online | Search Online | Search Online |
| Socket Type | FP6 | NA [2] | NA [2] |
| CPU Class | Laptop | Laptop | Desktop, Laptop, Mobile/Embedded |
| Clockspeed | 2.9 GHz | 4.0 GHz | 3.2 GHz |
| Turbo Speed | Up to 4.2 GHz | NA [2] | NA [2] |
| # of Physical Cores | 8 (Threads: 16) | 12 (Threads: 12) | 8 (Threads: 8) |
| Cache | L1: 512KB, L2: 4.0MB, L3: 8MB | NA [2] | NA [2] |
| TDP | 45W | NA [2] | 15.1W |
| Yearly Running Cost | $8.21 | NA | $2.74 |
| Other | with Radeon Graphics | 18 Core GPU | |
| First Seen on Chart | Q1 2020 | Q4 2023 | Q1 2021 |
| # of Samples | 3366 | 137 | 7675 |
| CPU Value | 0.0 | 0.0 | 0.0 |
| Single Thread Rating (% diff. to max in group) | 2616 (-45.7%) | 4818 (0.0%) | 3703 (-23.2%) |
| CPU Mark (% diff. to max in group) | 18576 (-32.7%) | 27592 (0.0%) | 14185 (-48.6%) |

https://www.cpubenchmark.net/

# Numerical examples

| Targets | Small | Sioux-Falls | Anaheim | Chicago-Sketch |
|---|---|---|---|---|
| **nodes** | 4 | 24 | 416 | 933 |
| **links** | 5 | 76 | 914 | 2950 |
| **Block** | 3 | 10~ | 12~ | 20~ |
| **OD-pair** | | | 1406 | 93,513 |
| **Purpose of the experiment** | Convergence | Comparing Algorithms & number of blocks | Comparing Algorithms & number of blocks | Comparing Algorithms |

https://github.com/bstabler/TransportationNetworks

# A Small Network

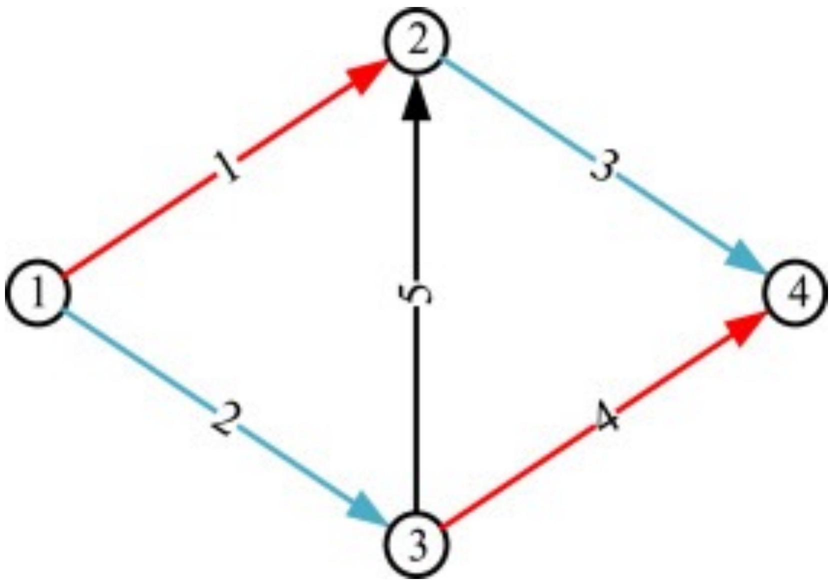
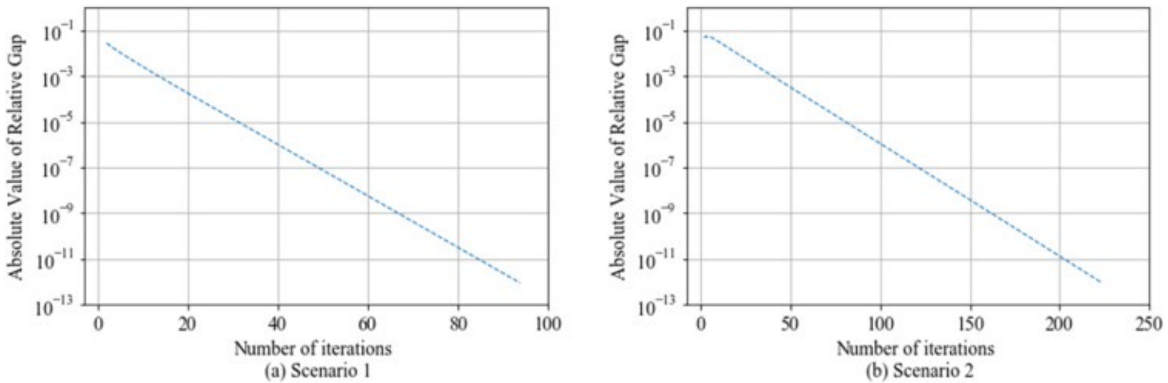



Table 1. Link attributes of the illustrative network.

| Link ID | Tail | Head | Free flow travel time | Capacity | Block ID |
|---|---|---|---|---|---|
| 1 | 1 | 2 | 3 | 10 | 1 |
| 2 | 1 | 3 | 2 | 10 | 2 |
| 3 | 2 | 4 | 4 | 10 | 2 |
| 4 | 3 | 2 | 1 | 10 | 3 |
| 5 | 3 | 4 | 5 | 10 | 1 |




**Convergence!**

# Sioux-Falls Network

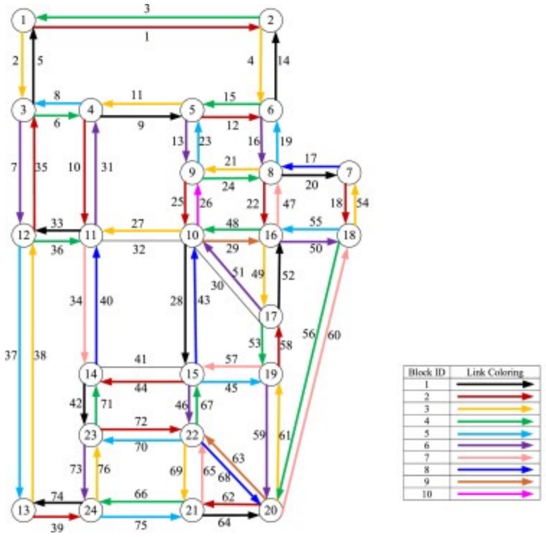
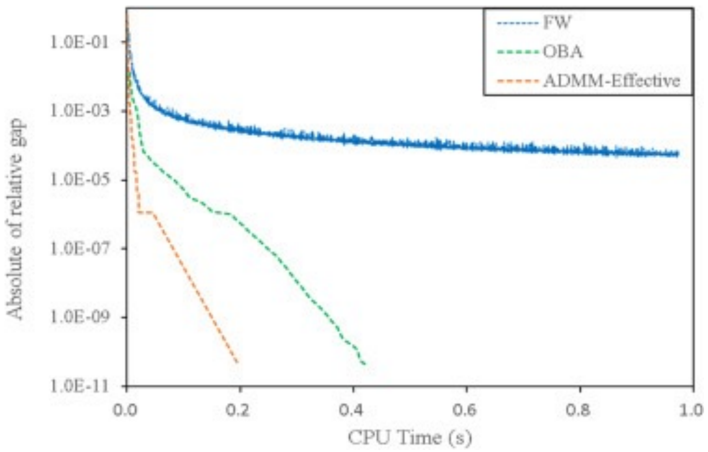
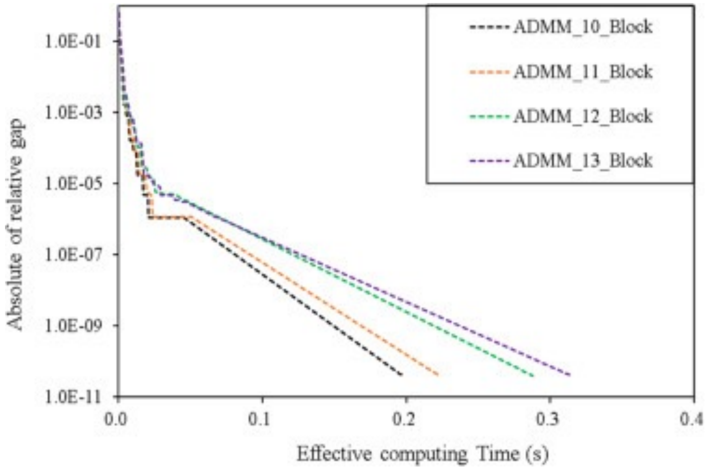



Table 5. Link blocking pattern.

| Block ID | Index of Member Links |
|---|---|
| 1 | 5,9,14,20,28,33,42,52,64,74 |
| 2 | 1,10,12,18,22,25,35,39,44,58,62,72 |
| 3 | 2,4,11,21,27,38,41,49,54,61,69,76 |
| 4 | 3,6,15,24,36,48,53,56,66,67,71 |
| 5 | 8,19,23,32,37,45,55,70,75 |
| 6 | 7,13,16,31,46,50,51,59,73 |
| 7 | 30,34,47,57,60,65 |
| 8 | 17,40,43,68 |
| 9 | 29,63 |
| 10 | 26 |



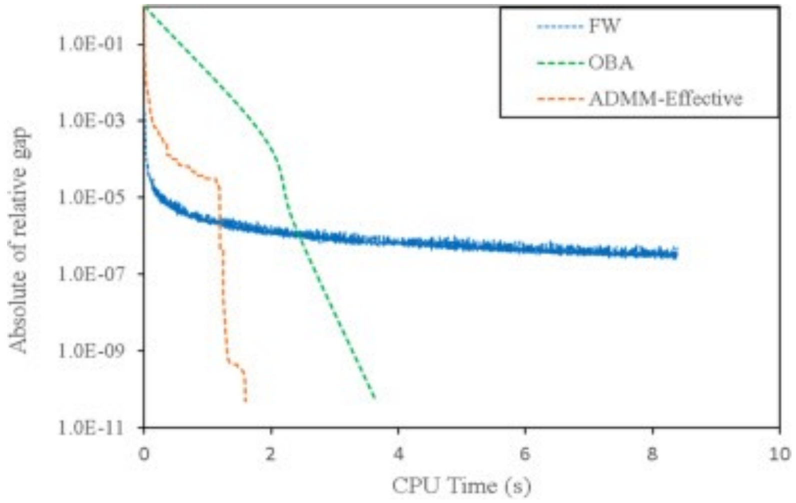

ADMM is better than FW, OBA

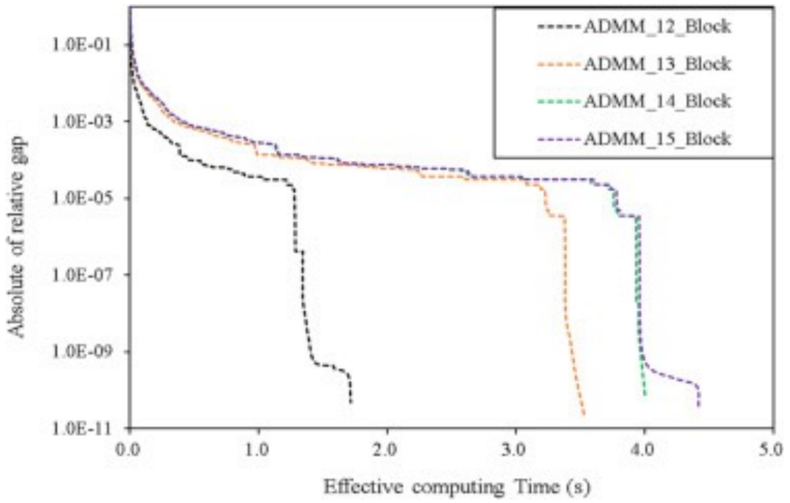



Fewer blocks, Faster convergence.

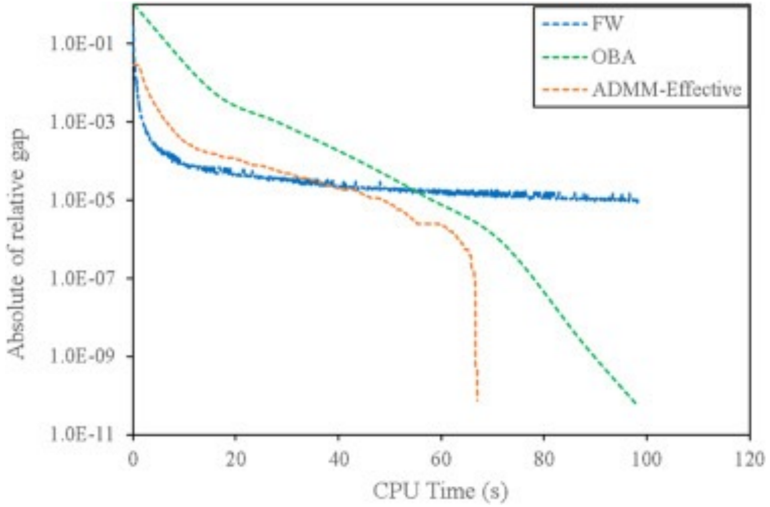
# Anaheim Network / Chicago-Sketch Network

Anaheim




Chicago



# Conclusions

## Main Challenge

Applying **parallel computing** approach to solve the **UE problem**

## Contribution

1. **Origin-base** formulation
2. The algorithm grouping links into **Blocks**

## Validation

- 4 numerical experiments
- The performance of ADMM is **superior** to some existing algorithms

*"This study presented an initial step on the aspect of using ADMM for parallel computing of UE."*

$$\min Z_2 = \sum_{a\in A}\int_0^{\sum_{o\in O} v_a^o} t_a(w)dw \quad (7)$$

$$\text{s.t.} \sum_{\substack{a\in A\\ i(a)=n}} v_a^o - \sum_{\substack{a\in A\\ h(a)=n}} v_a^o = g_n^o, \quad \forall o \in O, n \in N \quad (8)$$

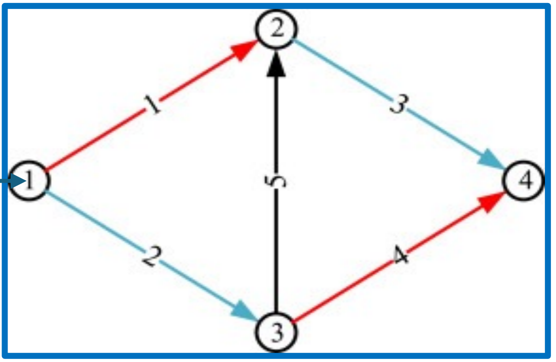
$$v_a \geq 0, \forall o \in O, a \in A \quad (9)$$

$$g_n^o = \begin{cases} \sum_{od\in W^o} q^{od}, n = o \\ -q^{od}, n = d \\ 0, \text{otherwise} \end{cases}, \forall o \in O, n \in N \quad (10)$$

$$L_0(\mathbf{v},\boldsymbol{\lambda}) = \sum_{a\in A}\int_0^{\sum_{o\in O} v_a^o} t_a(w)dw + \sum_{o\in O}\sum_{n\in N} \lambda_n^o H_n^o(v) \quad (12)$$

$$\max_{\boldsymbol{\lambda}} \inf_{\mathbf{v}\geq 0} L_0(\mathbf{v},\boldsymbol{\lambda}) \quad (13)$$

$$\mathbf{v}^* = \arg\min_{\mathbf{v}\geq 0} L_0(\mathbf{v},\boldsymbol{\lambda}^*) \quad (14)$$



# 所感

- 並列化が主目的ではないADMMが並列化手法と扱われるのは面白い
  - ADMMの分割という特性が並列化とマッチした？
  - とはいえ、大規模ネットワークならマルチスケールでいい気も…
  - そして、ブロック内リンク数=並列数なので、スパコンレベルではない？
  - 新たなUE計算のスタンダードとなるかは観察が必要
- 計算効率は大事
  - 変数同士の依存関係の解消が鍵
  - 計算環境にコア数が入っていない……
- 他の人の論文と比べると手法自体は簡単だった（気がする）