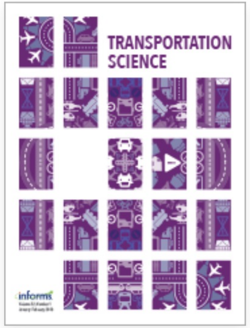
Simplicial Decomposition with Disaggregated Representation for the Traffic Assignment Problem



Volume 26, Issue 1

February 1992
Pages 1-68

Larsson, Torbjörn & Patriksson, Michael. (1992). Transportation Science. 26. 10.1287/trsc.26.1.4.
交通量配分問題に対する分解表現を用いた Simplicial Decomposition (単体分割) 法

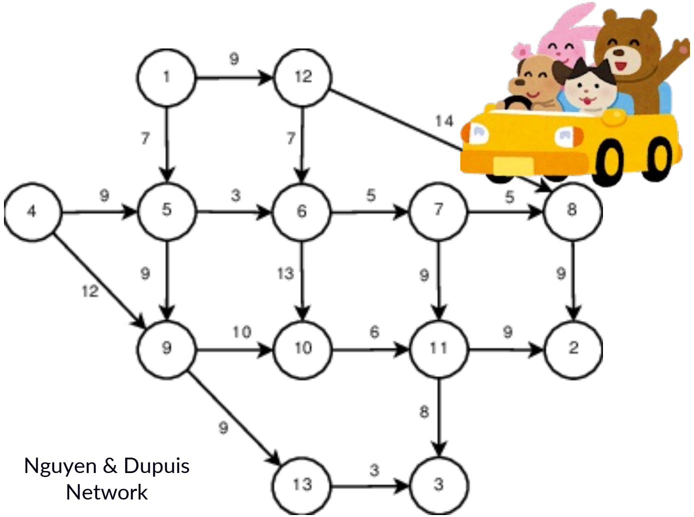
2024/05/27 理論談話会 #9

B4　重村奏画 / Soga Shigemura

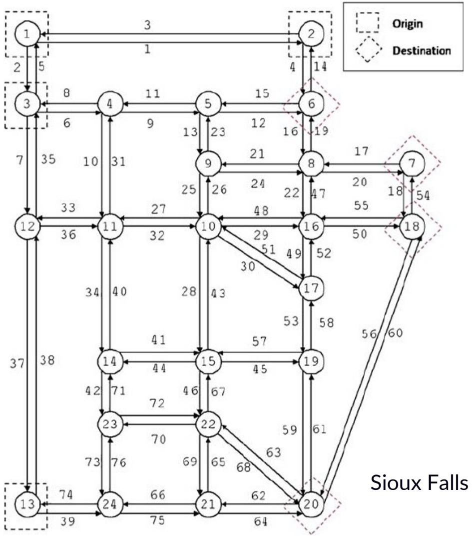
Objective

均衡配分の計算を効率化・並列化したい!

To Make Equilibrium Calculation Efficient & Parallel!



Nguyen & Dupuis Network



Sioux Falls Network

Abstract

User Equilibrium / UE (利用者均衡配分)

1956 — Frank-Wolfe Algorithm

↓

1979 — Simplicial Decomposition (SD)

↓

1992 — **Disaggregate Simplicial Decomposition (DSD)** ← **Proposed in this paper**

均衡配分の種類

	非混雑型 flow-independent	混雑型 flow-dependent
確定的経路選択	**all-or-nothing配分**	**利用者均衡配分(UE)**
確率的経路選択	**確率的配分**	**確率的利用者均衡配分(SUE)**

cf. スタートアップゼミ#2 by 増田さん

2 authors:

Torbjörn Larsson
Linköping University

123 PUBLICATIONS **2,205** CITATIONS

Michael Patriksson
Chalmers University of Technology

202 PUBLICATIONS **6,532** CITATIONS

Novelty / Utility / Reliability

Novelty

- Proposes an improved SD for UE problem, allowing the decomposed subproblems for each OD pair to be solved in parallel.
  利用者均衡配分のSD法を改善、部分問題の分解によりODペアごとに並列計算を可能に

Utility

- Requires fewer shortest path calculations and readily adapts to changes in network topology, especially useful for large-scale networks.
  最短経路探索の回数が減り、ネットワーク変化に容易に対応
  →大規模ネットワークで有用

Reliability

- Validates the algorithm's performance through numerical experiments on benchmark problems and a new large-scale network.
  ベンチマーク・大規模ネットワークの数値実験により検証

Solution

cf. 理論談話会#5 by 古橋さん

“An alternating direction method of multipliers for solving user equilibrium problem”

Algorithms

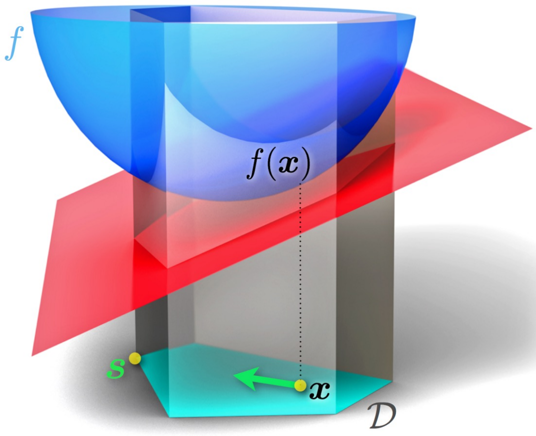
- Link-base: Frank-Wolfe algorithm, Gauss-Seidel iteration method
- Path-base ← This paper (DSD)
- Origin-base ~~(This paper)~~

“Larsson and Patriksson (1992) solved an OD-based auxiliary problem, and updated the solutions through a convex combination of extreme points.”

ODベースの補助問題を解き、極点の凸結合を通じて解を更新

“the decomposed subproblems can be solved in parallel.”

分割された部分問題が並列に解ける



Contents

User Equilibrium / UE (利用者均衡配分)

Frank-Wolfe Algorithm

↓

Simplicial Decomposition (SD)

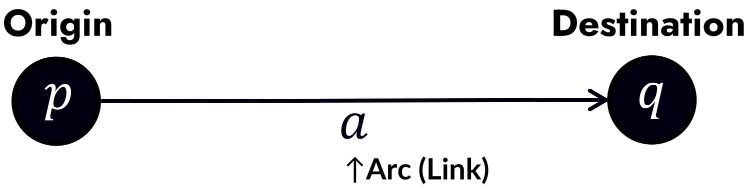
↓

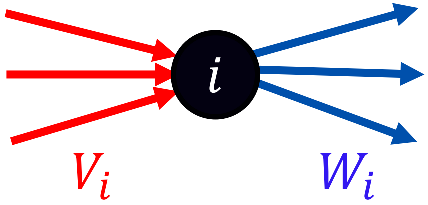
Disaggregate Simplicial Decomposition (DSD)

Definition

- Consider static traffic assignment problems, modelling peak-hour urban traffic
  ピーク時の都市交通における需要固定型利用者均衡配分モデル

Symbol	Definition
$G = (N, A)$	Transportation network (ネットワーク)
N	Set of nodes (ノードの集合)
A	Set of directed arcs (有向リンクの集合)
$a \in A$	A directed arc (有向リンク)
$t_a(\mathbf{f})$	Positive travel time for arc a (リンク a の正の旅行時間)
$\mathbf{f}$	Network flow (交通流)
$C \subset N \times N$	Set of OD pairs (OD ペアの集合)
$(p, q) \in C$	An OD pair (OD ペア)
d_{pq}	Positive flow demand for (p, q) (OD ペア (p, q) の正の需要)
f_{apq}	Flow from p to q through a (リンク a を通る p から q へのフロー)
$f_a = \sum_{(p,q)\in C} f_{apq}$	Total arc flow (リンクの総フロー)
$i \in N$	A specific node (特定のノード)
W_i, V_i	Arcs initiated/terminated at i (ノード i で始まる/終わるリンク)





Wardrop's Principles

1. **<u>User Equilibrium: 等時間原則</u>**
   The journey times in all routes actually used are equal and less than those that would be experienced by a single vehicle on any unused route.
   利用される経路の旅行時間は皆等しく、利用されない経路の旅行時間よりも小さいか、せいぜい等しい。

2. **<u>System Optimal: 所要時間最小則</u>**
   The average journey time is a minimum.
   道路ネットワーク上の総旅行時間が最小となる。

UE/FDの定式化 – 等価最適化問題への変換

■ Wardropの利用者均衡の等価最適化問題

UE/FD-Primal

$$\min Z(\boldsymbol{x}) = \sum_{a\in A}\int_0^{x_a} t_a(w)dw$$

s.t.

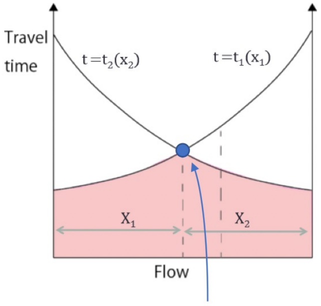
$$\sum_{k\in K_{rs}} f_k^{rs} - q_{rs} = 0 \qquad \forall rs \in \Omega$$

$$x_a = \sum_{rs\in\Omega}\sum_{k\in K_{rs}} f_k^{rs}\delta_{a,k}^{rs} \qquad \forall a \in \mathrm{A}$$

$$f_k^{rs} \geq 0 \qquad \forall k \in K_{rs}, \forall rs \in \Omega$$

$$x_a \geq 0 \qquad \forall a \in A$$

目的関数は直感的には下のように理解できる



均衡点：積分の値が最小となる点

$t_a(x_a)$: リンクaの旅行時間
x_a: リンクaの交通量
f_k^{rs}: ODペアrs間のパスkの流量
q_{rs}: ODペアrs間の分布交通量
$\delta_{a,k}^{rs}$: ODペアrs間のパスkがリンクaを含むか否か（True=1, False=0）

cf. スタートアップゼミ#2 by 増田さん

- 十分性の証明（詳しい証明は教科書や昨年度資料参照）
  UE/FD-PrimalのKKT条件が元の問題と一致することにより証明できる
- 解の一意性の証明（詳しい証明は教科書や昨年度資料参照）
  変数の実行可能領域が凸（∵制約条件式が全て線形）
  目的関数が狭義の凸関数⇔ Hessianが正定値（∵リンクパフォーマンス関数が単調増加）

Formulation: Traffic Assignment Problem

$$[\textbf{TAP}] \quad \min \quad T(\mathbf{f}) = \sum_{a \in \mathcal{A}} g_a(f_a) \qquad g_a(f_a) = \int_0^{f_a} t_a(s)ds$$

←旅行時間の最小化

$$\text{s.t.} \quad \sum_{a \in \mathcal{W}_i} f_{apq} - \sum_{a \in \mathcal{V}_i} f_{apq} = \begin{cases} d_{pq} & \text{if } i = p \\ -d_{pq} & \text{if } i = q \\ 0 & \text{otherwise} \end{cases} \qquad \forall\, i \in \mathcal{N} \quad \forall\, (p,q) \in \mathcal{C}$$

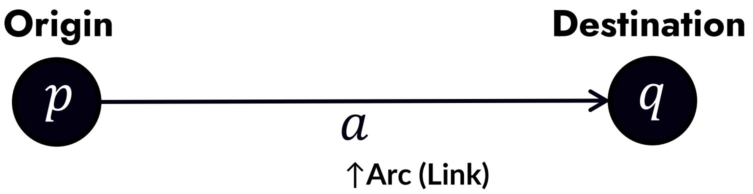
←流量保存則

$$\sum_{(p,q) \in \mathcal{C}} f_{apq} = f_a \qquad \forall\, a \in \mathcal{A}$$

$$f_{apq} \geq 0 \qquad \forall\, a \in \mathcal{A} \quad \forall\, (p,q) \in \mathcal{C}.$$

←流量は非負

↑ Arc-Node Formulation
リンク-ノード定式化



Contents

User Equilibrium / UE (利用者均衡配分)

Frank-Wolfe Algorithm

⬇

Simplicial Decomposition
(SD)

⬇

Disaggregate Simplicial Decomposition
(DSD)

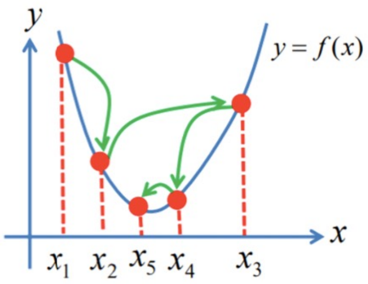
UE/FDの解法 User Equilibrium with Fixed Demand

cf. スタートアップゼミ#2 by 増田さん

■ **非線形最適化問題の一般的な解法**

ステップA：降下方向の探索
どの方向に向かえば目的関数が減少するか

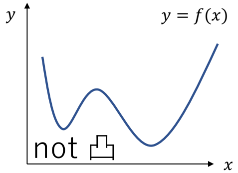
ステップB：ステップサイズの探索
降下方向にどこまで進めるか

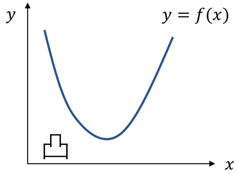


ステップAとステップBを繰り返すことで局所最適解を見つける

（UE/FD-Primalのような）**凸計画問題**では，局所最適解が大域的最適解に一致する→解の一意性

目的関数が凸関数で，実行可能領域が凸集合であるような最適化問題





fが凸関数
⇔任意のa,bに対して，
$\lambda f(a)+(1-\lambda)f(b) \geq f(\lambda a+(1-\lambda)b)\ \forall\lambda\in[0,1]$

→関数上の任意の2点をとって線分を引いた時，その線が必ず元の関数より上に来る

Frank-Wolfe Algorithm

「降下方向の探索」→「ステップサイズの探索」を繰り返し行う

- **Linearized Subproblem: 部分線形化問題**

$$[\mathbf{LP}] \qquad \min \quad \underline{T}\,(\mathbf{y}) = T\left(\mathbf{f}^{(k)}\right) + \nabla T\left(\mathbf{f}^{(k)}\right)^{\mathrm{T}} \cdot \left(\mathbf{y} - \mathbf{f}^{(k)}\right) \qquad \mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + l^{(k)} \cdot \left(\hat{\mathbf{y}}^{(k)} - \mathbf{f}^{(k)}\right)$$

←旅行時間の最小化

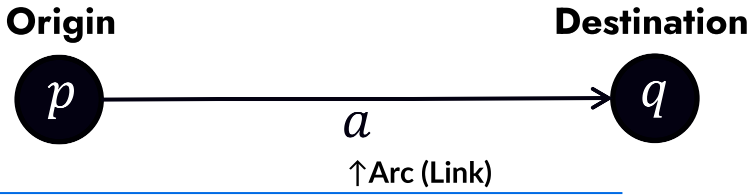
$$\begin{aligned} \text{s.t.} \quad \sum_{a \in \mathcal{W}_i} y_{apq} - \sum_{a \in \mathcal{V}_i} y_{apq} &= \begin{cases} d_{pq} & \text{if } i = p \\ -d_{pq} & \text{if } i = q \\ 0 & \text{otherwise} \end{cases} && \forall\, i \in \mathcal{N} \quad \forall\,(p,q) \in \mathcal{C} \\ \sum_{(p,q) \in \mathcal{C}} y_{apq} &= y_a && \forall\, a \in \mathcal{A} \\ y_{apq} &\geq 0 && \forall\, a \in \mathcal{A} \quad \forall\,(p,q) \in \mathcal{C} \end{aligned}$$

←流量保存則

←流量は非負

- **Line Search Problem: 直線探索問題**

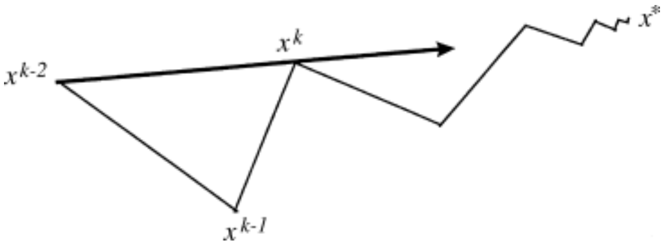
$$[\mathbf{LS}] \qquad \min_{l \in [0,1]} \overline{T}\,(l) = T\left(\mathbf{f}^{(k)} + l \cdot \left(\hat{\mathbf{y}}^{(k)} - \mathbf{f}^{(k)}\right)\right)$$



Frank-Wolfe Algorithm

Pros

- Efficient in the first iterations　初期反復における効率
- Ease of implementation　実装の容易さ



Cons

- Convergence rate is arithmetic
  → Slow convergence rate in later iterations　収束速度の遅さ
- Search direction tend to be perpendicular to the steepest descent direction as iterations increase
  反復が進むと探索方向が最急降下方向と直角に近くなる
- Generation of cyclic flows, which degrade performance
  循環フローによる性能低下

Contents

User Equilibrium / UE (利用者均衡配分)

Frank-Wolfe Algorithm

↓

Simplicial Decomposition (SD)

↓

Disaggregate Simplicial Decomposition (DSD)

Simplicial Decomposition (SD)

均衡解を端点解 (all-or-nothing配分) の凸結合 (重み付き重ね合わせ) と捉え、それらのウェイトを求める

- **Nonlinear Problem: 非線形問題**

[NLP] $$\boxed{\mathbf{x}^*} \in \arg\min_{\mathbf{x}\in\mathbf{X}} T(\mathbf{x})$$

X: 有界凸多面体集合

k回目の反復で $\left\{\hat{\mathbf{y}}^{(1)}, \hat{\mathbf{y}}^{(2)}, \ldots, \hat{\mathbf{y}}^{(l)}\right\}$

- **Master Problem: マスター問題**

[MP]
$$\begin{aligned} \min \quad & T\left(\sum_{i=1}^{l} \lambda_i \boxed{\hat{\mathbf{y}}^{(i)}}\right) \\ \text{s.t.} \quad & \sum_{i=1}^{l} \lambda_i = 1 \\ & \lambda_i \geq 0 \quad i = 1, \ldots, l. \end{aligned}$$

λ_i: 非負の重み係数で和が1

- MP **generalizes** the line search in the Frank-Wolfe algorithm.
  MPはFrank-Wolfeアルゴリズムの直線探索を一般化したもの。
- Converges with a **linear** rate, and the convergence is **finite**, allowing **removing** extreme points with small weights from the problem.
  線形収束し、収束が有限であるため、重みが小さな端点を削除することが可能になる。

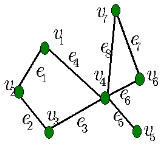
Simplicial Decomposition (SD)

- <u>T</u>raffic <u>A</u>ssignment <u>P</u>roblem (Again)

$$
\begin{array}{llrcll}
\text{[TAP]} & \min & T(\mathbf{f}) & = & \sum_{a\in\mathcal{A}} g_a(f_a) & \quad f_a = \sum_{(p,q)\in\mathcal{C}} \sum_{r\in\mathcal{R}_{pq}} \delta_{pqra} h_{pqr} \\
& \text{s.t.} & \sum_{r\in\mathcal{R}_{pq}} h_{pqr} & = & d_{pq} & \qquad \forall\ (p,q)\in\mathcal{C} \\
& & h_{pqr} & \geq & 0 & \quad \forall\ r\in\mathcal{R}_{pq} \quad \forall\ (p,q)\in\mathcal{C} \\
& & \sum_{(p,q)\in\mathcal{C}} \sum_{r\in\mathcal{R}_{pq}} \delta_{pqra} h_{pqr} & = & f_a & \quad \forall\ a\in\mathcal{A}.
\end{array}
$$

(δ_{pqra}): リンク-ルート接続行列
└ルートrがリンクaを含めば1
h_{pqr}: ルートr上のp→qへのフロー

接続行列
incidence matrix



	e_1	e_2	e_3	e_4	e_5	e_6	e_8	e_9
v_1	1	0	0	1	0	0	0	0
v_2	1	1	0	0	0	0	0	0
v_3	0	1	1	0	0	0	0	0
v_4	0	0	1	1	1	1	0	1
v_5	0	0	0	0	1	0	0	0
v_6	0	0	0	0	0	1	1	0
v_7	0	0	0	0	0	0	1	1

1

(これはノードとリンクの例)

↑ Arc-Route Formulation
リンク-ルート定式化

- If enumerated, grows exponentially with the size of network.
  全経路を列挙すると、ネットワークが大きくなると実用的ではない。
- →Solve restricted problem, find the shortest routes, and repeat.
- →制限問題を解いて最短経路探索をするのを繰り返す

→ Column Generation Method

Contents

User Equilibrium / UE (利用者均衡配分)

Frank-Wolfe Algorithm

↓

Simplicial Decomposition
(SD)

↓

**Disaggregate Simplicial Decomposition
(DSD)**

Disaggregate Simplicial Decomposition (DSD)

- Disaggregate Master Problem

$$
\begin{aligned}
\text{[DMP]} \qquad & \min \quad T\left(\sum_{j=1}^{m_1}\lambda_1^{(j)}\hat{\mathbf{y}}_1^{(j)}, \sum_{j=1}^{m_2}\lambda_2^{(j)}\hat{\mathbf{y}}_2^{(j)}, \ldots, \sum_{j=1}^{m_n}\lambda_n^{(j)}\hat{\mathbf{y}}_n^{(j)}\right) \\
& \text{s.t.} \quad \sum_{j=1}^{m_i}\lambda_i^{(j)} = 1 \qquad i = 1, 2, \ldots n \\
& \qquad\quad \lambda_i^{(j)} \geq 0 \quad j = 1, 2, \ldots, m_i \quad i = 1, 2, \ldots n.
\end{aligned}
$$

直積集合 $\mathbf{X} = \prod_{i=1}^{n}\mathbf{X}_i$ で表し、
各部分集合 $\mathbf{X}_i$ に対して1つの凸性制約を使って分割

λ_i: 非負の重み係数で和が1

- Restricted Master Problem

$$
\begin{aligned}
\text{[RMP]} \qquad & \min \quad T(\boldsymbol{\lambda}) \qquad f_a = \sum_{(p,q)\in\mathcal{C}} d_{pq} \sum_{i\in\hat{\mathcal{R}}_{pq}} \delta_{pqia}\lambda_{pqi} \\
& \text{s.t.} \quad \sum_{i\in\hat{\mathcal{R}}_{pq}} \lambda_{pqi} = 1 \qquad \forall\, (p,q) \in \mathcal{C} \\
& \qquad\quad \lambda_{pqi} \geq 0 \quad \forall\, i \in \hat{\mathcal{R}}_{pq} \quad \forall\, (p,q) \in \mathcal{C}.
\end{aligned}
$$

重み付きルートフローからリンクフローを計算

λ_{pqi}: 非負の重み係数で和が1

  - [RMP]はリンク-ルート定式化と同じである

Disaggregate Simplicial Decomposition (DSD)

- **Initialization:**

Let $k = 0$, and choose a point $\boldsymbol{\lambda}^{(0)}$ such that

$$\begin{aligned} \sum_{i \in \hat{\mathcal{R}}_{pq}} \lambda_{pqi}^{(0)} &= 1 && \forall\ (p,q) \in \mathcal{C} \\ \lambda_{pqi}^{(0)} &\geq 0 \quad \forall\ i \in \hat{\mathcal{R}}_{pq} && \forall\ (p,q) \in \mathcal{C}. \end{aligned}$$

- **Search direction generation:**

Let

$$i_{pq} \in \arg\max_{i \in \hat{\mathcal{R}}_{pq}} \left\{ \lambda_{pqi}^{(k)} \right\}$$

be the basic variable index and let N_{pq} denote the set of nonbasic variables for commodity (p,q), respectively. Compute

$$\mathbf{r}_{pq} = \nabla_{N_{pq}} T\left(\boldsymbol{\lambda}^{(k)}\right) - \nabla_{i_{pq}} T\left(\boldsymbol{\lambda}^{(k)}\right) \cdot \mathbf{1},$$

where each component of $\nabla T\left(\boldsymbol{\lambda}^{(k)}\right)$ is calculated as

$$\frac{\partial}{\partial \lambda_{pqi}} T\left(\boldsymbol{\lambda}^{(k)}\right) = d_{pq} \sum_{a \in \mathcal{A}} \delta_{pqia} g_a^{'} \left(f_a^{(k)}\right).$$

Disaggregate Simplicial Decomposition (DSD)

- **Convergence test:**

If $\overline{\mathbf{r}}^{(k)} = \mathbf{0}$, then terminate $\to \boldsymbol{\lambda}^{(k)}$ solves [**RMP**].

- **Line search:**

$$\text{Let} \quad l_{\max} = \min_{(p,q)\in\mathcal{C}} \left\{ \min_{i\in\hat{\mathcal{R}}_{pq}} \left\{ -\frac{\lambda_{pqi}^{(k)}}{\overline{r}_{pqi}^{(k)}} \; : \; \overline{r}_{pqi} < 0 \right\} \right\}.$$

Solve the line search problem

$$[\mathbf{LS}] \qquad \min_{l\in[0,l_{\max}]} \overline{T}(l) = T\left(\boldsymbol{\lambda}^{(k)} + l \cdot \overline{\mathbf{r}}^{(k)}\right),$$

- **New iteration point and convergence test:**

Let $\boldsymbol{\lambda}^{(k+1)} = \boldsymbol{\lambda}^{(k)} + l^{(k)} \cdot \overline{\mathbf{r}}^{(k)}$.

Contents

User Equilibrium / UE (利用者均衡配分)

Frank-Wolfe Algorithm

↓

Simplicial Decomposition
(SD)

↓

Disaggregate Simplicial Decomposition
(DSD)

Numerical experiments

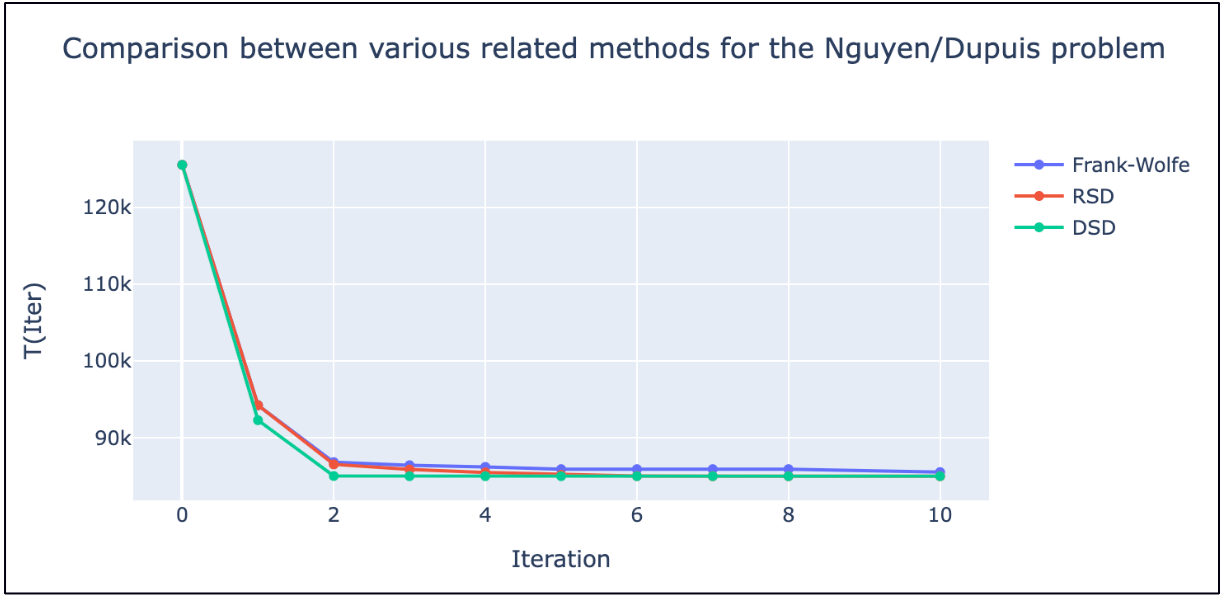
- FORTRAN-77 on a SUN 4/390 computer
  - CPU: Sun 4300 (25 **M**Hz)
  - RAM: Max 224 MB
  - (現在は2〜4**G**Hz, 4〜16GBが家庭用PCでは普通)
- Shortest route / 最短経路探索:

  Dijkstra's algorithm
- Line search / 直線探索:

  Armijo-type / アルミホ条件

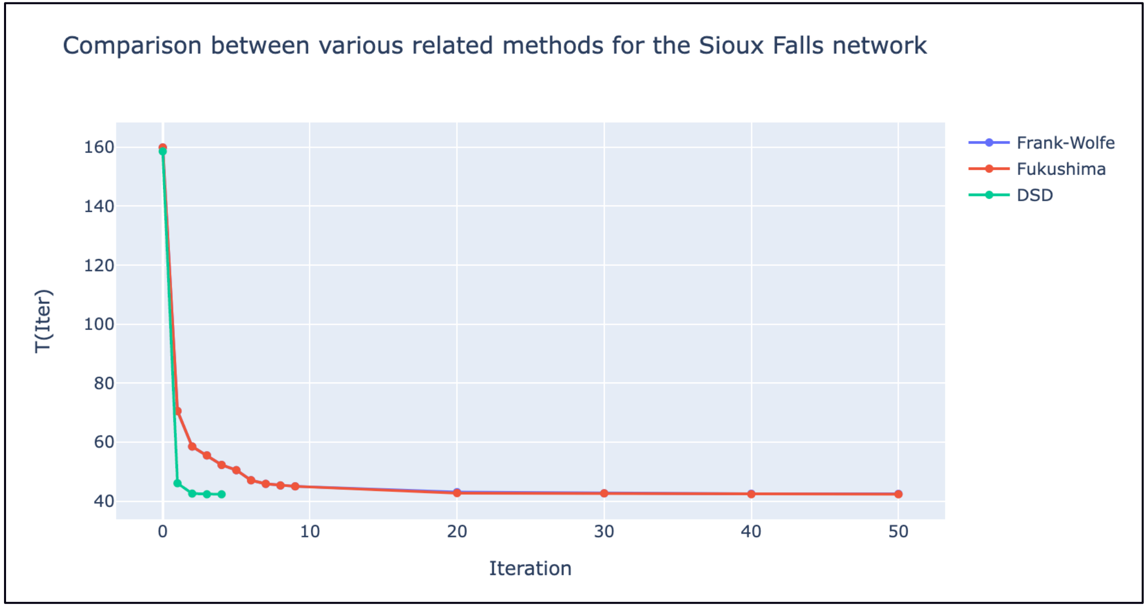
Numerical experiments

- Nguyen/Dupuis Problem
  - 19 arcs, 13 nodes and 4 commodities
  - CPU time: 0.15s



Numerical experiments

- Sioux Falls network
  - 76 arcs, 24 nodes and 528 commodities
  - CPU time: 7.04s



South Dakota

Numerical experiments

Sioux-Falls Network

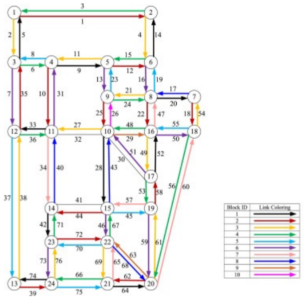
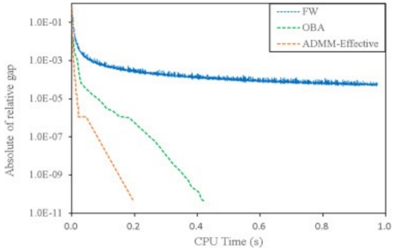


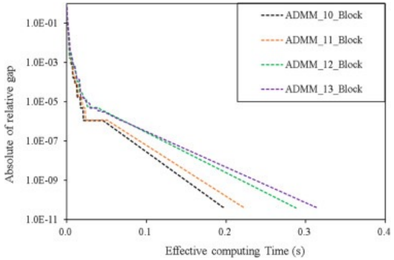
Table 5. Link blocking pattern.

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

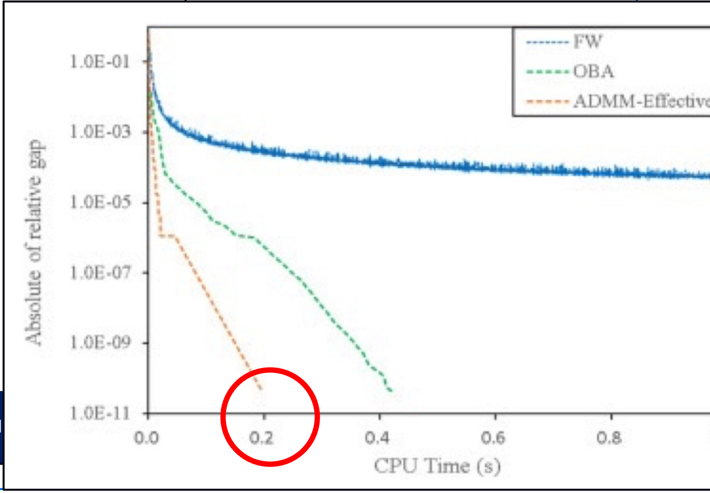
South Dakota



ADMM is better than FW, OBA



Fewer blocks, Faster convergence.



cf. 理論談話会#5 by 古橋さん

“An alternating direction method of multipliers for solving user equilibrium problem”

Contents

User Equilibrium / UE (利用者均衡配分)

Frank-Wolfe Algorithm

↓

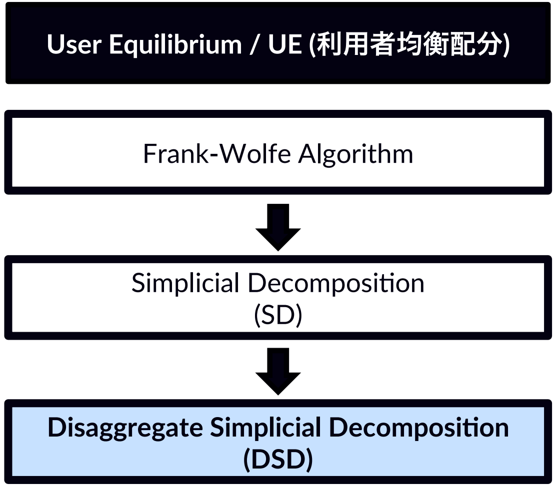
Simplicial Decomposition
(SD)

↓

Disaggregate Simplicial Decomposition
(DSD)

Conclusion

- DSDはFrank-Wolfe, SDと比較して計算効率が同等か上回っている。大規模ネットワークにおいてより効果的。
- 最短経路探索の回数が混雑していない状況では少なくて済む。
- リンクパフォーマンス関数の変化、需要、ネットワーク変化に応じて再配分が容易に行える。
- 今後は他の非線形問題へのDSDの適用、および実装の改善について研究を続ける。



所感

- 昔の論文をわざわざ自分で読む機会がないので良い機会だった
  - なかなかPDFにたどり着かず少し面倒だった
- Frank-Wolfe法は実装のシンプルさゆえに現代でも使われている
- コンピュータの処理速度の向上も飛躍的 (MHz→GHz)