

格子状の動的フローネットワークにおける避難施設配置問題

西井 彩乃¹ 照山 順一² 戸國 友貴² 東川 雄哉²

概要：動的フローネットワークは、各頂点に避難者数を示す供給量、各辺に容量と移動時間が与えられた有向グラフで定義される。動的フローネットワークの頂点上あるいは辺上に避難施設を配置すると、すべての避難者が避難施設に到着可能な最小時間、すなわち避難完了時間が定義できる。避難施設配置問題とは、避難完了時間を最小化する施設の配置を求める問題である。この問題については、これまでパス、木、サイクルなど限定されたネットワークに対してのみ多項式時間アルゴリズムが知られているが、より複雑なネットワークに対しては多項式時間アルゴリズムが知られていない。本稿では、辺の容量と移動時間が一定である格子状のネットワークについて、単一施設配置問題が多項式時間で解けることを示した。

キーワード：施設配置問題、多項式時間アルゴリズム、動的フローネットワーク

1. 序論

近年、地震などの自然災害が頻繁に発生しており、このような大規模災害に対して、堤防の設置などのハード中心の対策では対応しきれない現状にある。災害対策基本法の改定などにより、避難施設の整備が進み、これまで遅々として進まなかった津波避難タワーの設置や津波避難ビルの指定が加速化しており、避難施設の整備・避難路の確保・避難訓練等の事前準備がますます重要となっている。一方、津波・洪水・原発事故など、緊急を要する大規模災害に対する避難計画を策定する場合、考慮すべき重要な問題として交通渋滞による避難時間の遅延が挙げられる。実際、東日本大震災においては、交通渋滞に起因する逃げ遅れにより多くの犠牲者が出たことが知られている [1]。

この問題を解消するために、Ford and Fulkerson [2] によって提案された動的フローネットワークモデルを避難計画問題に適用することが考えられる。動的フローネットワークモデルでは、時間に対する人や物の動きを扱うことができるので、交通渋滞を定量的に加味した避難計画の策定が可能となる。動的フローネットワークは有向グラフを用いて定義され、グラフの頂点上には避難者の存在する地点を表す供給点と避難施設を表す需要点、さらに、各供給点に対して避難者数を表す供給量と各需要点に対して収容可能人数を表す需要量が与えられる。

動的フローネットワークにおける最も基本的な問題の一つとして最速輸送問題がある。この問題の目的は、すべて

の避難者が避難施設に到着可能な最小時間、すなわち避難完了時間を計算すると同時に、避難完了時間を達成する最適な避難者の流れを求めることである。最速輸送問題に対しては [3], [4], [5] などによって多項式時間アルゴリズムが与えられている。

本研究では、避難完了時間を最小化する施設配置問題を考える。これまで、パス [6], [7], [8], サイクル [8], 木 [9], [10], [11] などの限定されたネットワークにおける施設配置問題に対しては多項式時間アルゴリズムが提案されているが、その他のネットワーククラスについては多項式時間アルゴリズムが知られていない。そこで本研究では、より複雑なクラスである格子状のネットワークを対象として施設配置問題を扱う。格子状のネットワークは、既存研究で扱われてるネットワークに比べ、より実際の道路網に近い形状であると言える。本研究では、辺の容量と移動時間が一定である格子状の動的フローネットワークにおける単一施設配置問題に対し、初得多項式時間アルゴリズムを与える。

以下に本研究のアプローチを述べる。ネットワーク上で施設配置をおこなう対象として、頂点上または辺上が考えられる。ある頂点に施設を配置する場合、その施設に対する避難完了時間は、既存の最速輸送問題に対するアルゴリズム [5] を用いて多項式時間で計算できる。したがって、各辺において、避難完了時間を最小化する施設配置が多項式時間で計算できれば、ネットワーク全体における最適施設配置が多項式時間で求まる。以下ではネットワーク上の特定の辺における最適施設配置問題を扱う。

¹ 兵庫県立大学 社会情報学部社会情報科学科

² 兵庫県立大学大学院 情報科学研究科

2. 準備

2.1 問題のモデル

\mathbb{R} で実数の集合, \mathbb{R}_+ で非負の実数の集合を表す. 動的フローネットワークは, 組 $(G = (V, E), S^+, S^-, w, c, \tau)$ によって定義される. ここで, $G = (V, E)$ は頂点集合を V , 辺集合を E とする有向グラフである. また, 容量 $c: E \rightarrow \mathbb{R}_+$ は単位時間あたりに辺に入ることができるフローの上限を表し, 移動時間 $\tau: E \rightarrow \mathbb{R}_+$ は辺を移動するのにかかる時間を表す. $S^+ \subseteq V$ は供給点集合, $S^- \subseteq V$ は需要点集合を表す. $w: S^+ \cup S^- \rightarrow \mathbb{R}$ は頂点上にいる供給量または需要量を表す関数であり, $s \in S^+$ に対し $w(s) \geq 0$ は供給量を, $s \in S^-$ に対し $w(s) \leq 0$ は需要量を表す. また, 部分集合 $X \subseteq S^+ \cup S^-$ に対し $w(X) := \sum_{s \in X} w(s)$ とする.

動的フロー $f: E \times \mathbb{R}_+ \rightarrow \mathbb{R}_+$ は, ある辺にある時刻に流入する量を表す関数である. すなわち, 任意の辺 $e \in E$ と時刻 $\theta \in \mathbb{R}_+$ に対し, $f(e, \theta)$ は時刻 θ に辺 e に入る流量を表す. 本研究では連続モデルの動的フロー [12], [13] を取り扱う.

本研究では, 各頂点が平面上的格子点に配置され, 隣り合う頂点間に双方向の辺が存在する動的フローネットワークを入力とし, 単一需要点の最適配置を決定する問題を考える. さらに, すべての辺の容量および移動時間を一定とし, それぞれ非負の実数 c, τ で表す. また, すべての頂点を供給点とする. したがって, 問題の入力は, 需要点集合を持たない格子状の動的フローネットワーク $\mathcal{N} = (G = (V, E), S^+ (= V), \emptyset, w, c, \tau)$ となる. ここで, \mathcal{N} において隣り合う 2 頂点 u, v 間に避難施設が配置されたネットワークを定義する. 2 頂点 u, v 間において, u からの移動時間 y ($0 < y < \tau$) の場所に避難施設を配置した動的フローネットワークは, \mathcal{N} に以下の操作を施すことで得られる.

- (1) 有向辺 (u, v) , (v, u) を除く.
- (2) 需要量 $w(s^-) = -\sum_{v \in V} w(v)$ をもつ需要点 s^- を追加する.
- (3) 容量 c , 移動時間 y の有向辺 (u, s^-) および容量 c , 移動時間 $\tau - y$ の有向辺 (v, s^-) を追加する.

上記の操作で得られる動的フローネットワークを $\mathcal{N}_{uv}(y)$ と表す. 以下では, $\mathcal{N}_{uv}(y)$ における頂点集合および辺集合をそれぞれ $V(y), E(y)$ と表す.

2.2 避難完了時間

動的フロー f が, 任意の辺 $e \in E(y)$ と時刻 $\theta \in \mathbb{R}_+$ に対して

$$0 \leq f(e, \theta) \leq c,$$

また, 任意の頂点 $v \in V(y) \setminus \{s^-\} (= S^+)$ と時刻 $\theta \in \mathbb{R}_+$ に対して

$$\int_0^\theta \left(\sum_{(v,u) \in E(y)} f((v,u), t) - \sum_{(u,v) \in E(y)} f((u,v), t - \tau) \right) dt \leq w(v)$$

を満たし, さらに, 任意の頂点 $v \in V(y)$ に対して

$$\int_0^T \left(\sum_{(v,u) \in E(y)} f((v,u), t) - \sum_{(u,v) \in E(y)} f((u,v), t - \tau) \right) dt = w(v)$$

を満たす時刻 $T \in \mathbb{R}_+$ が存在するとき, 時刻 T について **実行可能な動的フロー** と呼ぶ. **避難完了時間** とは, 実行可能な動的フローが存在する最小の時刻を表す. 動的フローネットワーク $\mathcal{N}_{uv}(y)$ における避難完了時間を $\Theta^*(y)$ と表すとき, u, v 間における最適施設配置問題は以下の形で定式化される:

$$\begin{aligned} \text{(P1)} \quad & \text{minimize} && \Theta^*(y) \\ & \text{subject to} && 0 < y < \tau \end{aligned}$$

次に, 動的フローネットワークにおける避難完了時間の性質について述べる. 任意の供給点集合 $X \subseteq S^+$ について, $\mathcal{N}_{uv}(y)$ 上で時刻 0 から θ の期間に X から需要点 s^- へ送ることができる動的フローの量の最大値を $o_\theta(X, y)$ と表記する. このとき, 以下の定理が知られている.

定理 1. [3] 動的フローネットワーク $\mathcal{N}_{uv}(y)$ において, 時刻 θ について実行可能な動的フローが存在するための必要十分条件は, すべての頂点集合 $X \subseteq S^+$ に対して, $o_\theta(X, y) \geq w(X)$ が成り立つことである.

定理 1 を言い換えると, 時刻 θ について実行可能な動的フローが存在するための必要十分条件は

$$\min\{o_\theta(X, y) - w(X) \mid X \subseteq S^+\} \geq 0 \quad (1)$$

が成り立つことである. したがって, $\mathcal{N}_{uv}(y)$ における避難完了時間 $\Theta^*(y)$ は, 式 (1) を満たす最小の θ となる. ここで,

$$\Theta(X, y) := \min\{\theta \mid o_\theta(X, y) - w(X) \geq 0\} \quad (2)$$

と定義すると, $o_\theta(X, y) - w(X)$ が θ について単調非減少であることより,

$$\Theta^*(y) = \max\{\Theta(X, y) \mid X \subseteq S^+\} \quad (3)$$

が成り立つ.

2.3 2次元線分集合の包絡線

\mathcal{F} をそれぞれ異なる閉区間 $[a_1, b_1], \dots, [a_n, b_n]$ 上で定義される一変数線形関数 f_1, \dots, f_n からなる族とする. 以下では, このような \mathcal{F} を単に **2次元線分集合** と呼ぶ. さらに, 閉区間 $[a, b]$ 上で定義される関数 $f \in \mathcal{F}$ について,

$$\underline{f}(x) := \begin{cases} f(x) & (x \in [a, b]), \\ -\infty & (x \notin [a, b]) \end{cases}$$

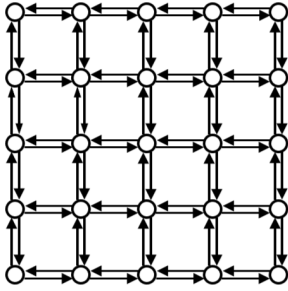


図1 入力のネットワーク \mathcal{N}

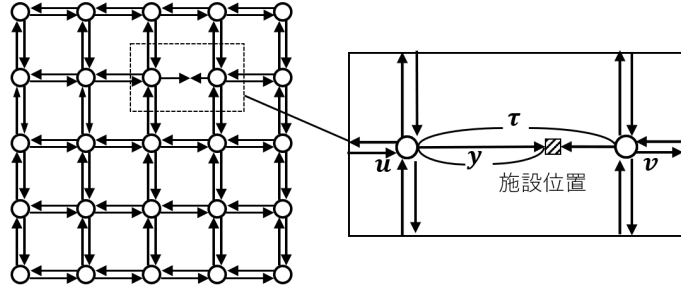


図2 $\mathcal{N}_{uv}(y)$

と定義するとき、2次元線分集合 \mathcal{F} の上側包絡線 $U_{\mathcal{F}}$ は次のように定義される:

$$U_{\mathcal{F}}(x) := \max_{1 \leq i \leq n} f_i(x) \quad \left(x \in \bigcup_{1 \leq i \leq n} [a_i, b_i] \right).$$

2次元線分集合 \mathcal{F} の上側包絡線 $U_{\mathcal{F}}$ において、構成する関数が切り替わる点、すなわち

$$\lim_{x \rightarrow p^+} \operatorname{argmax}_{1 \leq i \leq n} f_i(x) \neq \lim_{x \rightarrow p^-} \operatorname{argmax}_{1 \leq i \leq n} f_i(x)$$

を満たす点 $(p, U_{\mathcal{F}}(p))$ を $U_{\mathcal{F}}$ の折れ点と呼ぶ。このとき以下の定理が知られている。

定理 2. [14], [15] n 個の線分を含む 2次元線分集合 \mathcal{F} について、上側包絡線 $U_{\mathcal{F}}$ は高々 $O(n\alpha(n))$ 個の折れ点を持ち、それらは $O(n \log n)$ 時間で計算できる。ただし、関数 α はアッカーマン関数の逆関数である。

2次元線分集合 \mathcal{F} の上側包絡線 $U_{\mathcal{F}}$ において、隣接する折れ点間は線分で結ばれているため、折れ点のいずれかが上側包絡線 $U_{\mathcal{F}}$ の最小値を与える。よって、以下の系が与えられる。

系 3. n 個の線分を含む 2次元線分集合 \mathcal{F} について、上側包絡線 $U_{\mathcal{F}}$ の最小値をとる点を $O(n \log n)$ 時間で計算できる。

3. 特定の边上における施設配置問題

本節では、問題 (P1) に対する多項式時間アルゴリズムを与える。以下では、ネットワークの頂点数 $|V|$ を n で表す。式 (3) より、すべての $X \subseteq S^+$ に対して y の関数 $\Theta(X, y)$ が求まれば、それらの $0 < y < \tau$ における上側包絡線として目的関数 $\Theta^*(y)$ が表されるが、このような総当たりの方法は多項式時間アルゴリズムを与えない。本研究では、任意の y ($0 < y < \tau$) に対し $\Theta(X, y)$ を最大化する X が、 y に依存しない $O(\sqrt{n})$ サイズの供給点集合族に含まれることを示し、それに基づく多項式時間アルゴリズムを与える。3.1 節において関数 $\Theta(X, y)$ の性質を示し、3.2 節において $\Theta(X, y)$ を最大化する X の性質を示す。3.3 節において問題 (P1) に対する多項式時間アルゴリズムを与える。

3.1 関数 $\Theta(X, y)$ の性質

まず、以下の議論で用いる表記を導入する。任意の 2 供給点 $p, q \in S^+$ について、 p から q への $\mathcal{N}_{uv}(y)$ 上での最短到達可能時間を $d_{uv}(p, q)$ と定義する。このとき、 $d_{uv}(p, q)$ に対応するパスは辺 $(u, y), (v, y) \in E(y)$ を用いないことに注意する。また、任意の供給点集合 $X \subseteq S^+$ および供給点 $p \in S^+$ について、 X から p への $\mathcal{N}_{uv}(y)$ 上での最短到達可能時間を $d_{uv}(X, p)$ と定義する。すなわち、以下を満たす:

$$d_{uv}(X, p) := \min\{d_{uv}(x, p) \mid x \in X\}. \quad (4)$$

さらに、 X から u または v を経由して s^- に至る $\mathcal{N}_{uv}(y)$ 上での最短到達可能時間のうち、最小のものを $d_1(X, y)$ 、最大のものを $d_2(X, y)$ と定義する。すなわち、以下を満たす:

$$\begin{aligned} d_1(X, y) &:= \min\{d_{uv}(X, u) + y, d_{uv}(X, v) + \tau - y\}, \\ d_2(X, y) &:= \max\{d_{uv}(X, u) + y, d_{uv}(X, v) + \tau - y\}. \end{aligned} \quad (5)$$

以下に、 $d_1(X, y), d_2(X, y)$ に関する 2 つの補題を示す (証明は省略)。

補題 4. 任意の供給点集合 $X \subseteq S^+$ について、 $d_1(X, y)$ と $d_2(X, y)$ に対応するパスの対のうち、 $\mathcal{N}_{uv}(y)$ 上で辺素なものが存在する。

補題 5. すべての頂点集合 $X \subseteq S^+$ 、 $0 < y < \tau$ に対して、 $d_1(X, y), d_2(X, y)$ は以下を満たす:

$$(1) \quad d_{uv}(X, u) - d_{uv}(X, v) < 0 \text{ または、} d_{uv}(X, u) - d_{uv}(X, v) = 0 \text{ かつ } y \leq \frac{\tau}{2} \text{ のとき、}$$

$$\begin{cases} d_1(X, y) = d_{uv}(X, u) + y, \\ d_2(X, y) = d_{uv}(X, v) + \tau - y. \end{cases}$$

$$(2) \quad d_{uv}(X, u) - d_{uv}(X, v) > 0 \text{ または、} d_{uv}(X, u) - d_{uv}(X, v) = 0 \text{ かつ } y > \frac{\tau}{2} \text{ のとき、}$$

$$\begin{cases} d_1(X, y) = d_{uv}(X, v) + \tau - y, \\ d_2(X, y) = d_{uv}(X, u) + y. \end{cases}$$

補題 4 より、 $o_{\theta}(X, y)$ は、 $d_1(X, y)$ および $d_2(X, y)$ によ

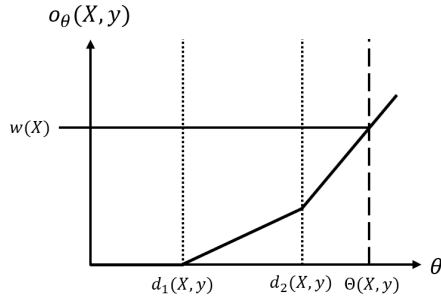


図3 $o_\theta(X, y)$ の関数

て定まり、 $o_\theta(X, y)$ を θ の関数とみると、以下の3つの区間からなる関数である (図3) :

$$o_\theta(X, y) := \begin{cases} 0 & 0 \leq \theta \leq d_1(X, y), \\ c(\theta - d_1(X, y)) & d_1(X, y) \leq \theta \leq d_2(X, y), \\ c(2\theta - d_1(X, y) - d_2(X, y)) & d_2(X, y) \leq \theta. \end{cases}$$

$o_\theta(X, y)$ は、傾きが単調非減少な区分線形関数であるので、それぞれの区分における式を用いて、以下のように表すことができる。

$$o_\theta(X, y) = \max\{0, c(\theta - d_1(X, y)), c(2\theta - d_1(X, y) - d_2(X, y))\} \quad (6)$$

ここで、式(5)を用いると、

$$\begin{aligned} c(\theta - d_1(X, y)) &= c(\theta - \min\{d_{uv}(X, u) + y, d_{uv}(X, v) + \tau - y\}) \\ &= \max\{c(\theta - d_{uv}(X, u) - y), c(\theta - d_{uv}(X, v) - \tau + y)\} \end{aligned} \quad (7)$$

となる。また、

$$d_1(X, y) + d_2(X, y) = d_{uv}(X, u) + d_{uv}(X, v) + \tau$$

であるから、

$$\begin{aligned} c(2\theta - d_1(X, y) - d_2(X, y)) &= c(2\theta - d_{uv}(X, u) - d_{uv}(X, v) - \tau) \end{aligned} \quad (8)$$

が成り立つ。式(7),(8)より、式(6)は以下の式に変形できる。

$$o_\theta(X, y) = \max\{0, c(\theta - d_{uv}(X, u) - y), c(\theta - d_{uv}(X, v) - \tau + y), c(2\theta - d_{uv}(X, u) - d_{uv}(X, v) - \tau)\}. \quad (9)$$

以下では、式(9)を用いて、 $\Theta(X, y)$ の性質を与える。式(2)より、 $o_\theta(X, y) = w(X)$ となる θ の値が $\Theta(X, y)$ となる。 $\Theta(X, y)$ を y に関する一変数関数とみなしたとき、以下の定理が成り立つ。

定理 6. 任意の $X \subseteq S^+$ に対して、 $\Theta(X, y)$ は $0 < y < \tau$

において、高々3つの区分をもつ区分線形関数であり、以下の式で表される：

$$\Theta(X, y) = \min \left\{ y + \frac{w(X)}{c} + d_{uv}(X, u), -y + \tau + \frac{w(X)}{c} + d_{uv}(X, v), \frac{w(X)}{2c} + \frac{d_{uv}(X, u) + d_{uv}(X, v) + \tau}{2} \right\}.$$

証明. 式(2)より、 $o_\theta(X, y)$ を構成する線形関数 $c(\theta - d_{uv}(X, u) - y)$, $c(\theta - d_{uv}(X, v) - \tau + y)$, $c(2\theta - d_{uv}(X, u) - d_{uv}(X, v) - \tau)$ それぞれが $w(X)$ となる θ の値のうち、最小のものが $\Theta(X, y)$ となる。 $c(\theta - d_{uv}(X, u) - y) = w(X)$ を満たす θ の値は、 $\theta = y + \frac{w(X)}{c} + d_{uv}(X, u)$ である。 $c(\theta - d_{uv}(X, v) - \tau + y) = w(X)$ を満たす θ の値は、 $\theta = -y + \tau + \frac{w(X)}{c} + d_{uv}(X, v)$ である。 $c(2\theta - d_{uv}(X, u) - d_{uv}(X, v) - \tau) = w(X)$ を満たす θ の値は、 $\theta = \frac{w(X)}{2c} + \frac{d_{uv}(X, u) + d_{uv}(X, v) + \tau}{2}$ である。 よって、これら3つの値の最小値が $\Theta(X, y)$ である。 また、いずれも定数関数または傾き ± 1 の線形関数であるから、 $\Theta(X, y)$ は、高々3つの区分からなる区分線形関数である。 \square

定理6より、 $d_{uv}(X, u)$ と $d_{uv}(X, v)$ の大小関係を考慮すると、 $\Theta(X, y)$ の概形は図4のようになる。

3.2 支配的な供給点集合族

2つの集合 X, X' について、 $\Theta(X, y) \geq \Theta(X', y)$ がすべての y について成り立つとき、 X が X' を**支配する**という。ある集合 X がいかなる集合にも支配されないとき、 X は**支配的な供給点集合**であるという。ここで、支配的な供給点集合 X の性質を考える。2つの供給点集合 X, X' について、 $d_{uv}(X, u) = d_{uv}(X', u)$ かつ $d_{uv}(X, v) = d_{uv}(X', v)$ のとき、式(9)より $o_\theta(X, y)$ と $o_\theta(X', y)$ は等しい関数である。また、式(2)より、 $w(X) > w(X')$ ならば $\Theta(X, y) > \Theta(X', y)$ である。よって、一組の $(d_{uv}(X, u), d_{uv}(X, v))$ に対して、 $w(X)$ が最大となる頂点集合が支配的な供給点集合である。 $w(X)$ を最大とする頂点集合は、頂点 u への最短到達可能時間が $d_{uv}(X, u)$ 以上である頂点、かつ、頂点 v への最短到達可能時間が $d_{uv}(X, v)$ 以上である頂点をすべて含む頂点集合である。2つの整数 i, j に対して、 $X_{i,j}$ を2頂点 u, v それぞれへの最短到達時間が $i\tau, j\tau$ 以上である頂点からなる頂点集合と定義する。すなわち、 $X_{i,j}$ は以下の式を満たす。

$$X_{i,j} := \{x \in V \mid d_{uv}(x, u) \geq i\tau, d_{uv}(x, v) \geq j\tau\}$$

ここで、支配的な供給点集合 $X_{i,j}$ の族を \mathcal{X} とする。つまり、

$$\mathcal{X} := \{X_{i,j} \mid 0 \leq i \leq 2\sqrt{n}, 0 \leq j \leq 2\sqrt{n}\} \quad (10)$$

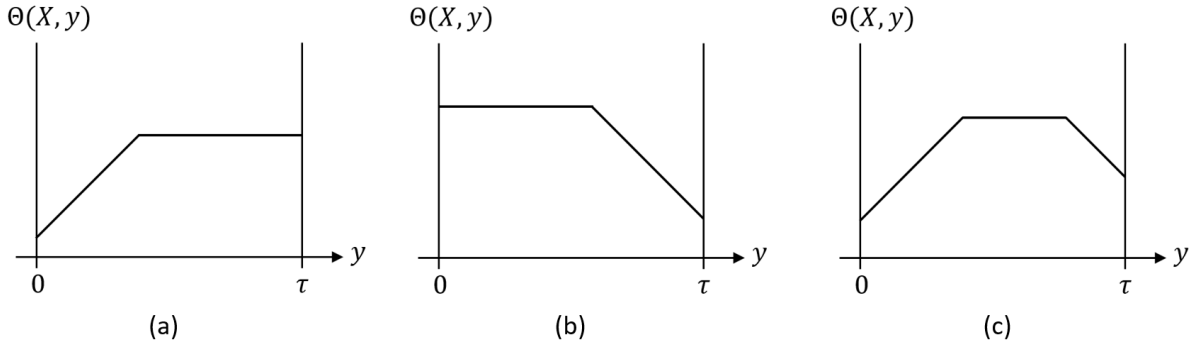


図 4 $0 < y < \tau$ における $\Theta(X, y)$ の概形. 3つの図はそれぞれ (a) $d_{uv}(X, u) < d_{uv}(X, v)$ の場合, (b) $d_{uv}(X, u) > d_{uv}(X, v)$ の場合, (c) $d_{uv}(X, u) = d_{uv}(X, v)$ の場合を示す.

である. $d_{uv}(X, u), d_{uv}(X, v)$ はともに τ の整数倍であることに注意すると, 族 \mathcal{X} はすべての支配的な供給点集合を含み, 以下の式が成り立つ.

$$\operatorname{argmax}\{\Theta(X, y) \mid X \subseteq S^+\} \cap \mathcal{X} \neq \emptyset. \quad (11)$$

式 (10) より, \mathcal{X} に含まれる集合の数が $O(n)$ であることがわかる. 以下では, \mathcal{X} に含まれる集合の数の上界をより厳密に与える. そのために, $(d_{uv}(X, u), d_{uv}(X, v))$ の組合せに関する以下の補題を示す (証明は省略).

補題 7. すべての頂点集合 $X \subseteq S^+$ に対して, 以下の式が成り立つ:

$$d_{uv}(X, u) - d_{uv}(X, v) = \pm \tau i (i = 0, 1, 2, 3).$$

$d_{uv}(X, u), d_{uv}(X, v)$ の最大値は, ネットワークが格子状であることから, 高々 $2\tau\sqrt{n}$ である. 補題 7 より, $d_{uv}(X, u), d_{uv}(X, v)$ のうち, 一方の値が決まれば, もう一方は高々 7 通りに絞られる. よって, 以下の定理が得られる.

定理 8. 支配的な供給点集合の族 \mathcal{X} に含まれる集合の数は $O(\sqrt{n})$ である.

3.3 アルゴリズム

我々のアルゴリズムは大きく 3つのステップからなる. まず, すべての支配的供給点集合 $X_{i,j}$ について, $w(X_{i,j})$ を求める. 次に, $\Theta(X_{i,j}, y)$ を構成する線分を求める. 最後に, 関数族 $\{\Theta(X_{i,j}, y) \mid X_{i,j} \in \mathcal{X}\}$ の上側包絡線の最小値を与える y を求める.

まず, すべての $X_{i,j} \in \mathcal{X}$ に対する $w(X_{i,j})$ の求め方を与える. 頂点数が n であることから, 任意の i, j に対して, $w(X_{i,j})$ を $O(n)$ 時間で求めることができる. また, 定理 8 より, $|\mathcal{X}| = O(\sqrt{n})$ であることから, すべての $w(X_{i,j})$ を求める時間は, $O(n\sqrt{n})$ 時間であることが自明に導かれる. 以下では, $w(X_{i,j})$ をより効率よく計算する方法を与える.

補題 9. すべての $w(X_{i,j})$ を $O(n)$ 時間で計算することができる.

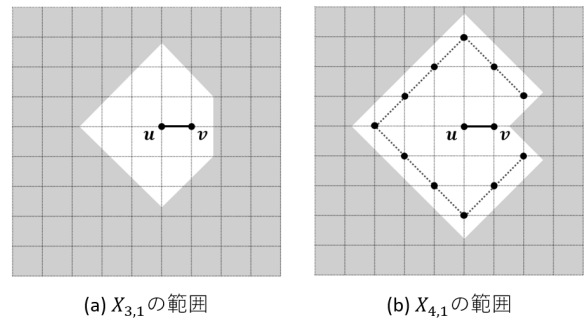


図 5 $X_{3,1}$ と $X_{4,1}$ の範囲. それぞれ灰色の範囲内の頂点からなる. 図 (b) の点線上にある頂点が $X_{3,1}$ と $X_{4,1}$ の差分である.

証明. 動的計画法による計算方法を与える. まず, $w(X_{0,0})$, すなわち, 全頂点の供給量の合計を $O(n)$ 時間で求める.

$i \neq 0$ または $j \neq 0$ である組 (i, j) に対して, $w(X_{i,j})$ を求める方法を, 2つの場合に分けて与える.

(Case 1: $i - j = 3$, または, $i = 1, 2$ かつ $j = 0$ の場合) $X_{i,j}$ と $X_{i-1,j}$ の関係に注目すると, $X_{i,j} \subset X_{i-1,j}$ であり, $X_{i-1,j} \setminus X_{i,j}$ は頂点 u からの最短到達可能時間が $(i-1)\tau$ である頂点からなる (図 5). よって, $w(X_{i-1,j})$ から, 頂点 u からの最短到達可能時間が $(i-1)\tau$ である頂点の供給量を除けばよい. 除くべき頂点数は高々 $2\sqrt{n}$ 個であるので, $O(\sqrt{n})$ 時間で計算可能である. 以上から, $w(X_{i,j})$ は $w(X_{i-1,j})$ の値を用いると $O(\sqrt{n})$ 時間で求まる.

(Case 2: その他の場合) $X_{i,j}$ と $X_{i,j-1}$ の関係に注目すると, $X_{i,j} \subset X_{i,j-1}$ であり, $X_{i,j-1} \setminus X_{i,j}$ は頂点 v からの最短到達可能時間が $(j-1)\tau$ である頂点からなる. Case 1 と同様に, $w(X_{i,j-1})$ から頂点 v からの最短到達可能時間が $(j-1)\tau$ である頂点の供給量を除けばよく, $w(X_{i,j})$ は $w(X_{i,j-1})$ の値を用いると $O(\sqrt{n})$ 時間で求まる.

小さい i, j から順に計算することで, $w(X_{0,0})$ 以外の $w(X_{i,j})$ は, それぞれ $O(\sqrt{n})$ 時間で計算可能である. 族 \mathcal{X} に含まれる $X_{i,j}$ の数は $O(\sqrt{n})$ であるから, 全体の計算時間は $O(n) + O(\sqrt{n}) \times O(\sqrt{n}) = O(n)$ 時間となる. \square

次に、 $\Theta(X_{i,j}, y)$ を構成する線分を求める。上で求めた $w(X_{i,j})$ と定理 6 より、 $\Theta(X_{i,j}, y)$ は、高々 3 つの線分からなる区分線形関数であり、以下の式で与えられる。

$$\Theta(X_{i,j}, y) = \min \left\{ y + \frac{w(X_{i,j})}{c} + i\tau, \right. \\ \left. -y + \tau + \frac{w(X_{i,j})}{c} + j\tau, \right. \\ \left. \frac{w(X_{i,j})}{2c} + \frac{i+j+1}{2}\tau \right\} \quad (12)$$

式 (3) より、関数族 $\{\Theta(X_{i,j}, y) \mid X_{i,j} \in \mathcal{X}\}$ の上側包絡線の最小値が最小の $\Theta^*(y)$ である。 $\Theta(X_{i,j}, y)$ は、式 (12) で与えられるように、高々 3 つの線分からなる。また、定理 8 より、 $|\mathcal{X}| = O(\sqrt{n})$ であるから、関数族 $\{\Theta(X_{i,j}, y) \mid X_{i,j} \in \mathcal{X}\}$ は $O(\sqrt{n})$ 個の線分によって定まる。すなわち、関数族 $\{\Theta(X_{i,j}, y) \mid X_{i,j} \in \mathcal{X}\}$ の上側包絡線は $O(\sqrt{n})$ 個の線分を含む 2 次元線分集合の上側包絡線である。よって、系 3 より、 $O(\sqrt{n})$ 個の線分を含む 2 次元線分集合の上側包絡線の最小値を与える y は $O(\sqrt{n} \log n)$ 時間で求まる。

これまでの議論をまとめると、すべての $X \in \mathcal{X}$ の $w(X)$ を計算するのに、 $O(n)$ 時間かかり、関数族 $\{\Theta(X, y) \mid X \in \mathcal{X}\}$ の上側包絡線の最小値を与える y を求めるのに $O(\sqrt{n} \log n)$ 時間かかる。よって、以下の定理が得られる。

定理 10. 格子状の動的ネットワークの特定の辺上に施設を配置する施設配置問題は、 $O(n)$ の計算時間で求まる。

施設配置候補点として、すべての辺上を対象とする問題を考える。この問題は、各辺ごとに問題 (P1) を解けばよく、辺の数は $O(n)$ であること、および定理 10 より、以下の系が得られる。

系 11. 格子状の動的ネットワークのすべての辺上の中で最適な施設配置は、 $O(n^2)$ 時間で計算可能。

次に、施設配置をおこなう対象が、すべての頂点上およびすべての辺上である問題を考える。施設が頂点上にある場合の避難完了時間を計算する現在最速のアルゴリズムは、Schlöter ら [5] によって提案された $\tilde{O}(m^2k^4 + m^2nk)$ 時間アルゴリズムである。ここで、 m は辺数、 k は需要点および供給点の総和、 \tilde{O} は Big O 記法から対数因子を省略したものである。これを本研究に適用すると、各頂点につき、 $\tilde{O}(n^6)$ 時間で避難完了時間を計算することができる。施設配置対象を頂点上のみとした場合、このアルゴリズムを各頂点ごとに適用することで、最適な配置点は $\tilde{O}(n^7)$ 時間で求めることができる。この結果と系 11 を合わせることで、以下の系が得られる。

系 12. 格子状の動的フローネットワークにおいて、避難完了時間を最小化する単一施設配置問題は、 $\tilde{O}(n^7)$ 時間で計算可能。

参考文献

- [1] 金城敬太, 松本行真: 東日本大震災の調査をもとにした避難行動についての個人や社会からの影響の分析, 日本都市学会年報, Vol. 45, pp. 104–112 (2011).
- [2] Ford Jr, L. R. and Fulkerson, D. R.: Constructing maximal dynamic flows from static flows, *Operations research*, Vol. 6, No. 3, pp. 419–433 (1958).
- [3] Hoppe, B. and Tardos, É.: The quickest transshipment problem, *Mathematics of Operations Research*, Vol. 25, No. 1, pp. 36–62 (2000).
- [4] Kamiyama, N.: Discrete Newton methods for the evacuation problem, *Theoretical Computer Science*, Vol. 795, pp. 510–519 (2019).
- [5] Schlöter, M., Skutella, M. and Van Tran, K.: A Faster Algorithm for Quickest Transshipments via an Extended Discrete Newton Method, *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, SIAM, pp. 90–102 (2022).
- [6] Higashikawa, Y., Golin, M. J. and Katoh, N.: Multiple sink location problems in dynamic path networks, *Theoretical Computer Science*, Vol. 607, pp. 2–15 (2015).
- [7] Bhattacharya, B., Golin, M. J., Higashikawa, Y., Kameda, T. and Katoh, N.: Improved algorithms for computing k-sink on dynamic flow path networks, *Workshop on Algorithms and Data Structures*, Springer, pp. 133–144 (2017).
- [8] Benkoczi, R., Bhattacharya, B., Higashikawa, Y., Kameda, T., Katoh, N. and Teruyama, J.: Locating Evacuation Centers Optimally in Path and Cycle Networks, *21st Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2021)*, Schloss Dagstuhl-Leibniz-Zentrum für Informatik (2021).
- [9] Mamada, S., Uno, T., Makino, K. and Fujishige, S.: An $O(n \log 2n)$ algorithm for the optimal sink location problem in dynamic tree networks, *Discrete Applied Mathematics*, Vol. 154, No. 16, pp. 2387–2401 (2006).
- [10] Chen, D. and Golin, M.: Sink evacuation on trees with dynamic confluent flows, *27th International Symposium on Algorithms and Computation (ISAAC 2016)*, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2016).
- [11] Higashikawa, Y., Golin, M. J. and Katoh, N.: Minimax regret sink location problem in dynamic tree networks with uniform capacity, *International Workshop on Algorithms and Computation*, Springer, pp. 125–137 (2014).
- [12] Fleischer, L. and Tardos, É.: Efficient continuous-time dynamic network flow algorithms, *Operations Research Letters*, Vol. 23, No. 3-5, pp. 71–80 (1998).
- [13] Higashikawa, Y. and Katoh, N.: A Survey on Facility Location Problems in Dynamic Flow Networks, *The Review of Socionetwork Strategies*, Vol. 13, No. 2, pp. 163–208 (2019).
- [14] Hart, S. and Sharir, M.: Nonlinearity of Davenport–Schinzel sequences and of generalized path compression schemes, *Combinatorica*, Vol. 6, No. 2, pp. 151–177 (1986).
- [15] Hershberger, J.: Finding the upper envelope of n line segments in $O(n \log n)$ time, *Information Processing Letters*, Vol. 33, No. 4, pp. 169–174 (1989).