

# 最速避難計画のモデリングと解法

加藤 直樹, 瀧澤 重志

最速避難計画問題は、動的ネットワークの頂点上に存在するすべての避難者ができるだけ早く避難所のいずれかに移動する計画を求める問題である。本稿では、この問題を高速に解くヒューリスティクスを紹介する。具体的に大阪市湾岸地域における津波からの最速避難計画に適用した例を紹介する。

キーワード：最速避難計画, 輸送問題, ネットワークフロー, 時間拡大ネットワーク

## 1. はじめに

21世紀に入り、地震、津波、火山噴火、台風の大規模な自然災害がたびたび発生するようになってきている。中でも、2011年に発生した東日本大震災は記憶に新しい。2015年において、今後30年間に南海トラフ地震が発生する確率が60~70%と言われており[1]、東海地方、紀伊半島、大阪湾、四国、九州の沿岸地域では巨大な津波に備える必要がある。東日本大震災の教訓として、防波堤、防潮堤などの物理的対策だけで津波に備えることには限界があり、避難計画の重要性が注目されており、これまで遅々として進まなかった津波避難ビルの指定が加速化している。避難計画の難しさの一つは、災害発生の不確定性である。災害発生場所、規模、時間帯などにより、人口分布に大きな違いがある。このため、事前に非常に多くのシナリオに対して、避難計画を立てておく必要があり、高速計算が求められる。また、住民の避難計画は、各自治体が立案するものなので、汎用のPCで手軽に計算できる必要がある。筆者らは、ネットワークフローモデルを用いた最速避難計画に関する研究をこの10年近くにわたって手がけてきた[2~5]。本稿では、ネットワークフローモデルを用いた最速避難計画のモデル化とアルゴリズムおよび適用例について紹介する。

本稿では、ネットワークモデルに基づいて最速避難計画を求める際に、従来用いられてきた時間拡大ネットワークを用いることなく最速避難の近似解を計算するアルゴリズムを紹介する。この手法の計算時間は対象となるネットワークサイズのみに依存し計算機のメ

モリの使用量が小さいため、汎用のPCで計算可能であるという利点がある。また計算実験対象として南海トラフ地震による津波被害が予想される、大阪市湾岸地域を取り上げ、従来手法と提案手法の比較を行いその結果を示す。

### 1.1 既往研究

最速避難計画問題の解法はFord and Fulkerson [6, 7]によって、最初に提案されている。その方法は、以下で概説するように、動的ネットワークを静的ネットワークに変換する時間拡大ネットワークを用いて、与えられた時間内にネットワークの始点（ソース）から終点（シンク）まで流すことのできる最大の流量（動的最大フロー）を求めるものである。そのアルゴリズムを繰り返し用いて、最速避難完了時刻を求める。Kamiyama et al. [3]はさらに避難所の容量を考慮した場合の定式化を行った。

時間拡大ネットワークを用いない最速輸送アルゴリズムはHoppe and Tardos [8, 9]によって提案されている。それは多項式時間アルゴリズムであるが、劣モジュラ関数の最小化を繰り返し計算する必要がある。しかし、現時点では頂点数が1,000を超えるような劣モジュラ関数の最小化は実行時間での計算が困難であるため、大規模な問題に対して、効率的とは言えない。

### 1.2 最速避難計画問題の定義

以下に、動的ネットワーク、時間拡大ネットワークについて説明する。

#### 1.2.1 動的ネットワーク

標準的な（静的）ネットワークフロー問題はグラフ（無向グラフでも有向グラフでもよい） $G = (V, A)$ 上で定義される。ここで $V$ は頂点集合、 $A$ は辺集合を表わす。各辺 $a$ は正の容量 $c(a)$ をもっている。これは、辺を通過できる資源量の上限を与えている。最大フロー問題は、始点（ソース）と呼ばれる頂点と終点（シンク）と呼ばれる頂点が指定されたとき、容量制約

かとう なおき

関西学院大学理工学部, JST CREST

〒669-1337 兵庫県三田市学園2-1

たきざわ あつし

大阪市立大学工学研究科, JST CREST

〒558-8585 大阪府大阪市住吉区杉本3-3-138

の下で始点から終点へできるだけたくさんの資源を輸送する問題である。

動的フロー問題は、標準的な最大フロー問題と同様に、各辺  $a \in A$  に対して正の容量  $c(a)$  が与えられているグラフ  $G = (V, A)$  に対して定義される。各辺  $a = (u, v)$  には容量  $c(a)$  以外に、整数の移動時間  $\tau(a)$  が与えられている。つまり、時刻  $t_0$  に  $u$  を出発したフローは時刻  $t_0 + \tau(a)$  に  $v$  に到着する。容量  $c(a)$  は単位時間に  $a$  に入ることのできる資源量の上限を定めている。別の言い方をすると、容量は辺の幅を表しており、人が辺上を同時に移動できる人数の上限を与えていると考えてもよい。時刻  $t = 0$  において頂点  $u$  にいる  $w$  人が辺  $(u, v)$  を通って  $v$  に移動しようとしている。単位時間当たり  $c(a)$  人しか  $a$  に入れないので、 $\lceil w/c(a) \rceil$  グループが  $a$  を移動する（最後のグループを除いて各グループは  $c(a)$  人からなっている）。最初のグループは時刻  $\tau(a)$  に  $v$  に到着し、最後のグループは時刻  $\tau(a) + \lceil w/c(a) \rceil - 1$  に到達する。いま、 $d_1$  人が、頂点  $u$  から辺  $a_1 = (u, v)$ 、 $a_2 = (v, w)$  を通って、頂点  $w$  に移動しようとしている。同時に、 $v$  にも  $d_2$  人がいて、 $w$  に移動しようとしているものとする。すると、 $u$  を出発した人が  $v$  に到達したとき、もともと  $v$  にいた人が辺  $a_2 = (v, w)$  に入ることが完了してなくて  $v$  に何人かが残っていると、 $u$  から  $v$  に到達した人は、 $v$  で待たされる。ここで、渋滞（滞留）が発生するわけである。つまり、動的フローモデルは、渋滞による待ちを表現できることに注意されたい。

さて、一般的な動的フローを用いた最速避難計画問題を定義しよう。

避難計画問題では、動的ネットワーク  $\mathcal{N} = (D, c, \tau, b, S^+, S^-)$  が与えられている。ここで、 $D = (V, A)$  は頂点集合  $V$  と辺集合  $A$  からなる有向グラフである。頂点  $x$  に対して  $\delta_D^+(x)$ 、 $\delta_D^-(x)$  を各々、 $x$  から出ていく辺集合、 $x$  へ向かう辺集合と定義する。各辺  $a \in A$  には容量  $c(a)$  と移動時間  $\tau(a)$  が与えられている。また、正のサプライが存在する頂点集合  $S^+$  とシンク集合  $S^- \subseteq V$  が与えられている。本稿では、都市における避難を考えているので、頂点は建物や室を表わし、辺は道路や廊下を表している。サプライ  $b(x)$  ( $x \in S^+$ ) は  $x$  にいる住民の数を表している。このとき、最速避難計画問題は、すべてのサプライ（避難者）が  $S^-$  のいずれかのシンクに到達する時刻を最小にするような動的フロー（避難方法）を見つける問題である。この問題は、最速フロー問題とも呼ばれる。動的ネットワークの例を図1に示す。各頂点に付した数字

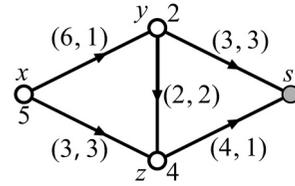


図1 動的ネットワークの例

はそのサプライを表している。辺に付された数字のペアは、左の数字は容量、右の数字は移動時間を表す。灰色の頂点がシンクである。

$\mathbb{R}_+$  と  $\mathbb{Z}_+$  は、各々、非負実数の集合、非負整数の集合を表すものとする。動的フロー  $f: A \times \mathbb{Z}_+ \rightarrow \mathbb{R}_+$  は次のように定義される。各辺  $a \in A$  と各時刻  $\theta \in \mathbb{Z}_+$  に対して、動的フロー  $f$  が実行可能とは容量制約

$$f(a, \theta) \leq c(a) \quad (\forall a \in A, \forall \theta \in \mathbb{Z}_+),$$

とフロー保存則

$$\sum_{a \in \delta_D^+(x)} \sum_{\theta=0}^{\Theta} f(a, \theta) \leq \sum_{a \in \delta_D^-(x)} \sum_{\theta=0}^{\Theta - \tau(a)} f(a, \theta) + b(x) \quad (\forall x \in V, \forall \Theta \in \mathbb{Z}_+),$$

を満たし、要求制約と呼ばれる次式を満たす  $\Theta^*$  が存在することである。

$$\sum_{s \in S^-} \sum_{a \in \delta_D^-(s)} \sum_{\theta=0}^{\Theta^* - \tau(a)} f(a, \theta) = \sum_{x \in V} b(x) \quad (1)$$

$f$  の避難完了時間を式(1)を満たす最小の時刻  $\Theta^*$  と定義する。このとき、最速避難計画問題とは、すべての実行可能動的フローの中で、避難完了時間を最小にするものを求める問題である。

実際の避難計画問題への応用を考えると、シンクは避難所を表しており、各シンクの受入れ可能な避難者数に上限がある。そこで、本稿では、シンクに対して容量関数  $l: S^- \rightarrow \mathbb{Z}_+$  が与えられているものとし、実行可能フロー  $f$  は次式を満たすものとする。

$$\sum_{a \in \delta_D^-(s)} \sum_{\theta=0}^{\Theta} f(a, \theta) \leq l(s) \quad (\forall s \in S^-, \forall \Theta \in \mathbb{Z}_+). \quad (2)$$

### 1.2.2 時間拡大ネットワーク

最速避難計画問題を解くために、Ford and Fulkeron [6, 7] は、時間拡大ネットワークを提案した。それは、時間制限  $\Theta$  を有する動的ネットワーク  $\mathcal{N}$  に対応する静的ネットワークで、 $\mathcal{N}(\Theta)$  と記す。 $\mathcal{N}(\Theta)$  の頂点集合は

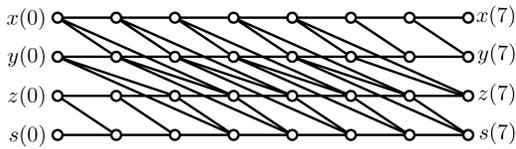


図2 図1の動的ネットワークに対する時間拡大ネットワーク. 各辺の向きは省略しているが, すべて左から右に向いている.

$$\{x(\theta) : x \in V, \theta \in \{0, \dots, \Theta\}\}$$

によって定義される. つまり, 元のネットワークの頂点  $x$  に対して, 各時刻  $\theta \in \{0, 1, \dots, \Theta\}$  に対応して, 頂点  $x(\theta)$  を設ける (図2).  $\mathcal{N}(\Theta)$  の辺集合は, 二つの部分からなっている. 一つ目は, 各辺  $a = (x, y) \in A$  と各  $\theta \in \{0, \dots, \Theta - \tau(a)\}$  に対して, 容量  $c(a)$  の辺  $a(\theta) = (x(\theta), y(\theta + \tau(a)))$  を設ける. 二つ目は, 各頂点  $x \in V$  と各  $\theta \in \{0, \dots, \Theta - 1\}$  に対して, 容量  $+\infty$  の辺  $(x(\theta), x(\theta + 1))$  からなるもので, 滞留辺と呼ばれる (図2の水平な辺). 各頂点  $x \in V$  に対して,  $x(0)$  のサプライは  $b(x)$  と定める.  $\theta \in \{1, \dots, \Theta\}$  に対する  $x(\theta)$  のサプライは0である.  $\mathcal{N}(\Theta)$  のシンク集合を  $\{s(\theta) \mid s \in S^-, \theta \in \{0, \dots, \Theta\}\}$  とする.

### 1.2.3 最速避難アルゴリズム

Ford and Fulkerson [6, 7] によって,  $\mathcal{N}$  における避難完了時間が  $\Theta$  以下であるための必要十分条件は  $\mathcal{N}(\Theta)$  において, ソース集合  $\{x(0) \mid x \in S^+\}$  からシンク集合へのサイズ  $\sum_{x \in S^+} b(x)$  のフローが存在することであると示された. そのような実行可能フローの存在は,  $\mathcal{N}(\Theta)$  の最大フローを求めることによって調べることができる. 時間拡大ネットワーク  $\mathcal{N}(\Theta)$  のサイズは  $\Theta$  に比例するので, 計算量は入力の変多項式サイズとなる. 本稿では, シンク容量を考慮するために, 時間拡大ネットワークを次のように修正する. まずスーパーシンク  $st$  を加え,  $s \in S^-$  に対して,  $s(\Theta)$  から  $st$  に容量  $l(s)$  の辺を設ける. すると,  $\mathcal{N}$  における避難完了時間が  $\Theta$  以下であるための必要十分条件は, このように修正した  $\mathcal{N}(\Theta)$  に, 上述の実行可能フローが存在することである.

このように時間拡大ネットワークを用いて最速避難を擬多項式時間で求めることができるが, ネットワークのサイズが大きくなると, その時間拡大ネットワークのサイズも巨大となる. 後に見るように, たとえば, 大阪市の津波浸水が予想されている地域では, 頂点数が約1.3万, 辺数が4万以上となる. また,  $\Theta$  は6,000 (秒) 以上と設定する必要がある. このとき,  $\mathcal{N}(\Theta)$  の

サイズは頂点数, 約8,000万, 辺数, 約2億4千万となる. この場合, 通常のPCで一般に知られている効率のよい最大フローのソフトウェアを用いても, 100日以上計算時間を要する. したがって, 大規模データに対しては, より効率のよいアルゴリズムの開発が必要である.

他方, すでに述べたように最速フロー問題における一般のネットワークに対する多項式時間アルゴリズムが, Hoppe and Tardos [9] によって提案されている. しかし, 高次元の多項式時間を必要とする劣モジュラー関数最小化をサブルーチンとして用いる必要があり, 実用的に効率がよいとは言えない.

以上の観点から, 筆者らは, 時間拡大ネットワークを用いない, 最速避難計画問題に対する実用的なヒューリスティックスを開発し, 大阪市の津波浸水が予想される地区への適用を行った [4]. 以下ではそのヒューリスティックスの概略を述べる.

## 2. 最速フロー問題に対する実用的なヒューリスティックス

時間拡大ネットワークを用いないヒューリスティックスの基本的な考え方は, 以下のとおりである. 動的ネットワークのソース  $s \in S^+$  からシンク  $t \in S^-$  への有向路  $P$  に着目しよう. サイズ  $b(s)$  のサプライを  $t$  に向かって流すことを考える. 1.2節のところで説明した考え方を一般化すると,  $c_{\min} = \min\{c(a) \mid a \in P\}$ ,  $\tau(P) = \sum\{\tau(a) \mid a \in P\}$  としたとき,  $\text{sup}(P) = \min\{b(s), l(t)\}$  だけ  $P$  に沿って流せる.  $\text{sup}(P)$  のサプライを流すのに必要な時間は

$$\text{length}(P) = \left\lceil \frac{\text{sup}(P)}{c_{\min}} \right\rceil + \tau(P) - 1$$

である. したがって, 最速避難に近い解を求めるには,  $\text{sup}(P)$  が大きく,  $\text{length}(P)$  が小さな  $s \in S^+$  からシンク  $t \in S^-$  への有向路  $P$  を多く見つける必要がある.

そのためにまず, スーパーソース  $ss$  とスーパーシンク  $st$  をネットワークに追加する.  $ss$  からすべてのソースに, サプライの値を容量とし, 移動時間が0の辺を追加する. すべてのシンクから  $st$  にシンク容量の値を容量とし, 移動時間が0の辺を追加する.  $ss$  から  $st$  への最小費用最大フローを見つける. ここで費用は移動時間を表すとす. そのフローを  $ss$  から  $st$  への辺素な有向路に分解し [10], 各有向路に沿って, サプライをシンクに流す. 各々の有向路に沿ったフローのことを鎖流 (chain flow) と呼ぶ. そうすると, いくつか

のソースのサプライが0になり、いくつかのシンクに容量の上限まで、サプライが溜まる。その後再び、新たに  $ss$  から  $st$  へ、最小費用最大フローの計算を行い、同様の計算を繰り返す。この計算をすべてのサプライがシンクに到達するまで行う。

まとめると、動的ネットワーク  $\mathcal{N} = (D = (V, A), c, \tau, b, S^+, S^-)$  が与えられているとき、まず実行可能な鎖流を獲得し、その後できるだけ隙間なくフローを流せるように、各鎖流の開始時刻・停止時刻を実行可能性を保ったまま調整する。この2段階の方法によって、ネットワークを時間拡大せずに、つまり大規模なネットワークを対象とした計算をすることなく高速に、最速避難に近い避難を得ることができる。

上記のアイデアをもう少し、詳細に説明しよう。鎖流の考え方はもともと、Ford and Fulkerson [7] によって導入された。彼らの扱う問題では、サプライの量やシンク容量は無限大であるので、得られた鎖流に対して時刻0から期間終了まで流せるだけ繰り返し流し続けることができる。そのようにして、動的フローを獲得することに用いていた。われわれの問題では、サプライやシンク容量制約を考慮するため、フローの開始時刻、終了時刻をそれぞれの鎖流について考える必要がある。

図3を用いて、上記のアイデアを具体的に説明しよう。図において、 $s_0, s_1$  がソース、 $t_0, t_1$  がシンクである。 $s_0, s_1$  に付した数字はサプライを表している。 $t_0, t_1$  に付した数字はシンク容量を表している。辺に付された数字のペアは、左の数字は容量、右の数字は移動時間を表す。

提案アルゴリズムは、まず、 $ss$  から  $st$  への最小費用最大フローを求める。最大フローの値は2であることから、最大フローの中で、費用最小のフローを求めると、そのフローは鎖流1と鎖流2の和からなる。鎖流1, 2とも単位時間当たりのフローの値は1である。時刻0から1単位時間、鎖流1と鎖流2の両方に1単位時間フローを流す。すると、鎖流2の終点のシンク  $t_1$  のシンク容量は1なので、これ以上、鎖流2を流すことができず、鎖流2はこの時点で終了する。その後鎖流1に沿って3単位時間フローを流した後、 $s_0$  のサプライが0になり、鎖流1も終了する。鎖流1の最後のサプライが  $t_0$  に到着する時刻は10である。

次に、鎖流1, 2の実行に伴って、 $s_0$  の残りのサプライは0、 $t_0$  の残りのシンク容量は1になり、 $t_1$  の残りのシンク容量は0になる。この変更を反映したネットワークに対して  $ss$  から  $st$  への最小費用最大フロー

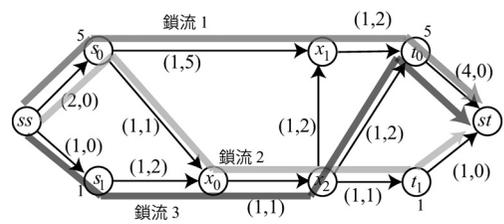


図3 ネットワークの例

を求めると、鎖流3が得られる。鎖流3の単位時間当たりのフローの値は1で、開始時刻は10である。鎖流3は1単位のサプライを  $t_0$  に送り終えた時点で終了する。その終了時刻は15である。

しかし、鎖流1, 2と鎖流3の間の開始時刻は大きく離れている。これは、確実に実行可能になるまで、鎖流3を開始していないからである。そこで、鎖流集合をその実行可能性を保ったまま隙間なく流すよう開始時刻・終了時刻を調整することを次に行う（詳細は [4] を参照）。図3の例題では、鎖流3の開始時刻を0に設定できる。

### 2.1 避難完了時間の遅延抑制パラメータの導入

提案するヒューリスティックスの基本的アイデアは、最小費用最大フローを計算し、それを鎖流に分解して、鎖流に沿って可能な最大の時間、そのフローを流すというものであった。最小費用最大フローは「静的ネットワーク上の最大フローの中で、費用の（この場合移動時間の）和が最小となるような経路」を選択する。つまり費用の最小化よりも流量の最大化が優先される。したがって、鎖流に分解したとき、鎖流の中には移動時間が長大なものが含まれている可能性がある。それが原因で避難完了時刻が大きな値になりうる。これを避けるために、最小費用最大フローを鎖流に分解する際に、分解して得られた鎖流集合の中で長さ（移動時間）最小のものに対して  $\alpha$  倍より長い鎖流を考慮の対象としないようにする。 $\alpha$  を遅延抑制パラメータと呼ぶ。この  $\alpha \geq 1$  をパラメータとして避難完了時間などの変化を見る実験も行った (3.2 節)。

## 3. 大阪市における最速避難計画の事例への適用

次に示す計算環境の下で計算実験を行った。道路ネットワークのデータは拡張版全国デジタル道路地図データベースから、避難所容量は大阪市が公表している避難所情報を使用し、南海トラフ地震による大阪市湾岸地域の津波浸水範囲は大阪府防災会議によるもの [11] を使用した。図4に対象の大阪市湾岸地域を示す。浸



図4 大阪市湾岸地域（津波の浸水範囲地図は [11] より）

水範囲の境界線から海側のエリアが想定される浸水範囲であり、その中に示された道路ネットワークが対象のネットワークである。避難人数は平成17年度国勢調査の夜間人口に基づいている。また建物情報に関してはZMapTownII 2008/09年度（Shape版）を使用した。全国デジタル道路地図データベース、ZMapTownは東京大学空間情報科学研究センターの研究用空間データを伴う共同研究により提供いただいた。また最大フローの計算はLEDA6.4を用いた。計算機の実環境は以下に示すとおりである：Intel(R) Xeon(R) CPU E5-2687Wv2@3.40 GHz, RAM: 256 GB, gcc 4.4.7.

計算実験では対象地域に対して時間拡大ネットワークによる手法と提案近似手法で計算を行い、1) 避難完了時間、2) 各時刻での避難完了者数の推移、3) 計算時間の比較から本手法を評価する。

### 3.1 大阪市湾岸地域

対象とする地域内の避難人数は、189,498人、避難所数197カ所であり、浸水範囲から外部へ通じる頂点数は389点である。道路ネットワークの頂点数は13,085、辺数は40,514である。

上述のパラメータ $\alpha$ を用いない場合、 $\alpha = 1, \alpha = 1.1$ の三つの場合に対して、提案近似手法を適用した結果を図5と表1に示す。さらに、各ソースの避難時間を図6に示す（黒ずんでいる部分が避難完了時間が長いところである）。表1において、時間拡大ネットワークを用いる手法の計算は時間がかかるため、単位時間を5秒としている。したがって単位時間を1秒とした

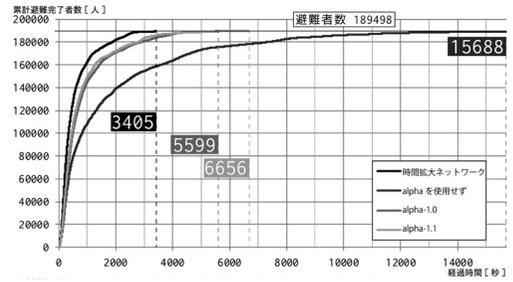


図5 避難完了者数の推移（大阪市湾岸地域）



図6 ソースごとの避難完了時間（大阪市湾岸地域）

表1 大阪市湾岸地域を対象とした計算結果（避難完了時間、計算時間）

計算手法	避難完了時間 [秒]	計算時間 [秒]
時間拡大ネットワークによる手法	3,405	30,735.2
近似解法（パラメータ $\alpha$ 使用せず）	15,688	313.5
近似解法（ $\alpha = 1.0$ ）	5,599	18,752.3
近似解法（ $\alpha = 1.1$ ）	6,656	3,275.3

場合よりも大幅に計算時間が短縮されている（計算精度も落ちている）ことに注意されたい。

2.1節で説明した理由により、パラメータ $\alpha$ を用いない場合、最速避難完了時間3,405秒を超えるような長さの鎖流が多量に存在する（鎖流の長さはその経路で避難した場合の移動時間と等しい）。パラメータ $\alpha$ を導入するとその点はかなり改善されている。実際、 $\alpha = 1$ のときに避難完了時間は最小の5,599秒となった。最速避難完了時刻3,405秒において97%程度の人が避難完了しており、良好な結果と言える。避難時間が遅い地域が集中しているのは港区、此花区、西淀川区であり（図6の丸で囲っている部分）、特に西淀川

表2 ランダムグラフに対する計算結果

頂点数	避難完了時間が最小となる $\alpha$	提案手法の避難完了時間 / 最速避難完了時間	提案手法の計算時間 / 時間拡大ネットワークによる手法の計算時間
1,000	1.1	1.04	0.0007
2,000	1	1.007	0.001
3,000	1.2	1.0099	0.0018
4,000	1.1	1.0023	0.0023
5,000	1.1	1.0012	0.0044
6,000	1	1.0090	0.0272
7,000	1.1	1.0262	0.105*
8,000	1.1	1.120	0.129*
9,000	1.1	1.051	0.152*
10,000	1.4	1.024	0.079*

区については現地調査の結果、津波避難ビルが少なく、その情報提示も不十分であることもわかった。

### 3.2 パラメータ $\alpha$ の考察

連結ランダムグラフを頂点数 1,000 から 10,000 まで 1,000 ごとに一つずつ、全部で 11 生成して同様の比較を行った (表 2 参照)。ただし頂点数 7,000 以上のものについては、時間拡大ネットワークによる手法は時間がかかるため、単位時間を 5 秒としている。したがって単位時間を 1 秒とした場合よりも計算時間は大幅に短縮されている (計算精度も落ちている) ことに注意されたい。

$\alpha$  の値が大きくなると計算時間が大きくなる傾向がある。避難完了時間についても同様である。これらの影響はネットワークごとに異なっているので、対象ごとに最適な  $\alpha$  を探索することが望ましい。

## 4. おわりに

本稿では、最速避難計画問題に対して、ネットワークフローに基づくモデリングの紹介を行った。また、実際的に有用な時間拡大ネットワークに基づく解法を紹介した。しかし、大規模な問題に対しては、計算時間が非常に大きくなるという欠点があることから、高速解法が実際にも必要である。本稿では、筆者らが開発

した、高速なヒューリスティクスを紹介し、大阪市における最速避難計画の適用事例を示した。

最速避難計画問題は、理論的にも、実際的にも、より一層の高速解法が望まれており、今後の研究の進展が期待される。

### 参考文献

- [1] 地震調査研究推進本部地震調査委員会, 「南海トラフの地震活動の長期評価 (第二版) について」, p. 13, 表 2, [http://www.jishin.go.jp/main/chousa/13may\\_nankai/nankai2\\_shubun.pdf](http://www.jishin.go.jp/main/chousa/13may_nankai/nankai2_shubun.pdf)
- [2] N. Kamiyama, “A study on the quickest flow problem in dynamic networks and the arborescence problem in directed graphs: A theoretical approach to evacuation planning in urban areas,” 京都大学博士学位論文, 2009.
- [3] N. Kamiyama, A. Takizawa, N. Katoh and Y. Kawabata, “Evaluation of capacities of refuges in urban areas by using dynamic network flows,” The 8th International Symposium on Operations Research and Its Applications (ISORA 2009), Lecture Notes in Operations Research 10, September, pp. 453–460, 2009.
- [4] 大田章雄, 神山直之, 瀧澤重志, 加藤直樹, “最速輸送問題に対する高速近似解法の提案及び避難計画への応用に関する研究,” 電子情報通信学会技術研究報告, COMP, コンピューテーション, **113**(488), pp. 29–34, 2014.
- [5] A. Takizawa, M. Inoue and N. Katoh, “An emergency evacuation planning model using the universally quickest flow,” *The Review of Socionetwork Strategies*, **6**, pp. 15–28, 2012.
- [6] L. R. Ford and D. R. Fulkerson, “Constructing maximal dynamic flows from static flows,” *Operations Research*, **6**, pp. 419–433, 1958.
- [7] L. R. Ford and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, 1962.
- [8] B. Hoppe and É. Tardos, “Polynomial time algorithms for some evacuation problems,” In *Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA’94)*, pp. 433–441, 1994.
- [9] B. Hoppe and É. Tardos, “The quickest transshipment problem,” *Mathematics of Operations Research*, **25**, pp. 36–62, 2000.
- [10] T. Hartman, A. Hassidim, H. Kaplan, D. Raz and M. Segalov, “How to split a flow?,” In *Proceedings of IEEE INFOCOM*, pp. 828–836, 2012.
- [11] 大阪府防災会議南海トラフ巨大地震災害対策等検討部会, 「大阪府域の被害想定について (人的被害・建物被害)」, <http://www.pref.osaka.jp/attach/20357/00000000/01siryu4.ppt%20.pdf> (2015 年 6 月 8 日閲覧)