

避難完了時間を最小化する梅田地下街の垂直避難場所の領域分割手法

PARTITIONING METHOD OF VERTICAL EVACUATION AREAS IN UMEDA UNDERGROUND MALL MINIMIZING EVACUATION COMPLETION TIME

環境図形科学研究室 山本 遼

Spatial & Graphic Science Lab. Ryo YAMAMOTO

地下街において、河川氾濫や津波による浸水発生時には接続する建物を利用した垂直避難が推奨されるが、大規模な地下街の場合、避難者数が建物の収容可能人数を超過し、避難が遅れる可能性がある。そこで接続ビルの収容可能人数を考慮しつつ、避難完了時間が最小となるような避難領域分割を求める手法の提案を本研究の目的とする。提案手法により、梅田地下街を対象として避難領域分割と避難完了時間の算出を行い、さらに提案手法の妥当性について検討する。

When we escape from flood in the underground mall, we should evacuate to the building connected the underground mall. However, in the case of a large-scale underground mall, the number of evacuees will be larger than the capacity. Further evacuation time may be delayed. Therefore, the purpose of this paper is partitioning method of vertical evacuation areas minimizing evacuation completion time. We calculate the evacuation area division and evacuation time of the Umeda Underground Mall using proposed method. We also discuss the validity of the proposed method.

1. はじめに

近年、台風や集中豪雨などによる河川氾濫や津波による浸水に対する防災対策が各地で行われている。特に大規模かつ複雑な空間を形成している梅田地下街では、利用者も多いことから地下街全体で統一した避難計画の策定が急がれている¹⁾。現在進められている計画では、浸水発生時に建物を利用して安全な階数まで垂直避難することを想定し、その際に避難所となる地下街接続ビルの収容可能人数を算出している。しかし、その収容可能人数を考慮して、地下街に存在する膨大な避難者の割当てまでは考えられていないのが現状である。

また既往研究として瀧澤ら²⁾は、梅田地下街を対象に Multi-Agent System (MAS) を用いて、避難者が最寄りの接続ビルへ向かい垂直避難をすると想定して、避難シミュレーションを行なった。その結果、避難完了時間は約 24 分となったが、避難者数が収容可能人数を超過する接続ビルがいくつか発生することがわかっている。この超過した避難者を適切に誘導できなければ避難完了時間が遅延する可能性も考えられる。

以上の背景より本研究では、各接続ビルの収容可能人数を制約としつつ、避難完了時間を最小化する避難領域の分割を求める問題 (Partition problem Minimizing Evacuation Completion Time with Capacity constraint:

PMECTC) を定式化し、汎用の数理計画ソルバーを用いて最適な領域分割を求める。定式化するにあたって、Dynamic Tree Network (DTN) と呼ばれる無向木上で定義される避難完了時間の推定式³⁾を用いる。この際 MAS を用いて、計算に必要なグラフのパラメータをキャリブレーションする。また、本稿では避難完了時間を考慮せず、避難者の総避難距離の最小化を目的関数とした領域分割を求める問題 (Partition problem Minimizing Total Evacuation Distance with Capacity constraint: PMTEDC) も同様に定式化と最適化を行う。それぞれの結果を比較することで、PMECTC による避難完了時間の短縮の程度を確認する。さらに得られた領域分割で避難シミュレーションを行い、MAS との避難完了時間の比較も行うことで、DTN 上における避難完了時間推定式、それを利用した定式化の妥当性も検証する。

2. Dynamic tree network の概要

いま、点(ノード)集合を V 、辺(エッジ)集合を E 、無向木のネットワークを $T = (V, E)$ と表すとす。無向木とは図 1 のような始点と終点と同じになるような経路が存在せず、エッジ上の進む方向が決まっていないグラフのことである。さらに各エッジの距離を l 、各ノード上に存在する避難者数を p 、辺容量を c 、避難者の歩行

速度を τ としたとき, DTN は $N = (T, l, p, c, \tau)$ と定義される. ここで辺容量は, 単位時間当たりには辺を通行できる避難者数の上限であり, 現実における通路幅を意味している.

次に, 避難場所となる根を $x \in V$ とし, 図1のような根を x とする無向木 $T(x)$ の避難完了時間 $\theta(x)$ を考える.

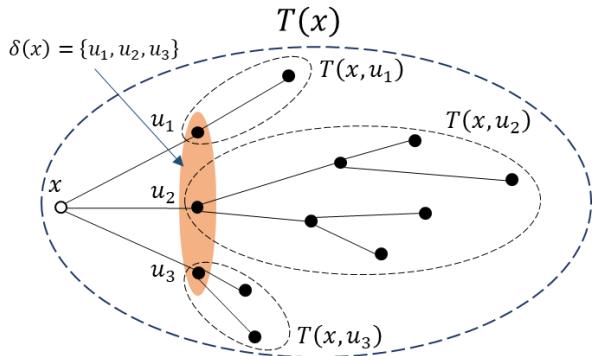


図1 無向木 $T(x)$ の例

x に隣接する節点集合を $\delta(x)$ とすると, $\theta(x)$ は次式で表わされる.

$$\theta(x) = \max\{\theta(x, u) | u \in \delta(x)\} \quad (1)$$

$\theta(x)$ を最大にする節点を $\hat{u} \in \delta(x)$ とすると, $\theta(x) = \theta(x, \hat{u})$ となることが分かる.

さらに, 辺容量 c が一定であるとき, 木をパスに変換しても変換前後で避難完了時間は変化しないことが Kamiyama らによって知られている⁴⁾. 木からパスへの変換は, 節点から根までの距離が小さいものから順にパス上へ並べることで可能である. 変換後のパスにおいて, 各ノードから根 x に向かって避難者が移動したとする. このとき 1秒あたりに根 x に到達する避難者数は最大 c 人である. ノード上の避難者が残っている状態で, 後ろのノードからの避難者が到達してしまうと渋滞となる. すなわち到達者数が c 人となる時間が連続する1つのグループ(クラスタ)が発生する(図2).

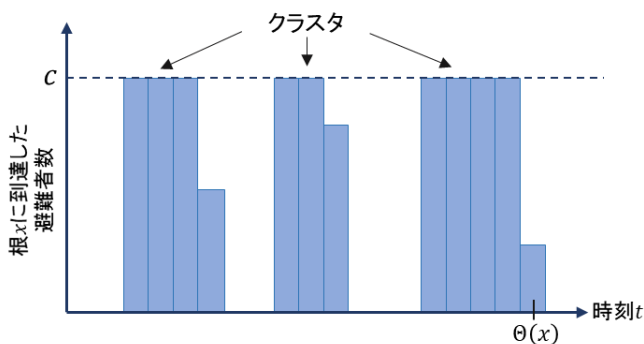


図2 渋滞のクラスタの例

ここで任意のクラスタの避難完了時間について考える. クラスタの先頭の避難者がノード $v \in V$ に存在し, ノード v から根 x までの距離を $d(x, v)$, $d(x, v_f \in V) \geq d(x, v)$ となるノード v_f の集合を $V_f(t, v)$ とする. このときノード v を先頭としたクラスタの避難完了時間 $\theta_{cluster-v}$ は次式で表すことができる.

$$\theta_{cluster-v} = \frac{d(x, v)}{\tau} + \left\lceil \frac{\sum_{w \in V_f(x, v)} p(w)}{c} \right\rceil - 1 \quad (2)$$

上式では, 歩行者密度が高いほどクラスタの時間幅が広がるので, 避難完了時間がその分延びることが示されている.

$\theta(x, \hat{u})$ を求めるには, 最後に到達するクラスタのみ考えればよい. すなわち式(2)が最大となるクラスタを求めればよいので, 避難完了時間は

$$\theta(x) = \max_{v \in V} \left\{ \frac{d(x, v)}{\tau} + \left\lceil \frac{\sum_{w \in V_f(x, v)} p(w)}{c} \right\rceil - 1 \right\} \quad (3)$$

となる. これが DTN における避難完了時間推定式である.

3. 収容人数制約付き最大避難完了時間最小化問題

以上の準備より, PMECTC の定式化について説明する. まず, 地下街の通路を連結なグラフとして表現し, 地下街全体の避難完了時間が最小となるように, グラフのノードを避難場所となる接続ビルに割り当てて考える. グラフのノードとその集合を $v \in V$, 避難場所とその集合を $t \in T$ とおく. さらにノード v 上の避難者数を $p(v)$, $p(v) > 0$ であるノードの集合を V_N , ノード v から避難場所 t までの最短距離を $d(t, v)$, $d(t, v_f \in V) \geq d(t, v)$ となるノード v_f の集合を $V_f(t, v)$, 避難場所 t の収容可能人数を $r(t)$, 避難者の歩行速度を一律に τ , t へ至る経路の平均的な辺容量を $c(t)$ とする.

以上と式(3)をもとに, ノード v が避難場所 t に配分されるかどうかを $x(t, v)$ の 0-1 変数で表すとすると, 避難領域分割問題は式(4)の混合整数計画問題として定式化できる. 本式の4つ目の制約条件は, ノード v が避難場所 t に割り当てられるとき, v に隣接し, v よりも t までの最短距離が小さいノードの集合を $G(t, v)$ とした際に, ノード $g \in G(t, v)$ のうち少なくとも1本は t に割り当てられることを意味している. この制約により各避難場所の領域内のグラフが必ず連結となる. さらに, 6つ目の制約条件は式(3)の条件に相当するが, 同制約条件の右辺第2項は, 最適化の都合上, 整数条件を外している.

$$\begin{aligned}
& \text{Minimize } \theta_{Max} \\
& \text{s.t. } x(t, v) \in \{0,1\} \quad \forall v \in V, \forall t \in T \\
& \sum_{t \in T} x(t, v) = 1 \quad \forall v \in V \\
& r(t) \geq \sum_{v \in V} x(t, v) \cdot p(v) \quad \forall t \in T \\
& x(t, v) \leq \sum_{g \in G(t, v)} x(t, g) \quad \forall v \in V, \forall t \in T \quad (4) \\
& \theta_{Max} - \theta(t, v) \geq 0 \quad \forall v \in V_N, \forall t \in T \\
& \theta(t, v) = \left(\frac{d(t, v)}{\tau} - 1 \right) \cdot x(t, v) \\
& \quad + \frac{\sum_{u \in V_f(t, v)} p(u) \cdot x(t, u)}{c(t)}
\end{aligned}$$

次に PMTEDC を PMECTC と同様に混合整数計画法を用いて式(5)のように定式化する。目的関数は避難者の目的地までの最短距離の合計値の最小化である。制約は避難完了時間の考慮がないこと以外は PMECTC と同じである。

$$\begin{aligned}
& \text{Minimize } \sum_{t \in T} \sum_{e \in E} x(e, t) \cdot d(e, t) \cdot p(e) \\
& \text{s.t. } x(t, v) \in \{0,1\} \quad \forall v \in V, \forall t \in T \\
& \sum_{t \in T} x(t, v) = 1 \quad \forall v \in V \quad (5) \\
& r(t) \geq \sum_{v \in V} x(t, v) \cdot p(v) \quad \forall t \in T \\
& x(t, v) \leq \sum_{g \in G(t, v)} x(t, g) \quad \forall v \in V, \forall t \in T
\end{aligned}$$

4. 最適化

本研究では、高速な数理計画ソルバーの1つである Gurobi optimizer 7.5.1 を使用し、PMECTC と PMTEDC の解を求める。これらはいずれも混合整数計画問題であり、分枝限定法により解が求められる。以降、Gurobi optimizer を単にソルバーと呼ぶ。まずソルバーで最適解を求めるために、解かせたい問題を記述した LP ファイルを作成する。グラフデータ、エッジ長さ、避難者数、収容人数などの地下街のデータから、問題の LP ファイルを作成するプログラムを Visual C++ 2015 により実装した。作成された LP ファイルをソルバーに読み込ませ、その計算結果を出力し、GIS を用いて可視化する。計算環境は、OS: Windows 10 Professional 64 bit, CPU: Intel Core i7-4790K, Memory: 16GB である。

以下に、問題設定と結果について述べる。

4.1. 設定

対象範囲は東西南北に約 1.1km 四方の範囲に広がる梅田地下街の通路、接続ビル、さらに5つの地下鉄駅構内を含んだ図 3 に示す範囲である。MAS ソフト

(SimTread) による計算では平面で計算する必要があるため、図 3 のように階の異なる駅構内は地下街から切り離して表記している。この範囲をグラフで表現するのだが、グラフの精度による最適化に必要な計算時間の変化も比較するために、ノード間の距離間隔がおおよそ 10, 20, 30m となる 3 パターンのグラフを作成した(図 5 参照)。それぞれのノードとエッジの数を図 4 に示す。グラフ化する際、階の異なる駅構内と改札、接続ビルとその階段を距離 0 のエッジで接続している。また出口が複数ある接続ビルについて、収容可能人数が 1,000 人未満の場合は同様に出口を距離 0 のエッジで仮想ノードに接続し、この仮想ノードを目的地として接続ビルの収容可能人数を与える。対して 1,000 人以上の場合は各出口を目的地とし、それぞれに割り当てられた避難者の合計が接続ビルの収容可能人数以下となるように設定する。地下街、駅構内に分布する避難者は、過去に行われた歩行者断面交通量調査等の結果²⁾から推定し、地下街全体に避難者を 14,782 人配置した。ただし今回は電車の乗客を除いている。



図 3 対象範囲

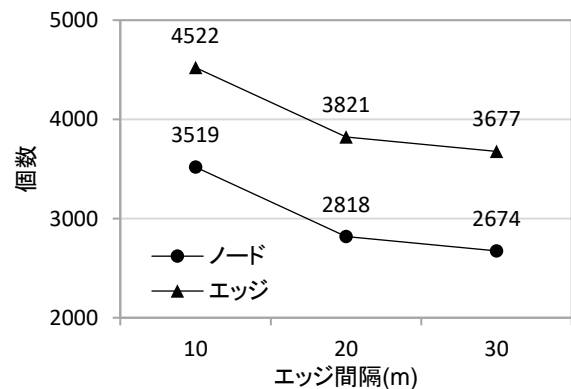


図 4 各ノード間隔グラフのエッジ・ノード個数



図5 ノード間隔 30mのネットワーク

式(4)では、避難者の歩行速度 τ を設定する必要がある。本研究では平地での歩行速度を 1.0m/s、階段でのそれを 0.45m/s と仮定するが、式(4)は歩行速度が一定という制限があるため、階段部分の長さを 1.0/0.45 倍に延長して、常に $\tau = 1.0\text{m/s}$ で歩行するものとした。さらに、避難場所 t へ至る経路の平均的な辺容量 $c(t)$ (人/s)も設定する必要がある。そのために事前に MAS を用いて、避難場所毎に 15 分以上の渋滞が発生するようシミュレーションを行い、各避難者の避難場所への避難時間を求め、次式で辺容量を決定した。表 1 左列に求められた各接続ビルの辺容量を示す。

$$c(t) = \frac{\text{避難完了 600 秒前までの } t \text{ への避難者数}}{600 \text{ 秒}} \quad (6)$$

4.2. 結果

ノード間隔の異なるネットワーク 3 パターンそれぞれについて、PMECTC と PMTEDC による定式化と最適化をおこなった。PMTEDC は最適化結果のみでは避難完了時間がわからないので、式(3)の推定式を用いて最大避難完了時間を算出した。

それぞれの計算時間と最大避難完了時間を表 1 にまとめた。最適化の計算に要した時間は、PMTEDC ではすべてのネットワークで 30 秒以下と比較的短かった。一方 PMECTC の同ネットワークでは、5 日間計算機を動かし続けたが収束しなかったため、計算を中断した。ノードの間隔が大きいグラフほどギャップは小さくなり、30m 間隔では 11.6% まで計算が進んでいた。そこで 30m 間隔について、それぞれの最大避難完了時間をみると PMTEDC では 38 分以上かかっていたが、

表 1 各最適化結果の計算時間・最大避難完了時間

定式化	エッジ 間隔 (m)	計算時間	双対 ギャップ (%)	式(3)より推定した 避難完了時間 (m:s)
PMECTC	10	5 日	19.7	27:24
	20	5 日	17.4	25:51
	30	5 日	11.6	25:44
PMTEDC	10	21 秒	0	38:15
	20	11 秒	0	38:22
	30	14 秒	0	38:21

※PMECTC は全てのネットワークにおいて計算時間 5 日間で計算を中断

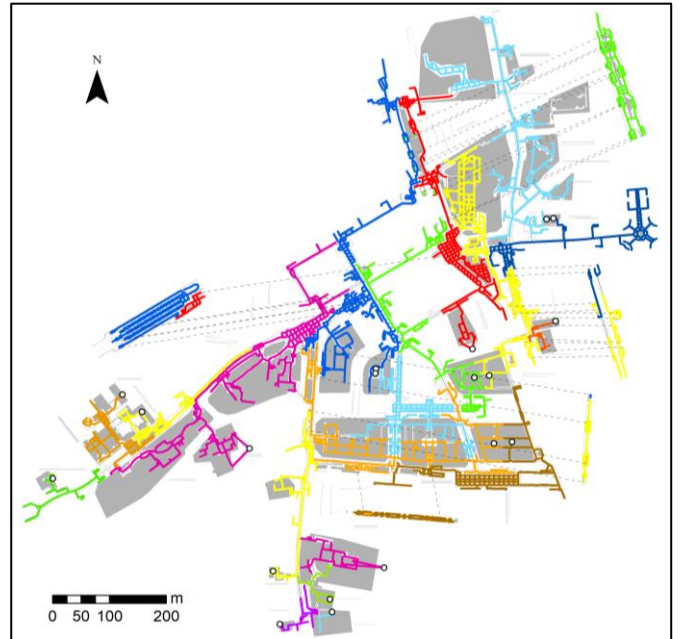


図6 PMECTC による梅田地下街の垂直避難領域分割
(ノード間隔 30m ネットワーク)

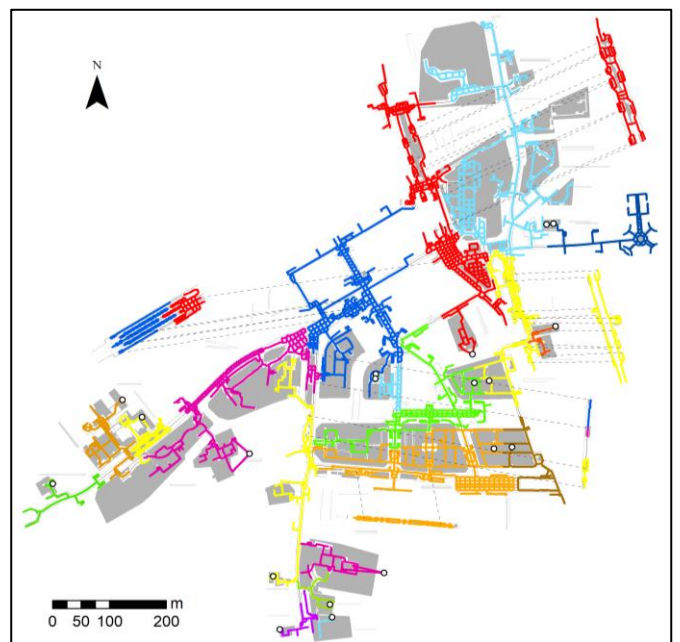


図7 PMTEDC による梅田地下街の垂直避難領域分割
(ノード間隔 30m ネットワーク)

表 2 各接続ビルの属性・避難者数・収容率
(ノード間隔 30mネットワーク)

接続ビル	辺容量 (人/s)	収容可能人数 (人)	避難者数(人) 収容率(%)			
			PMECTC		PMTEDC	
A	1.53	109	29人	26.6%	29人	26.6%
B	1.41	59	50	84.7	50	84.7
C	1.33	122	111	91.0	72	59.0
D	1.18	707	47	6.6	47	6.6
E	1.26	797	797	100.0	797	100.0
F	1.22	406	299	73.6	128	31.5
G	1.38	549	0	0.0	0	0.0
H	1.38	145	69	47.6	69	47.6
I	1.83	3,859	2,473	64.1	3,859	100.0
J-1	1.07	6,708	1473	44.9	688	32.9
J-2	1.08	-	1,540	-	1,520	-
K-1	1.22	2,115	947	100.0	1,375	76.2
K-2	1.23	-	1,168	-	237	-
L-1	1.32	2,477	1,815	94.7	2,292	100.0
L-2	1.07	-	530	-	185	-
M	1.06	130	37	28.5	37	28.5
N	1.33	860	860	100.0	860	100.0
O-1	1.32	2,537	1,576	100.0	2,185	100.0
O-2	1.32	-	961	-	352	-

表 3 各接続ビルの避難完了時間
(ノード間隔 30mネットワーク)

接続ビル	避難完了時間 (m:s)			
	PMECTC		PMTEDC	
	式(3)の 推定値	MAS	式(3)の 推定値	MAS
A	02:58	02:44	02:58	02:45
B	04:35	04:17	04:35	04:17
C	05:01	04:52	04:51	04:21
D	03:30	03:17	03:30	03:15
E	13:05	13:03	13:05	13:10
F	07:59	07:36	05:14	05:10
G	00:00	00:00	00:00	00:00
H	03:14	02:60	03:14	03:02
I	25:44	25:45	38:21	38:16
J-1	25:44	25:36	13:30	13:19
J-2	25:43	25:55	25:25	25:50
K-1	16:04	15:55	21:52	21:45
K-2	17:55	17:52	05:18	05:13
L-1	25:43	25:37	31:44	31:32
L-2	10:58	10:47	05:35	05:23
M	03:52	03:26	03:52	03:27
N	15:28	15:03	14:17	14:02
O-1	22:19	22:15	29:58	29:37
O-2	14:17	14:08	06:36	06:20

PMECTCでは26分以下になり、12分以上も短縮されたことがわかった。ここで表2にノード間隔30mネットワークにおける各接続ビルの属性、避難者数、収容率をまとめた。PMTEDCではビルIに4,000人近く避難者が集中していたが、PMECTCではビルIの避難者数が1,000人以上減少している。またビルOにおいて、2つの出口の避難者数の偏りがPMTEDCでは解消していることが分かる。表3は間隔30mにおける各接続ビルの避難完了時間をまとめたものである。式(3)の推定式を用いて、領域分割結果から各避難所の避難完了時間を算出したものがPMECTC、PMTEDCのそれぞれ左列になる。PMTEDCではビルIの避難完了時間が突出して大きい、PMECTCは避難者数の多いビルの避難完了時間がおおよそ25分40秒に収束していることがわかる。

さらに図6,7に30m間隔ネットワークにおけるPMECTCとPMTEDCによる垂直避難領域分割結果を示す。前者のそれは後者のそれと大きく異なっている。例えばPMECTCでは、ディアモールの円柱の広場から大阪駅へ向かう広い通路の左右で、異なる接続ビルに向かっているなど、通常では思いつきにくい複雑な領域分割が得られた。

5. MASによる避難完了時間の検証

得られた領域分割結果を元に、MASの歩行者シミュレーターSimTread v2⁵⁾を用いて、避難完了時間を算出する。SimTreadのデフォルト状態では、避難者は最寄りの避難場所へと移動するが、目指すべき避難場所を避難者毎に設定することも可能である。本研究では図6,7で示した領域分割の結果をもとに、避難者の避難先を設定した。領域分割の際に使用したデータを基に避難者の配置を行い、歩行速度も同じに設定し、避難者は避難場所となる接続ビルの階段を上り、避難階に到達した時点で避難完了とする。

表3のPMECTC、PMTEDCのそれぞれ右列にMASによる各避難ビルの避難完了時間を示す。式(3)の推定式との差は大きいところでも30秒以下である。このことから、避難完了時間の推定値とMASのシミュレーションによる避難完了時間の差は小さく、構築したネットワークモデルの計算の妥当性が示されたといえる。

6. 通行不可能な通路発生時の領域分割

上記の結果から、PMECTCの定式化による避難完了時間の妥当性が示された。そこで4章の設定とは異なる条件についてPMECTCによる最適化を行い、領域分割や避難完了時間に影響するのかを検証する。

今回は浸水や火災によって通行不可能となった通路

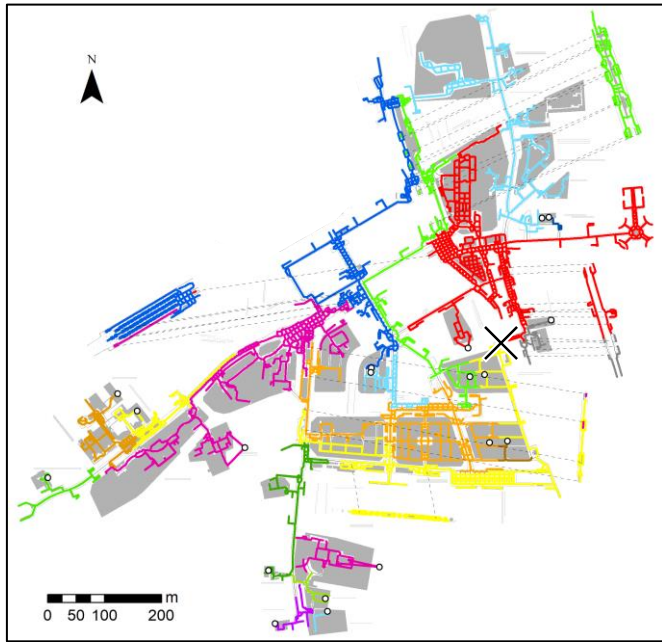


図8 通行不可パターン1
最大避難完了時間：32分21秒

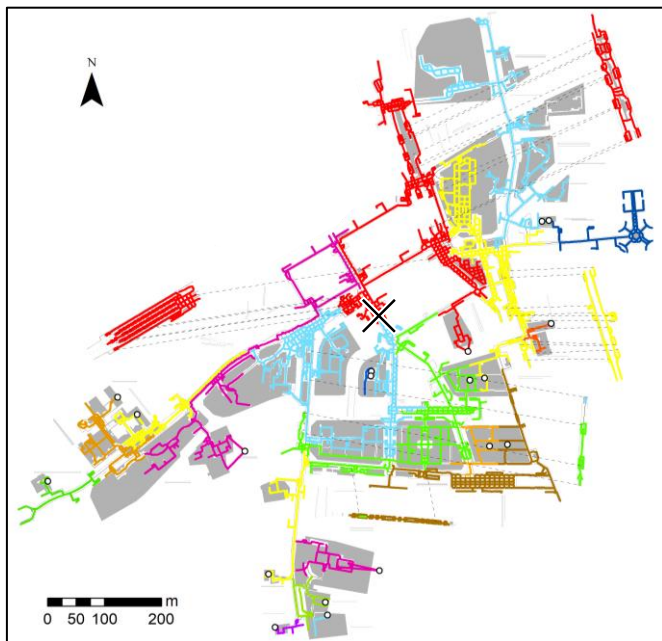


図9 通行不可パターン2
最大避難完了時間：38分3秒

が発生した場合を想定する。避難者数や歩行速度など基本的な条件は4章と同じで、ノード間隔30mのネットワークを使用する。通行不可とする位置のエッジを削除することで、通行不可を表現することができる。

本誌では図8, 9に示す2パターンを紹介する。それぞれ図中のバツ印の位置が通行不可にした場合のPMECTCによる領域分割結果である。それぞれ5日以内に最適化計算が完了し、最大避難完了時間はどちらも30分を超過する結果となった。図6のPMECTCの領域分割結果と比較すると、通行不可とした位置付近

の領域が大きく変化していることがわかる。このように通行不可による避難時間の変化をみることで、どの箇所が避難において重要なかが明確になり、想定外の事故や災害発生時の避難についても考察することができる。

7. まとめ

本研究では、梅田地下街を対象とした接続ビルの収容人数を制約とした避難領域分割問題について、DTN上の避難完了時間推定式を用いたPMECTCを提案し、避難完了時間を考慮せず総避難距離を最小化するPMTEDCの結果と比較した。MASによる検証の結果、PMECTCの避難完了時間は約25分となり、PMTEDCのそれと比べて12分以上の避難完了時間の短縮が可能であることを示した。またPMECTCによる各接続ビルの避難完了時間とMASの結果は近い値となり、定式化の妥当性を示すことができた。さらに通行ができないというような状況も条件を変更することで計算可能であり、避難計画への応用も期待できる。しかし、最適化に要する時間が長くかかっており、今後はこの短縮を図る必要がある。一つの方向性として、現在のネットワークモデルが避難誘導を行うには分割が複雑であるため、領域の境界が単純な形状となるような定式化の改善を行いたい。

参考文献

- 1) 大阪市地下空間浸水対策協議会：大規模な地下空間の浸水対策の取り組み(2017年10月2日確認)
<http://www.city.osaka.lg.jp/kikikanrishitsu/page/0000259323.html>
- 2) 瀧澤重志, 高木尚哉, 谷口与史也：浸水被害を想定した梅田地下街の垂直避難シミュレーション, 大阪市立大学都市防災研究プロジェクト 都市防災研究論文集, 2, pp.35-38, 2015.
- 3) Y. Higashikawa, M. J. Golin and N. Katoh: Minimax Regret Sink Location Problem in Dynamic Tree Networks with Uniform Capacity, Journal of Graph Algorithms and Applications, vol.18, no. 4, pp. 539–555, 2014.
- 4) N. Kamiyama, N. Katoh and A. Takizawa: An Efficient Algorithm for Evacuation Problem in Dynamic Network Flows with Uniform Arc Capacity, IEICE Transaction on Fundamentals, Vol.E89-D, No.8, 2372–2379, 2006.
- 5) 木村謙ほか：マルチエージェントモデルによる群集歩行性状の表現：歩行者シミュレーションシステムSimTreadの構築, 日本建築学会計画系論文集, 74(636), pp.371-377, 2009.