Vol.2019-MPS-125 No.10
2019/9/25

ベイズ最適化を用いたフロー型実時間避難計画

姫野 湧太[1]　徳永 潤平[1]　榎原 博之[2]　上田 修功[3]

概要：避難計画に関する研究が土木や OR をはじめとした様々な分野で注目されている．地震などの大規模な災害は建造物の崩壊や道路の陥落により避難経路の状況が変化するおそれがあるため，災害後に避難計画の策定を行い，状況の変化に適応させた計画を提供することが非常に重要である．そこで，本研究では，避難計画の分野で既に知られている時間拡大ネットワークを用いた手法にベイズ最適化を適用することにより質の高い避難計画を高速に導出する手法を提案する．本手法を実在する地域を対象に適用し，従来手法との比較を行い，本手法が計算コストを抑えた上で効率の良い避難計画を導き得ることを示す．

1. はじめに

近年，避難計画の策定に関する研究が様々な分野において注目されている．避難計画に基づいた経路に適切に要避難者を誘導することにより，迅速かつ安全な移動を実現することができる．また，シミュレーションを行うことにより，避難においてボトルネックとなっている要素を解析することが可能であり，道路や避難所のどの部分を改善すべきという提案ができる．

一般的に避難計画は災害が発生する前に策定することが多い．しかし，大規模な災害は想定外の事態が多く発生するため，様々なシナリオにおいて避難計画を策定する必要がある．例えば，地震や台風などの規模の大きい災害は建物の崩壊や道路の陥落など，平素は通行可能な道路が落下物で塞がれ通れなくなるなど，避難経路に影響を及ぼすおそれがある．そのような場合，事前に策定した避難計画では実際の状況との矛盾が発生し得るため，避難者が却って混乱してしまう．そこで，災害によって変化した状況を適用させた地図データに基づいて再度避難計画の策定を行い，より効果的な避難計画を提供することが非常に重要となる．つまり，災害発生後にそのシナリオにおいて最適な避難計画に基づいた指示を避難者を待たせることなく素早く提供することが可能な策定手法が必要である．

本研究では，ベイズ最適化 [1] を用いて最大フローの計算回数を低減し，単位時間を動的に変化させて冗長な辺や頂点を省くことにより，ネットワークフローを用いた高速な避難計画の策定手法を提案する．その後，実際の地理情報に対して提案手法及び先行手法を適用し，避難完了時間及び計算時間，最大フローの計算回数の観点から評価を行う．

[1] 関西大学大学院理工学研究科
Graduate School of Science and Engineering, Kansai University
[2] 関西大学システム理工学部
Faculty of Engineering Science, Kansai University
[3] 理化学研究所　革新知能統合研究センター
RIKEN, Center for Advanced Intelligence Project

2. 避難計画の策定手法

避難計画の策定手法には様々なものがあり，モデル化手法の違いにより，1 つのエージェントを 1 人の避難者と見立てそれぞれに行動モデルを設定し，個々のエージェントに属性を持たせるマルチエージェントモデル，避難者をネットワークを流れるフローとして評価するネットワークフローモデルの 2 つに分けることができる．それぞれのモデル化手法の特徴を以下に示す．

- マルチエージェントモデル

  あらかじめ設定された意思決定メカニズムと行動計避難シミュレーションに基づいて自律的に行動するものをエージェントとよび，そしてそれらが相互関係をもつ集合体のことをマルチエージェントと呼ぶ．マルチエージェントが各々相互作用を及すことによってシステム全体の流れが生まれ，その全体の流れによってエージェントの個々の動きが動的に変化する様子を表現するモデルをマルチエージェントモデルと呼ぶ．設定する属性によって避難者の心理とそれに伴う行動特性などを避難シミュレーションに適用させることができるため，より現実に即した避難計画を導出可能だが，避難者 1 人を 1 つのエージェントで表すため計算コストが高い特徴がある．それによって，計算時間が膨大となるおそれがある．そのため，この手法は避難訓練の参考にされるなど，災害発生前に効率の良い避難計画を要する場合に向いているといえる．マルチ

エージェントモデルを用いた避難シミュレーションとして，構造計画研究所が開発したマルチエージェントシミュレータ (MAS) がある．

- ネットワークフローモデル

  実際の地理情報をグラフ理論におけるネットワーク，対象のエリアに存在する避難者群の避難行動をそのネットワークを流れるフローでそれぞれモデル化を行い，与えられた時間内にネットワークの出発点から到着点まで流すことが出来る最大のフローを求めることにより，ある時間までの安全なエリアに移動可能な人数を求めることができる．なお，この手法は避難者群をフローとして扱うため，避難者数が多くなっても計算時間が膨大になることがない．

本研究では災害発生後に策定アルゴリズムを実行し避難計画を迅速に避難者に提供することを目的としているため，実時間で解を導出する手法が必要となる．そして計算コストが避難者数に依存せず，比較的処理の負荷が少ない後者のネットワークフローを用いた避難計画の策定手法を扱う．

2.1 先行研究

マルチエージェントモデルを用いた避難シミュレーション手法の例として，新井らの研究 [3] が挙げられる．この研究の目的は，健常者に加え，車椅子や高齢者など，避難に要するスペースや避難時の歩行の速さが異なる集団での効率の良い避難方法の導出である．これは，エージェントにそれらの特徴を適用した属性を設定し，シミュレーションを行うことで実現されている．しかし，計算時間については一切考慮されておらず，災害発生後ではなく，災害発生前に用いる手法として用いられてる．また，マルチエージェントモデルにおいて，エージェントに与えるパラメータをネットワークフローを用いた手法により決定するハイブリッドな手法も Noh ら [4] によって提案されているが，これも同様に計算時間は考えられていない．

ネットワークフローを用いた最速避難計画を導出する手法は Ford and Fulkerson[2][5] よって，最初に提案されている．その方法は移動時間の要素を持つ動的ネットワークを移動時間の要素を持たない静的ネットワークに変換する手法である時間拡大ネットワークを用いて，始点から終点までの最大の流量を求めるもので，そのアルゴリズムを繰り返し用いて，最速避難完了時間を求める．さらに，Kamiyama ら [6] によって避難所の容量を考慮した場合の定式化が行われた．なお，時間拡大ネットワークは混雑が予想される状況で，渋滞を防ぐための必要な設備や面積を検討する研究 [7] にも用いられている．しかし，従来の時間拡大ネットワークを用いた手法は多大な計算時間を要するため，加藤らが時間拡大ネットワークを用いない手法 [8] を提案した．この手法は時間拡大を行っていないため，大幅に計算時間を抑えることができたが，解の悪化がみられた．

3. フローを用いた避難計画

本節では，時間拡大を用いた手法と，加藤らが提案した時間拡大を用いないヒューリスティックな手法を説明する．なお，本研究ではそれらを比較手法に用いる．

3.1 最大フロー問題

標準的な (静的) ネットワークフロー問題はグラフ $G = (V, A)$ で定義される．ここで V は頂点集合，A は辺集合を表している．また，各辺 a は容量関数 $c(a)$ をもっており，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 s_i) と呼ばれる頂点と終点 (シンク t_i) と呼ばれる頂点が指定されたとき，容量関数を満たした上で始点から終点まで，できるだけ多くの資源を輸送可能なフローを見つける問題である．また，そのフローを最大フローと呼ぶ．最大フロー問題は，エドモンド・カープのアルゴリズム [9] で多項式時間で解くことができる．以下より，エドモンド・カープ法について詳しく説明する．

3.1.1 エドモンド・カープ法

エドモンド･カープ法は Ford らによって提案されたフォード・ファルカーソン法を改良したものである．フォード・ファルカーソン法は，その時点で得られているフローに対し，あとどれだけ流せるかを表したネットワークである残余ネットワークを作成し，そこでパス (増加パス) を見付けて，それに添ってフローをできる限り増加させる．この手順を，増加パスが無くなるまで繰り返すことにより最大フローを導出する．フォード・ファルカーソン法は深さ優先探索によりパスを探索しているが，エドモンド・カープ法は幅優先探索を用いている．これにより，問題点であった収束性や計算量の観点で改良されている．

3.2 動的ネットワーク

$N = (V, A, c, \tau, S^+, S^-, sup)$ で動的ネットワークは定義され，各辺 a に整数の移動時間関数 $\tau(a)$ が与えられている．移動時間関数を持っていることにより渋滞による待ちを表現することができる．避難計画において，頂点集合 V は建物や交差点を表し，辺集合 A は道路を表している．また，容量関数 c は単位時間あたりにその道路を通ることが出来る避難者数の上限，移動時間関数 $\tau(a)$ は辺 a を通過するのに要する時間を表す．避難者の存在する出発点 (s_i) の集合を S^+，避難所や津波の予測浸水域外などの避難先となる到達点 (t_i) の集合を S^- とし，供給関数 sup はある頂点に存在する避難者数を表している．

3.3 時間拡大ネットワーク

Ford ら [5] によって提案された，動的ネットワークの各

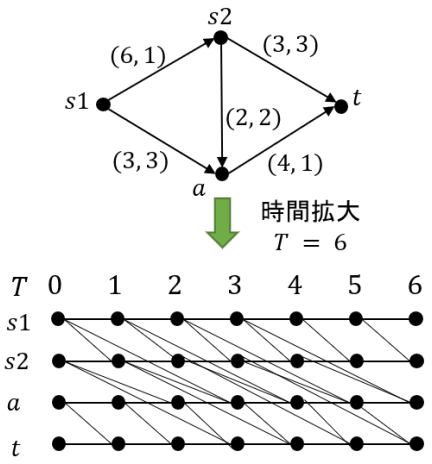

図 1 時間拡大の例

辺の各時刻の状況を移動時間の要素を持たない静的ネットワークに変換する手法である．元の動的ネットワークの各ノードを複製し，各ノードから隣接ノードへ移動時間を考慮した辺を追加することにより構築する．期間 T を 6，単位時間を 1 としたときの動的ネットワークと対応する時間拡大ネットワークの例を図 1 に示す．図 1 において，辺に付されている数字はそれぞれ道の幅を表す辺の容量 c と接続されている頂点から頂点への移動にかかる時間 τ を表している．

3.4 時間拡大ネットワークを用いる手法

以下より，原始的な時間拡大ネットワークを用いて避難計画を導出する手法を説明する．

3.4.1 仮想的な辺及び頂点の追加

前述した時間拡大ネットワークは各避難所に存在する避難者数 (供給関数 $sup(s_i)$) 及び各避難所の収容可能な人数 ($|sup(t)|$) を考慮していないが，避難問題をモデル化したネットワークフローを考える際にはそれらを考慮しなければならない．そこで各ソース，シンクの供給関数を考慮するためにいくつかの仮想的な頂点を加える．各点の供給関数を考慮した時間拡大ネットワークの構成方法は [6] で説明されている．構成手順を簡単に説明すると次のようになる．図 1 の下図に仮想的な頂点及び辺を追加したものを図 2 に示す．

(1) 各ソース s に対して代表点となる点 s_i^{rep} を 1 つずつ追加し，容量 ∞, 移動時間 0 の辺を追加する．この作業を各シンク t 側にも同様に行い，代表点 t_i^{rep} 及び辺を追加する．

(2) スーパーソース ss，スーパーシンク st と呼ばれる 2 点を追加し, 容量をソース側は存在する避難者数, シンク側は避難所の収容可能人数とし，移動時間を 0 とした辺を追加する．

3.4.2 避難完了時間の導出

前項のいくつかの頂点及び辺を追加した時間拡大ネットワークを用いて以下の操作を行うことにより最速避難時間を導出する．

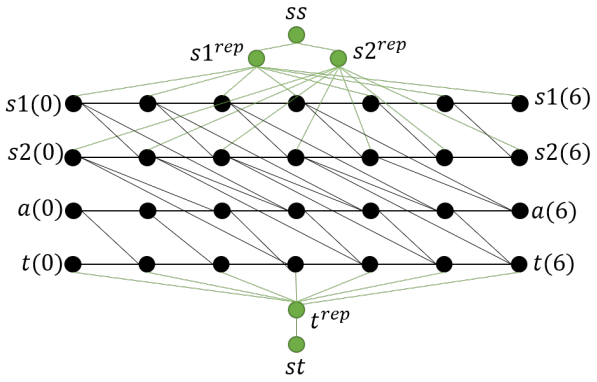

図 2 避難計画における時間拡大ネットワーク

(1) 期間 $T = 0$ とする．

(2) 得られた最大フローが総避難者数を上回るまで次の操作を繰り返す．

(a) 期間 T を単位時間分増やして時間拡大する．

(b) ss から st までの最大フローを計算する．

(3) 操作 (2) のループが終了したときの T が避難完了時間となる．

この手法は単位時間の点で厳密な避難経路を導出可能だが，対象のネットワークが非常に大きくなり，それに伴い計算コストが多大になるという問題がある．

3.5 時間拡大ネットワークを用いない手法

時間拡大ネットワークはネットワークサイズが非常に大きくなり，計算コストが増大する．このことから加藤らが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 [8] を提案した．この手法は計算時間は前項のものと比較して大幅に短縮されるが，ネットワークによっては避難完了時間が非常に大きくなるという問題点ある．この手法は実行可能な鎖流の獲得，鎖流の時間的圧縮という 2 つの作業により実現される．以下に，各々について説明する．

3.5.1 鎖流

鎖流の考え方もまた Ford and Fulkerson [2] によって提案された．そこでの鎖流はフローの経路及び流量の組を鎖流としていたが，避難計画においてはそれらに加えて，フローの開始時刻と終了時刻をそれぞれの鎖流について考える．

3.5.2 鎖流の獲得

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，避難者数の値を容量とし，移動時間が 0 の辺を追加する．すべてのシンクから st に避難所の収容人数の値を容量とし，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．最小費用最大フローとは最大フローの中で費用 (この場合は移動時間) が最小のフローである．そのフローを

ss から st への経路に分解する [10]．各々の経路に沿ったフローが鎖流 (chain flow) である．これにより，いくつかのソースの避難者数が 0 になり，いくつかのシンクに収容可能人数の上限まで避難者が溜まる．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避難者が避難所に到達するまで行う．

3.5.3 鎖流の時間的圧縮

前項のアルゴリズムのみだと求められた鎖流の開始時刻が異なるステップで求められた鎖流の開始時刻と大きく離れている可能性がある．そこで，実行可能性を保ったまま隙間なく流すように開始時刻及び終了時刻を調整する．これは「ある経路のフローが流れている最も遅い時刻より+1秒以上であれば同時に同じ辺を使用することはなく実行可能性は保たれる」という方針のもと実現される．

4. 提案手法

4.1 先行研究の問題点及び研究目的

前項で述べた時間拡大ネットワークを用いた避難計画の策定手法 (以下，従来手法) は元のネットワークのサイズや期間 T によっては，時間拡大後のネットワークが非常に大きなサイズになることがある．この手法では時間拡大後の大きなネットワークに対して最大フローを繰り返し求めるため，計算コストが膨大になる．そのため，災害発生後に避難計画を導出し，迅速に避難者にそれを提供することは難しい．

それに対して，加藤らが提案した手法は時間拡大を行っていない分，計算時間は従来手法と比較して大幅に抑えることができた．しかし，対象のネットワークによっては避難完了時間が非常に大きくなってしまう．鎖流の獲得において，最大フローのうち最小の費用 (時間) のフローを鎖流として選んでいるため，選択されたフローの中に移動時間が長大なものが含まれている可能性があり，それが原因で避難完了時間が大きな値になりうる．

そこで，災害発生後に素早く，安全な避難経路を避難者に提供することができ，避難開始後に住居の倒壊などにより道路状況が変わったときには，随時避難計画を策定し，リアルタイム性を持った避難計画を導出することのできる手法を提案する．従来手法を時間拡大ネットワークを用いた従来手法を以下の 2 つの方針で改良する．

- ベイズ最適化を用いることにより最大フローの計算回数を減らし，全体の計算コストを抑える
- 単位時間を動的に調整することにより冗長な辺及び頂点を省き，計算コストを抑える

以下より，各方針の詳細を説明する．

4.2 最大フローの計算回数の低減

本手法では解が得られるまで繰り返し最大フロー問題を解くため，避難計画の策定における全体の計算コストに大きく影響する．そのため，この計算回数を抑えることが計算時間の短縮には非常に重要である．通常，従来手法においての最大フローの計算回数は避難完了時間と同数となる (単位時間を 1 にした場合)．例えば，避難完了時間が 100 の場合は最大フローの計算を 100 回行うこととなる．この内の大半 (主にループの開始直後) の最大フローの計算はあまり意味がなく，省くことができる．しかし，避難完了時間は対象のネットワークや避難者の人数によって大きく異なるため，最適な最時間拡大における期間 T の初期値を見つけることは容易ではない．

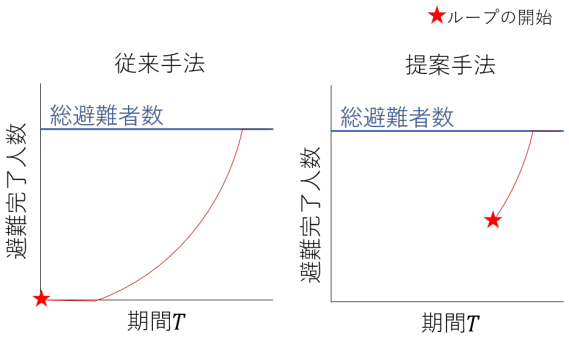

図 3 最大フローの計算のイメージ

そこで，総避難者数に対して避難が完了した人数の割合 (以下，避難率) と計算時間を評価値としたベイズ最適化を避難計画の策定手法に適用することにより，様々なシナリオにおいて適切な T の初期値を設定する．また，それに伴い，効率の良い避難計画を計算時間を抑えて導出可能な手法を実現する．図 3 に従来手法と提案手法の最大フロー問題の計算を行うループの開始点のイメージを示す．

4.2.1 ベイズ最適化

最適化すべき変数の目的関数が陽に記述できない場合の最適化 (Black box 最適化) 手法であり，近年，機械学習におけるハイパーパラメータ探索や，実験計画に多用される手法である [11]．

ベイズ最適化は，逐次的な探索アルゴリズムにより，ガウス過程回帰に基づき，最適化すべき変数の候補値を逐次的に探索する．ランダムな探索に比べ非常に効率的な探索が可能であり，本研究の様に，一度の評価にコストがかかる応用に適した手法と言える．

4.2.2 ベイズ最適化の避難計画への導入

ベイズ最適化に基づく提案法では，以下の評価値を最大化する避難計画を逐次探索することになる．なお，α は各パラメータの重みを表している．

$$\text{評価値} = \alpha * \text{計算時間}^{-1} + \text{避難率} \tag{1}$$

まず，期間 T を初期化する．次いで，その期間 T の値で，3.4.2 で説明した避難計画を策定し，その計画で式 (1) を評価し，その評価値に基づいてベイズ最適化により，よ

り評価値が大きくなる期間 T を逐次的に求め，その期間 T での避難計画を策定する．以上の操作を収束判定を満たすまで実行することで，式 (1) をできるだけ大きくする避難計画を効率良く求めることができる．

4.3 単位時間の調整

単位時間は時間拡大後のネットワークサイズに直接関与する．1 秒毎に時間拡大を行った場合は 1 秒以内の厳密解を導出できるが，避難に用いられない冗長な頂点や辺が多く存在してしまう．逆に，単位時間を大きく設定してしまうと時間拡大後のネットワークサイズが小さくなり，計算時間は短縮される (図 4) が，避難完了時間は増大する．

そこで，避難の進捗状況に応じて単位時間を動的に変化させ，冗長な辺の発生を抑えることにより効率の良い解を高速に導出する．

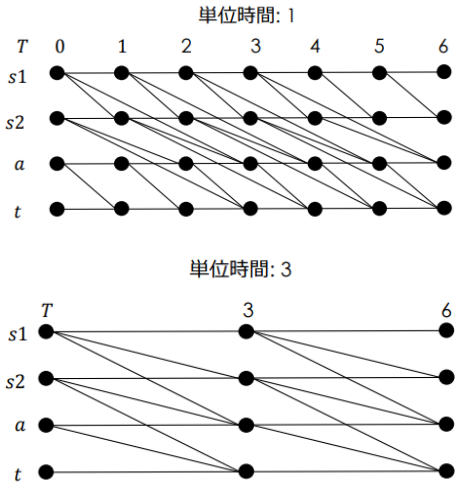

図 4 単位時間を変化させたときの例

本研究では，式 (2) のように単位時間の初期値を 1 とし，ベイズ最適化により決定された期間 T における避難率の逆数と単位時間の積を次ステップの単位時間に設定する．なお，T は現在のステップ，T' を次のステップで設定される期間を表している．そうすることにより，冗長な辺を省略することが可能で．高速に避難計画を導出することができる．なお．避難の推移と単位時間の関係のイメージを図 5 に示す．

$$T' = T * (1/\text{避難率}) \tag{2}$$

5. 計算機実験

3.4 で説明した従来手法，3.5 で述べた加藤らの手法，そして前項の提案手法を，実際の地理情報を元に構築した動的ネットワークに対して実行し，避難完了時間及び計算時間の比較を行う．以下で，計算機実験を行ったシナリオ，パラメータについて説明する．なお，地図データは

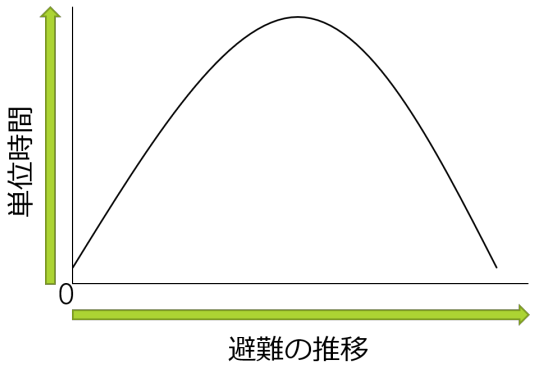

図 5 避難の推移と単位時間のイメージ

OpenStreetMap[12] より取得した． OpenStreetMap から取得できる辺は，始点及び終点となる頂点の情報と道の長さや道の幅を表している属性を持っている．

5.1 シナリオ

以下の特徴の違う 2 つのシナリオにおいて計算機実験を行い，各手法の比較を行う．避難者は 1.25[m/s] で避難を行い，道の長さをこの移動速度で割ったものを各辺の持つ移動時間関数とする．

- 関西大学千里山キャンパス及びその周辺 (シナリオ 1)
- 新国立競技場及びその周辺 (シナリオ 2)

例としてシナリオ 1 を図 6 に示す．

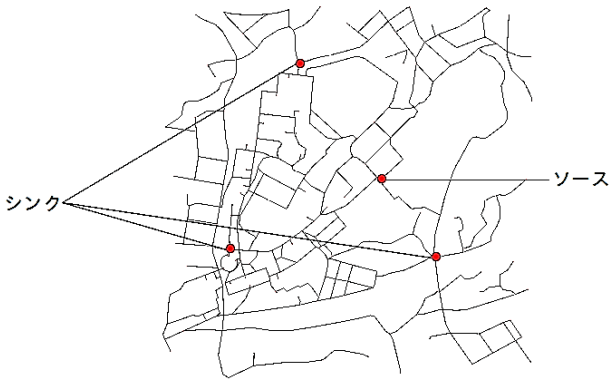

図 6 関西大学千里山キャンパス及びその周辺地域

5.2 パラメータ

実験パラメータを表 1 に示す．提案手法において，式 (1) の重み α は 0.5, 1.0, 2.0 の 3 つのパターンで実行する．

なお，シンク (避難所) の容量は避難者数に対して十分にあると仮定する．

表 1　実験パラメータ

項目	設定値
避難者数	2000[人]
移動速度	1.25[m/s]
道幅	1, 2, 5, 10[m]
単位時間 (従来手法)	1[秒]

5.3 実験結果

5.3.1 シナリオ 1

従来手法，加藤らの手法，提案手法を適用したときの避難完了時間，計算時間，最大フローの計算回数をまとめたものを表 2 に示す．提案手法が従来手法と同等の避難計画を，$\alpha = 0.5$ のとき 81%程度の計算時間で導出している．また，それに伴い最大フローの計算回数も大幅に低減されている，

表 2　実験結果 (シナリオ 1)

手法	避難完了時間 [秒]	計算時間 [秒]	計算回数 [回]
従来手法	38496	5454	38496
加藤らの手法	75130	2154	—
$\alpha = 0.5$	38496	4417	38477
$\alpha = 1.0$	38496	4754	38454
$\alpha = 2.0$	41020	3918	37545

5.3.2 シナリオ 2

従来手法，加藤らの手法，提案手法を適用したときの避難完了時間，計算時間，最大フローの計算回数をまとめたものを表 3 に示す．シナリオ 2 においては，$\alpha = 0.5$ のとき 68%程度の計算時間で導出している．最大フローの計算回数もシナリオ 1 と同様の傾向がみられる．

表 3　実験結果 (シナリオ 2)

手法	避難完了時間 [秒]	計算時間 [秒]	計算回数 [回]
従来手法	87542	9454	87542
加藤らの手法	105457	2755	—
$\alpha = 0.5$	87542	6421	47575
$\alpha = 1.0$	90455	5474	45421
$\alpha = 2.0$	95421	5002	44741

6. まとめ

本研究では，従来の時間拡大ネットワークを用いた手法をベイズ最適化を用いることによる最大フロー計算回数の低減，時間拡大における単位時間の可変化という 2 つの方針の元，改良を行い高速化を図った．そして，本手法，従来手法，加藤らの手法を実際の地域に適用し，性能比較を行い，提案手法が従来手法よりも大幅に計算時間を抑えつつも，同等の避難完了時間を導出することが出来た．また，評価値の重み α の調整による緊急度や優先度に応じた避難計画の策定が可能だった．なお，本手法は災害時以外でも混雑する駅やイベント会場での動線計画にも応用できる．

今後の課題として，現状では期間 T の初期値の探索にのみ用いたベイズ最適化を，単位時間も含めた探索にも拡張することでより様々な状況において効率の良い避難計画を導出可能な順応性の高い避難計画策定手法の考案などが挙げられる．

謝辞　本研究の一部は，JSPS 科研費 18K11484 と，JSPS 科研費 17K01309，関西大学大学院理工学研究科高度化推進研究費，関西大学先端科学技術推進機構「緊急救命避難支援のための情報通信技術に関する研究開発」研究グループの助成を受けている．

参考文献

[1] Pelikan, Martin, David E. Goldberg, and Erick Cantú-Paz. "BOA: The Bayesian optimization algorithm." Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1. Morgan Kaufmann Publishers Inc., (1999).
[2] Ford Jr, Lester Randolph, and Delbert Ray Fulkerson. Flows in networks. Princeton university press, (2015).
[3] 新井健, 増田浩通, and 落合哲郎. "災害弱者を考慮したマルチエージェント避難シミュレーションモデル." 第 3 回 KKMAS コンペティション論文集 (2003): 117-125.
[4] Noh, Dong-jin, Jeongin Koo, and Byung-In Kim. "An efficient partially dedicated strategy for evacuation of a heterogeneous population." Simulation Modelling Practice and Theory 62 (2016): 157-165.
[5] Ford Jr, Lester R., and Delbert Ray Fulkerson. "Constructing maximal dynamic flows from static flows." Operations research 6.3 (1958): 419-433.
[6] Kamiyama, Naoyuki, et al. "Evaluation of capacities of refuges in urban areas by using dynamic network flows." The 8th International Symposium on Operations Research and Its Applications (LNOR 10). (2009).
[7] 渡部大輔, 鳥海重喜, and 田口東. "東京オリンピック・メインスタジアムへの観戦客に対する新宿御苑を活用した動線計画." 都市計画論文集 52.3 (2017): 1341-1348.
[8] 加藤直樹, and 瀧澤重志. "最速避難計画のモデリングと解法." オペレーションズ・リサーチ (2015): 437-442.
[9] Edmonds, Jack, and Richard M. Karp. "Theoretical improvements in algorithmic efficiency for network flow problems." Journal of the ACM (JACM) 19.2 (1972): 248-264.
[10] Hartman, Tzvika et al. "How to split a flow?" 2012 Proceedings IEEE INFOCOM (2012): 828-836.
[11] Snoek, Jasper, Hugo Larochelle, and Ryan P. Adams. "Practical bayesian optimization of machine learning algorithms." Advances in neural information processing systems. (2012).
[12] OpenStreetMap, "`https://www.openstreetmap.org`"