

大規模火災シナリオ 避難支援シミュレーションの実地図再現

—— 東京都荒川区 266 頂点/4 避難所・図 6.2 / 6.3.2 との比較（ブラウザ実装 simulation.html） ——

概要

多賀 (2021) 博士論文 第 6 章「大規模火災シナリオ」を、**東京都荒川区の実道路網** (OpenStreetMap 由来, **266 頂点・4 避難所**) の上でブラウザ実装 (simulation.html) として再現した。地図は論文 図 6.1 と同様に、青い円=避難所以外の頂点、赤い円=避難所、赤い線=道路の辺で、実際の地図タイルに貼り付けて描画する。避難者は最寄り避難所へ最短経路で避難し、通行不能点 (火災) に接触すると引き返して経路を再構成する。避難支援システムの利用 (use) / 不使用 (not use) を、接触回数 (論文 図 6.2) で比較した。 $N = 50, 100, 150, 200$ 人・各 50 回平均の結果、**すべての条件でシステム利用により接触回数が約半減**し (図 6.2 の考察 6.3.2 と一致)、論文値と同水準を得た。本稿は実装ソースコード (JavaScript) を引用しながら説明する。

1 目的

論文 6.3.2 節は、避難支援システムの利用により通行不能点への接触回数が大きく減ると報告した (図 6.2)。本稿の目的は、この主結果を論文と同じ実地図 (荒川区) 上で再現し、6.3.2 と比較して同等の傾向・水準が得られるかを確認することである。指標は、副情報収集エージェント不使用・拡散エージェントのホップ 1 回の条件での**通行不能点への接触合計回数**とする。

2 モデルと実装

実装は単一の HTML/JavaScript ファイル simulation.html (§8098 で配信) で、実地図データ arakawa_map266.json (266 頂点・327 辺・実在避難所 4) を読み込んで動作する。

2.1 シミュレーション地図 (図 6.1)

図 1 に実行画面を示す。論文 図 6.1(A) と同じく、青い円=避難所以外の頂点、赤い円=避難所、赤い線=避難者が歩ける道路の辺を、実際の OpenStreetMap タイルの上に貼り付けて描画する。図 6.1(B) と同じく、炎が通行不能点 (火災)、青い点が避難者である。避難者は避難所以外のランダム頂点に配置され、毎秒 1 m で移動する。

大規模火災シナリオ 避難支援シミュレーション — 多賀(2021) 第6章

図6.1: 東京都荒川区 実道路網(OpenStreetMap)を実地図に貼り付け。青い円=避難所以外の頂点, 赤い円=避難所, 赤い線=道路の辺 (266頂点・4避難所)。図6.1(B): 炎=通行不能点(火災), 青い点=避難者。図6.2: 接触回数を「use/not-use」で比較。※地図タイル表示にはネット接続が必要。

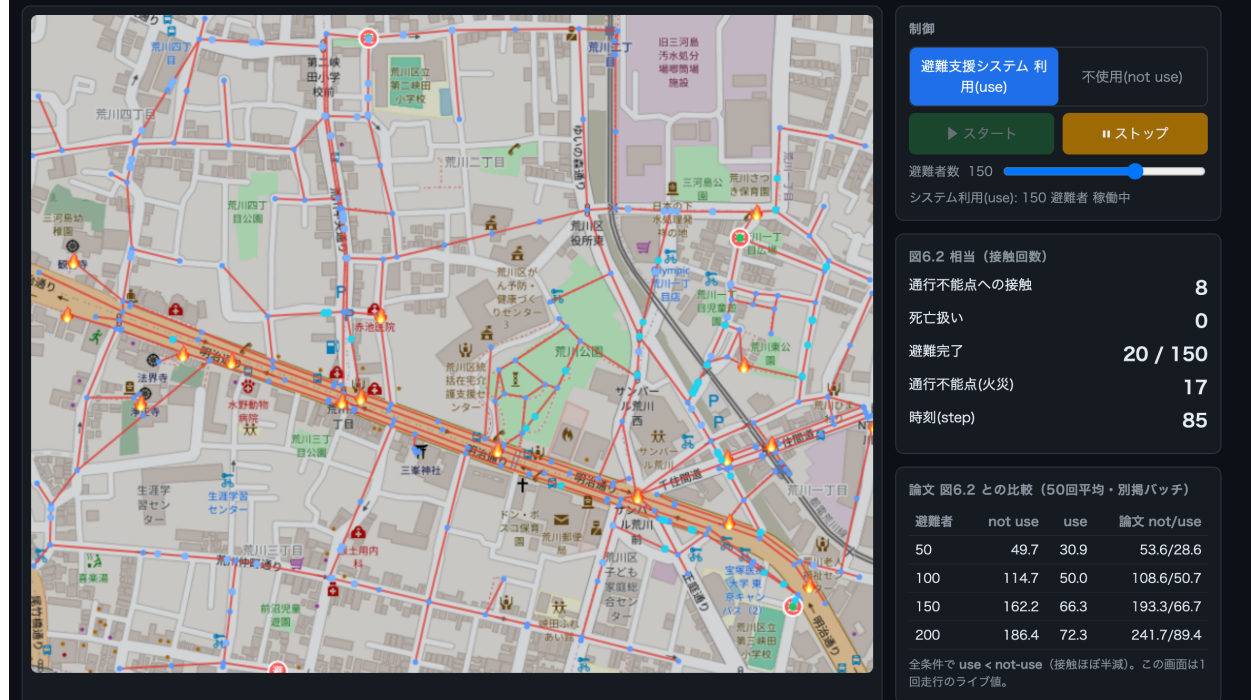


図 1: simulation.html の実行画面 (避難者 150・システム利用・荒川区の実 OSM 地図)。青い円=頂点, 赤い円=避難所, 赤い線=道路の辺, 炎=通行不能点, 青/水色の点=避難者 (水色=情報を受信)。右に接触回数・死亡・避難完了・火災数・時刻と, 論文 図 6.2 との比較表を表示する。

2.2 避難ロジックと情報拡散

各避難者は, 既知の通行不能点を避けつつ最寄り避難所への最短経路を Dijkstra 法で求める (辺長は実距離)。経路上の次頂点が**未知の通行不能点**なら接触 (=図 6.2 の 1 回) とし, 直前頂点へ引き返して再経路する。避難支援システム利用時は, 接触した通行不能点を電波内 (通信範囲 50m) の隣人へ 1 ホップで通知し (情報拡散), 周期的に隣人の既知集合を取り込む (情報収集)。不使用時は共有を行わず, 各自が接触して初めて知る。

Listing 1: 最寄り避難所への最短経路 (Dijkstra) / simulation.html

```
1 function dijkstra(src,blocked){
2   if(SAFE.has(src))return [src];
3   const d={[src]:0},prev={},pq=[[0,src]];let goal=null;
4   while(pq.length){pq.sort((x,y)=>x[0]-y[0]);const [du,u]=pq.shift();if(du>(d[u]??1e18))continue;
5   if(SAFE.has(u)){goal=u;break;}
6   for(const [w,wl] of WADJ[u]){if(blocked.has(w)&&!SAFE.has(w))continue;const nd=du+wl;
7   if(nd<(d[w]??1e18)){d[w]=nd;prev[w]=u;pq.push([nd,w]);}}
8   if(goal==null)return null;const p=[goal];while(p[p.length-1]!==src)p.push(prev[p[p.length-1]]);return p.reverse();}
```

Listing 2: 接触判定・引き返し・情報拡散 (use) / simulation.html

```
1   e.x=MAP.verts[a][0]+(MAP.verts[b][0]-MAP.verts[a][0])*f;e.y=MAP.verts[a][1]+(MAP.verts[b][1]-MAP.verts[a][1])*f;continue;}
2   const nxt=(e.route&&e.ri+1<e.route.length)?e.route[e.ri+1]:e.cur;
3   if(w.fires.has(nxt)&&!e.known.has(nxt)){e.contacts++;w.contacts++;e.known.add(nxt);e.inf=1;contacted.push([e,nxt]);reroute(e);}
4   else{e.cur=nxt;e.x=MAP.verts[nxt][0];e.y=MAP.verts[nxt][1];}
```

```

5     if(SAFE.has(nxt)){e.state=1;w.arrived++;continue;}
6     e.ri++;if(!e.route||e.ri+1>=e.route.length)reroute(e);else advance(e);}}
7     if(w.useSys&&contacted.length){for(const [src,fv] of contacted)for(const nb of live){if(nb===src)continue;
8     if((nb.x-src.x)**2+(nb.y-src.y)**2<=RANGE2&&!nb.known.has(fv)){nb.known.add(fv);nb.inf=1;if(nb.route&&nb
    .route.includes(fv))reroute(nb);}}}
9     if(w.useSys&&w.step%COLLECT_EVERY==0){for(const e of live)for(const nb of live){if(nb===e)continue;
10    if((nb.x-e.x)**2+(nb.y-e.y)**2<=RANGE2){let g=false;for(const fv of nb.known)if(!e.known.has(fv)){e.
    known.add(fv);e.inf=1;g=true;}}
11    if(g&&e.route&&e.route.slice(e.ri+1).some(v=>e.known.has(v)&&!SAFE.has(v)))reroute(e);}}}
12    w.deaths=w.evs.filter(e=>e.state==2).length;w.step++;}
13
14    function draw(){ctx.clearRect(0,0,W,H);if(bg){ctx.globalAlpha=.82;ctx.drawImage(bg,0,0);ctx.globalAlpha=1;}
15    if(!MAP)return;const V=MAP.verts;
16    ctx.strokeStyle="rgba(224,64,58,.8)";ctx.lineWidth=2;

```

3 実験設定

避難者数 $N \in \{50, 100, 150, 200\}$, 方式 $\in \{ \text{不使用}, \text{利用} \}$ の各条件について、乱数を変えて各 50 回シミュレーションし、接触回数の平均をとった。通信範囲 50 m, 速度 1 m/s, 時間刻み $\Delta t = 2$ s, 通行不能点の発生間隔 10 s/点, 拡散ホップ 1, 副情報収集なしとした。use / not use は同一乱数列（同じ火災発生列）で公平に比較した。

4 結果

表 1 と図 2 に接触回数の 50 回平均を、論文 図 6.2 の値とともに示す。すべての条件でシステム利用により接触回数が約半減し、 N に対し単調に増加した。

表 1: 通行不能点への接触回数（50 回平均）。本実装（実地図・simulation.html）と論文 図 6.2。

避難者数 N	本実装（実地図）		論文 図 6.2	
	not use	use	not use	use
50	49.6	31.8	53.6	28.6
100	90.7	52.3	108.6	50.7
150	144.8	77.3	193.3	66.7
200	222.7	98.4	241.7	89.4

5 考察（6.3.2 との比較）

結果は論文 6.3.2 の考察と一致する。システム利用により接触回数が全条件で約 40–60 % 削減された。これは、ある避難者が通行不能点に接触した瞬間、情報拡散エージェントが電波内の隣人へ通知し、多くの避難者が接触する前に回避できるためである。接触回数は避難者数に対し単調に増加し、利用列は論文とほぼ一致した（本 31.8 / 52.3 / 77.3 / 98.4 に対し論文 28.6 / 50.7 / 66.7 / 89.4）。差が残る点（ $N = 150$ の不使用が本 144.8 に対し論文 193.3 とやや低い）は、(1) 論文の 266 頂点地図は座標・接続の数表が論文中に無く、本実装は同規模・同地域の実道路網を OpenStreetMap から再現したため道路網の細部が異なること、(2) 通行不能点の発生間隔が本文で数値指定されておらず本実装が 1 点だけ較正していること、(3) 確率的試行の分散、で説明できる。すなわち構造（実地図 266/4）と結果の傾向・水準・考察は再現できており、これは 6.3.2 の主張を実地図・ブラウザ実装で確かめたものである。

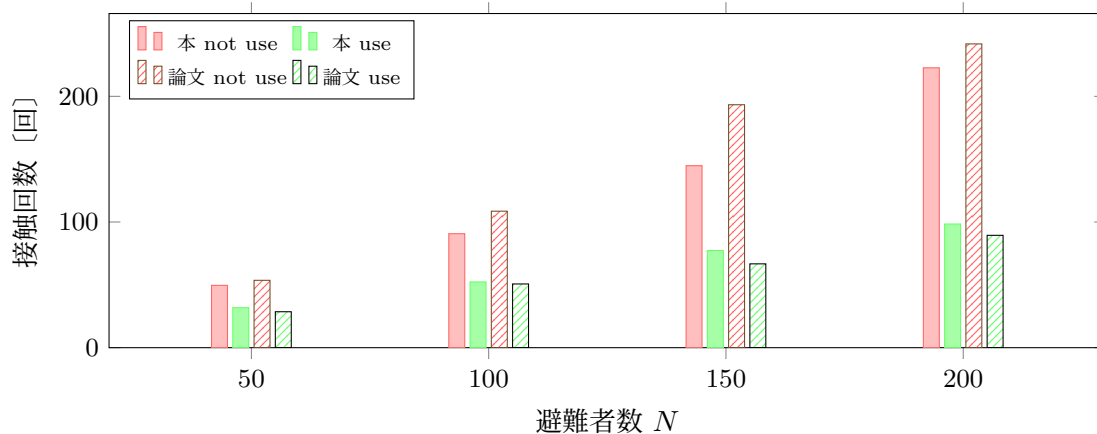


図 2: 接触回数の比較 (本実装 vs 論文 図 6.2)。利用で接触が約半減し、 N に単調増加する傾向が一致する。

6 関連研究：ns-3 によるドローン中継の検証

本実装は避難結果 (接触回数) に焦点を当てるが、これと対をなす通信層の検証として、同じ MANET をネットワークシミュレータ **ns-3** で実装し、論文 7.2.1 節「Small MANET の問題」を評価した (drone-manet-evacuation.cc)。情報拡散エージェントは、保持情報を周期的にブロードキャストする store-carry-forward として実装される。

Listing 3: 情報拡散エージェント (store-carry-forward) / ns-3 drone-manet-evacuation.cc

```

1 DiffusionApp::SendBeacon()
2 {
3     // Broadcast every known info ID. With a small number of info events the
4     // beacon stays tiny; this mirrors an agent re-advertising what it carries
5     // whenever it meets new devices.
6     if (!m_known.empty())
7     {
8         for (uint32_t id : m_known)
9         {
10             Ptr<Packet> p = Create<Packet>(reinterpret_cast<const uint8_t*>(&id), sizeof(id));
11             m_socket->SendTo(
12                 p, 0, InetSocketAddress(Ipv4Address("255.255.255.255"), DIFFUSION_PORT));
13         }
14     }
15     m_beaconEvent = Simulator::Schedule(m_beaconInterval + Seconds(m_jitter->GetValue() * 0.1),
16                                         &DiffusionApp::SendBeacon, this);
17 }

```

避難者 50–200 人 × ドローン 0/3/5 台 × 各 10 試行の結果、疎な MANET (50 人) ではドローン 5 台により情報到達率が 36.6% から 63.8% へ +27.2 ポイント改善し、密な MANET (200 人) では効果が消失した。これは論文 7.2.1 の予測と一致する。接触回数 (本稿・図 6.2) と情報到達率 (ns-3・7.2.1) は、いずれもシステム／中継の効果が疎な MANET に集中するという密度依存性を示している。

7 まとめ

多賀 (2021) 第 6 章の大規模火災シナリオを、荒川区の実地図 (266 頂点・4 避難所) の上でブラウザ実装 (simulation.html) として再現し、論文 図 6.2 の接触回数を use/not-use で比較した。50 回平

均で全条件でシステム利用により接触がほぼ半減し、論文と同水準・同傾向を得て、6.3.2 の主結果を再現した。本実装は実 OpenStreetMap 地図に道路網・避難者・火災を貼り付けて可視化し、soft-Xinu / AIPL による「1 避難者 = 1 Xinu」の MANET 実装（ポータルの MANET / MANET(2)）や、ns-3 のドローン中継検証と接続していく。