

災害時緊急水上輸送システムの技術開発

その3：河川を用いた被災者輸送のシミュレーション解析

物流研究センター ＊間島 隆博、勝原 光治郎

1. はじめに

災害時に総合的かつリアルタイムな水上輸送計画の作成を可能とする「災害時緊急水上輸送システム、シミュレーションプログラム」を開発した。このシステムを用い図1に示す河川での、**救援物資**に関する輸送シミュレーションの解析、結果については既に報告済みである^{(1),(2)}。ここでは新たに、**被災者輸送**を目的としたシミュレーターを構築し、物資輸送と同様、図1に示す関東地方（隅田川、荒川、小名木川）における被災者輸送能力の把握を試みた。なお、被災者移動需要のデータとして、本講演集の「その2：被災者輸送需要の推計結果」に示したOD表を使用する。



図1 解析対象河川とリバーステーションの位置

2. 解析手法

シミュレーター本体で採用している手法はマルチエージェントシステムである。この手法は、「局所的な情報に基づき行動するエージェントがシステムの中で相互作用し、新たな状況を作り出し、その状況に影響されてエージェントが行動するというシステム全体と個（エージェント）の関係を

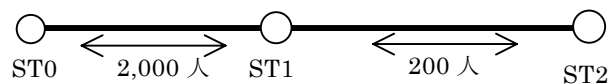
表現するシミュレーション方法」である。今回、被災者輸送の問題に対応して新たに2つのエージェントを追加する。一つは、被災者輸送船エージェントであり、もう一つは被災者エージェントである。

被災者輸送船は**物資輸送船**の運行形態^{(1),(2)}と大きく異なる。物資輸送船は、適切な供給地、需要地の組み合わせが輸送のたびに異なる。一方、被災者輸送船に関しては、あらかじめ決められた航路を往復あるいは循環する、いわゆる定期船の運航形態が妥当と思われる。

被災者輸送船は路線に沿って往復（各駅停車）を繰り返すが、移動需要を満たすべく輸送ネットワーク内を複数の路線であらかじめ構成する必要がある。一方、被災者エージェントは路線の情報を考慮して乗り換え等の旅程を自律的に決定しなければならず、この過程をプログラム化する必要がある。以降の節ではネットワーク内の路線決定の過程と被災者の旅程決定アルゴリズムの概要について記す。

2. 1 路線の決定

次の例では、ステーション ST0 と ST1、ST1 と ST2 が隣り合い、そのステーション間の移動量はST0,1間がST1,2間と桁違いの差があることを意味している。



路線を ST0,1,2 の往復航路で構成すると、船舶数はST0,1間の需要を満たすべく決定することになる。ST1,2間にとってこの船舶数は過剰であり、不効率な輸送システムとなってしまう。よって、路線の決定は、隣接ステーション間の移動量をク

ラス(移動量のオーダーが等しいものをクラスとした)分けし、各クラスをまとめて路線を構成する方針をとる。

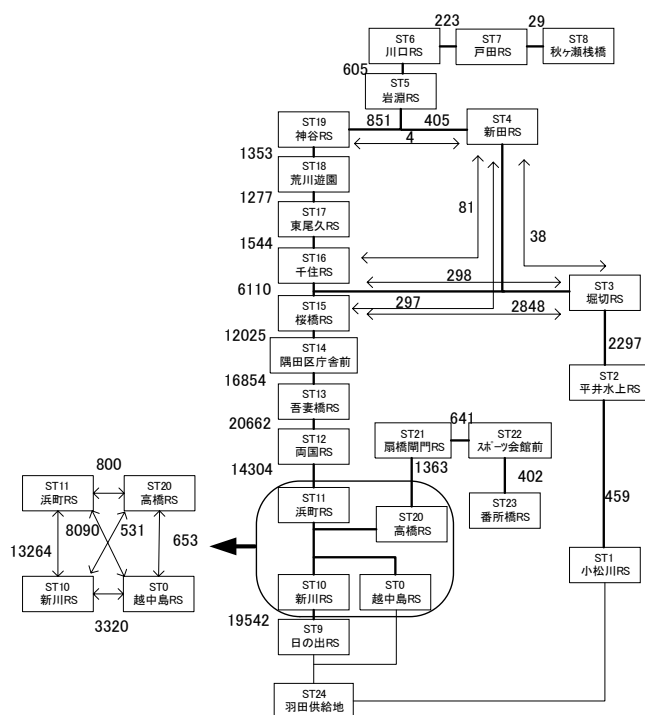


図2 ステーション間の移動量

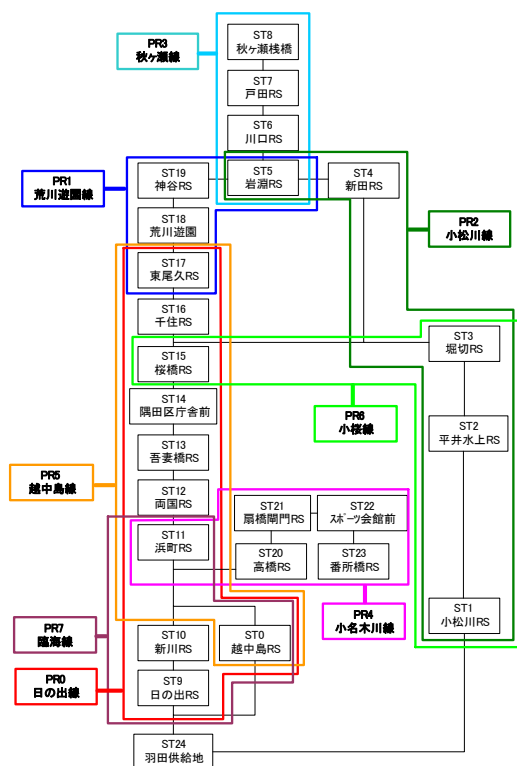


図3 設定路線

図2では「その2:被災者輸送需要の推計結果」に示したOD表と図1に示した輸送ネットワークから推計した、各ステーション間の移動量を示す。ここでクラス分けの指標として、隣接ステーション間の移動量に常用対数をとったCを用いることにする。これより、

C=1.0~2.7 : 秋ヶ瀬線(PR3),小名木川線(PR4)

C=1.0~3.2 : 小松川線(PR2)

C=2.7~3.2 : 荒川遊園線(PR1)

C=3.2~ : 小桜線(PR6),日の出線(PR0)

C=3.7~ : 越中島線(PR5)

その他 : 臨海線(PR7)

となり、クラス分けされた隣接ステーション間を結ぶ航路をまとめ図3の路線が得られる。なお、図2から隅田川、特に下流域の移動量が圧倒的に大きいことが分かる。

2.2 被災者の旅程決定アルゴリズム

被災者エージェントの旅程決定アルゴリズムは#1,#2の2種類考案した。アルゴリズム#1で目的の旅程が決定できない場合、#2が起動する。

(1) アルゴリズム#1の旅程決定方法

- ① 路線は通過順に並ぶST(ステーション)#で定義され、(各駅停車以外は考慮しない)各被災者輸送船にはどの路線に属するかが登録されている。
- ② 出発ST(ステーション), 目的STを結ぶ最短航路を抽出する。
- ③ ②の航路のSTを通過順に並べる。
- ④ ①の路線を構成するST#のデータから逆に、③で選んだSTが属する路線#を得、隣り合うSTの両方に共通する路線を記録する。その際、条件として路線が保持するST#も向きが同じ必要がある。(往復以外に図4-1、PR2のような循環路線も考慮するため)
- ⑤ ④で抽出された、路線の全組み合わせを記憶し、旅程候補とする。
- ⑥ ⑤の旅程候補で乗り換え回数が最小のものを選択するが、複数ある場合はすべての組み合わせを保持する。被災者は保持している旅程(路線)に合致する最初にやってきた船舶に乗

船する。

図4-1の場合、このアルゴリズム#1によりST1からST6へ向かう旅程として、
ST1-(PR1)→ST2-(PR2)→ST3-(PR2)→ST5-(PR3)→ST6
が得られる。

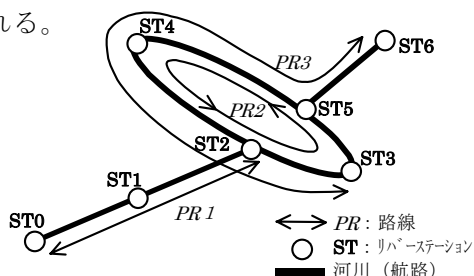


図4-1 路線、航路の模式図

(2) アルゴリズム#2の旅程決定方法

アルゴリズム#1では2つのステーションを結ぶ最短航路の利用を前提にしているため、下のような路線構成ではST0からST1へ向かう旅程が見出せない問題が起きる。

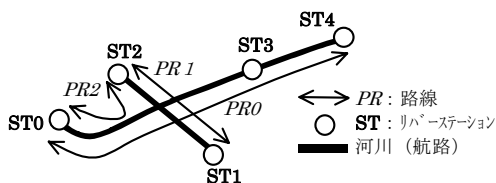


図4-2 路線、航路の模式図

この問題の解決には、航路（河川）上に構成される路線がつながってさえいれば、旅程を見出せるアルゴリズムが必要となる。以下では、そのアルゴリズム#2について図4-2を例として記述する。なお、このアルゴリズムでは乗り換えの際、一度利用した路線およびステーションを2度利用しないことを前提としている。

- ① ステーションを行に、路線を列に並べた行列を作成し、各路線に属するステーションのセルに0を入力する。
- ② 次に、出発STであるST0の行で0が記述された列内を走査し、ST0の行以外で0が記入してあるセルを1加算する。（この操作は乗船による移動を表現している。）
- ③ ②で1に変化したセルが属する列内で0が記

述されたセルに2(=1+1)を加算する。（この操作は乗り換えを意味している。）

- ④ ②③の操作を目的STであるST1の行内の0が他の数字に置き換わるまで繰り返す。

操作①の結果

		路線		
		PR0	PR0	PR1
ST0 ST1 ST2 ST3 ST4	ST0	0		0
	ST1		0	
	ST2		0	0
	ST3	0		
	ST4	0		

操作②の結果

路線		
PR2	PR1	PR2
0		0
	0	
	0	1
1		
1		

操作③の結果

		路線		
		PR0	PR0	PR1
ST0 ST1 ST2 ST3 ST4	ST0	0		0
	ST1		0	
	ST2		2	1
	ST3	1		
	ST4	1		

操作④の結果

路線		
PR2	PR1	PR2
0		0
	3	
	2	1
1		
1		

- ⑤ 最終的に得られた行列で、数字を昇順でたどることにより旅程が決定できる。上の場合、

No.0 : (ST0,PR0) (ST0,PR2)

No.1 : (ST2,PR2) (ST3,PR0) (ST4,PR0)

No.2 : (ST2,PR1)

No.3 : (ST1,PR1)

が得られ、この組み合わせのうち連続する3つのNo.から取り出した(ST, PR)が行列内で直角な位置関係であることが要求される。これより、最終的に以下の組み合わせだけが旅程の候補となる。

ST0→(PR2)→ST2→(PR1)→ST1

なお、複数の旅程がある場合は、さらに、最小距離となる旅程を採用する。

3. 解析

まず、図3で示される各路線を独立に解析する。例えば「路線PR0を独立路線として解析する」とはOD表で出発、目的STがPR0に属する移動量のみを入力して（他は0人）解析することを意味する。その後、図3の全路線を組み合わせた解析を行う。この際、各路線に投入する船舶数と輸送能力との関係を観察する指標として、飽和隻数を定義する。

3.1 飽和隻数

飽和隻数、 C_{sat} とは路線に就航できる有効な船舶

数で、それ以上船舶数を増加させても、輸送量の増加や所要時間の短縮が見込めない隻数を意味している。模式図を図5に示す。

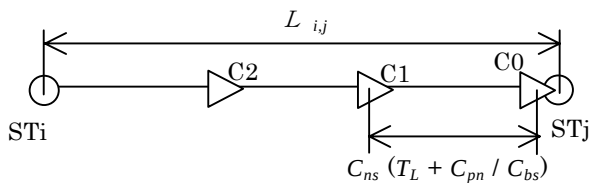


図5 飽和隻数導出の模式図

今、船舶C0がステーションSTjに到着したとすると、接岸時間+乗降時間の間、このステーションは他の船舶が利用できない状況になる。よって、次にSTjにやってくる船舶C1の最適な到着時刻はC0の離岸時刻であり、C1がC0と隔てる距離は、C0の接岸時間+乗降時間にC1の航行速度を乗ずればよい。つまり、ステーション間距離とC0,1間距離との比が飽和隻数となる。なお、往路復路で考えると飽和隻数は2倍になり、STi,j間に他のステーションがあってもこの関係は変化しない。飽和隻数を式で表せば、

$$C_{sat} = 2L_{ij} / \{C_{ns}(T_L + C_{pn} / C_{bs})\}$$

C_{sat} : ステーションi,j間の飽和隻数

L_{ij} : ステーションi,j間の距離

T_L : 接岸離岸時間

C_{pn} : 旅客定員

C_{bs} : 乗降速度

C_{ns} : 航行速度

なお、被災者輸送船に利用できる船種の候補として、屋形船、水上バスが挙げられる。関東地方における調査結果^③から、この船種についてまとめると、隻数:212隻,平均定員:103人(定員総数:21843人)というデータが得られる。なお、平均定員数は100人程度であるが、500人を越す船舶も8隻含まれている。

3.2 独立路線の解析

図3に示した各路線に投入できる船舶数を把握してみる。解析対象とする路線は、PR0,5,7の3路線とし、移動需要が最も大きな隅田川下流域のステーションを含む。この3路線は互いにステーシ

ョンを共有しているが、これを独立に解析してみる。よって、OD表で出発、到着、両STが解析対象路線内に含まれる移動量だけを利用する。また、ステーションに集合する被災者は4時間半(6:00~11:30)をかけて均一に集まるとした。図6には航路に就航させた船舶数と被災者が経験した平均所要時間を示す。

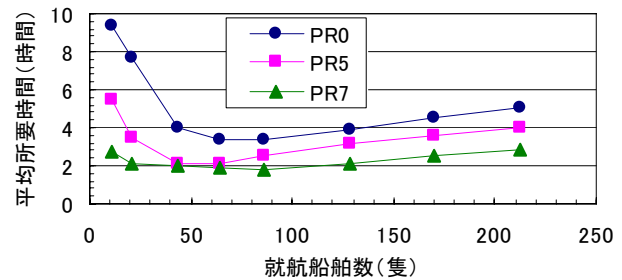


図6 独立路線解析の結果

図から各路線とも就航隻数が数十隻までは隻数の増加により所要時間も減少し、より早く移動が完了する。しかしながら、それ以上に船舶数を増加させると所要時間が増加し、不用意な船舶数の増加はかえって悪化を招くことが分かる。

表1 独立路線の予想結果

路線	L (km)	所属 ST数	T _c (分)	OD _i (人)	T ₁₀₀ (分)	ΔT _{sat} (分)	C _{sat} (隻)	C _{nd} (隻)	C ₃₀ (隻)	独立路線用 隻数
PR0	20.9	10	181	44148	0.6	3.3	54	295	6	54
PR1	8.2	4	68	1124	24	3.2	21	3	2	3
PR2	18.9	5	143	1411	19.1	2.9	49	7	5	7
PR3	14.1	4	106	155	174	2.9	37	1	4	4
PR4	5.8	5	58	522	52	3.8	15	1	2	2
PR5	13.7	8	124	24173	1.1	3.5	36	111	4	36
PR6	11.4	4	89	1550	17	3.0	30	5	3	5
PR7	8.1	5	73	21860	1.2	3.5	21	59	2	21
合計							262	482	28	132

L : 路線の片道距離

T_c : 路線の往復時間{=2Lx1000÷310(m/分=10knot)+接岸、乗降時間(2.5分×ST数)}

C_{sat} : 飽和隻数

OD_i : 独立路線内の被災者輸送人数

T₁₀₀ : 路線内全STの乗船待ち人数が100人となる時間=100÷(OD_i÷270分)

ΔT_{sat} : 飽和船団のSTへの到着間隔

C_{nd} : 路線内で乗船待ち人数が100人を越えないために必要な隻数

= T_c ÷ T₁₀₀

C₃₀ : 所属STに30分間隔で船舶が接岸するために必要な隻数= T_c ÷ 30

表1には各路線別に解析した飽和隻数を示す。なお、作表の際、次の仮定を行った。

- ステーション間は最短距離を航行し、航行速度は10knotとした。
- 離接岸時間の和は1分とし、船舶の平均定員数が100人であることから、乗降速度を100

人/分⁽⁴⁾として、乗降時間は1.5(分)とした。

- ・ 人の集合する時間は4.5時間(6:00～11:30)とし、この時間に一様に集合する。

これより、表内の飽和隻数はほぼ図6の所要時間が最小となる隻数に一致している。また表では、 C_{nd} (路線内で乗船待ち人数が100人を越えないために必要な隻数)及び C_{30} (所属STに30分間隔で船舶が接岸するために必要な隻数)も合わせて計算されており、最終列が各独立航路へ就航させる船舶数である。なお、この最終判断は次式による。

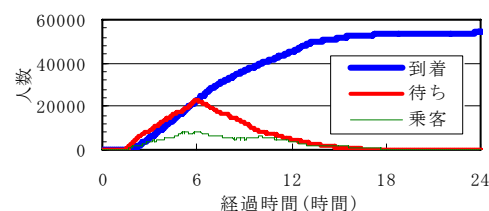
$$\text{独立路線の船舶数} = \min(C_{sat}, \max(C_{nd}, C_{30}))$$

3.3 被災者輸送解析

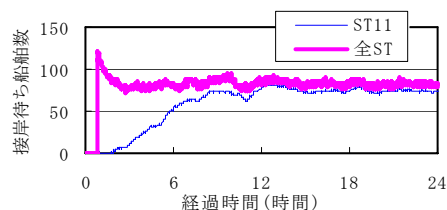
前節で独立路線に対する基本的な輸送量の傾向が解明された。これより全路線の解析を進める。この解析では、複数の路線に、同時に船舶が就航するため、各路線を航行する船舶の干渉が起こる。特にST11浜町では、4つの路線の船が接岸するため、大きな滞留が予想される。まず、表1の船舶数をそのまま適応してみる。解析時間は24時間とし、被災者は各STに6:30～11:00の4時間半の間一様に集合するとした。解析結果を図7に示す。なお、図中0時は午前5:00である。

ほぼ全員が24時間以内に目的地へ到達しているが、ST20からST1,2へ向かう被災者は目的地へ到着できなかった。旅程距離が長いことと、船舶が滞留するST11での待ち時間が長かったことが理由である。図(a)では乗船待ち人数が集合完了時刻11時頃(図では6時間)で2万人にも達する。この人数は11時まで単調増加していることから、時間あたりの集合人数と船舶への乗船人数の釣り合いがとれていないことになる。このつりあいをとるために必要な船舶数は表1の C_{nd} であり、隅田川の3路線以外はこの隻数を上回る設定である。よって、図中の乗船待ち人数はほぼ隅田川沿いのステーション内の人数となる。また、図(b)で示される通り、全ステーション合計の接岸待ち船舶数(船舶の滞留隻数)は80隻程度であるが、大多数はST11での接岸待ちであり、予想された現象が起きた。また、図(c)は所要時間の頻度分布で、多くの被災者が数時間で目的地へ到達している。この

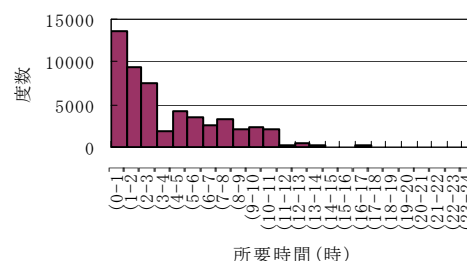
結果、全被災者の平均所要時間は3.8時間であった。



(a) 到着、乗船待ち、乗船人数



(b) 接岸待ち船舶数



(c) 所要時間の度数分布

図7 132隻就航時の解析結果

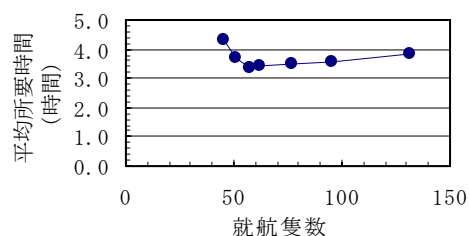


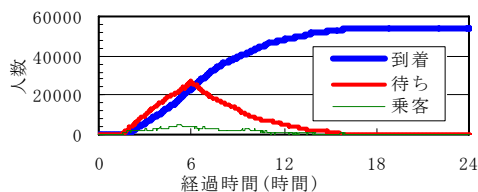
図8 全路線への就航隻数と所要時間の変化

そもそも、独立な路線による飽和隻数を基礎としたため、複数の路線が乗り入れるST11で大きな滞留が起きた。過剰な船舶数は図6で見られたように、かえって所要時間を増大させる。そこで、船舶数を減少させて所要時間の変化をしてみる。ST11を航行する路線以外では、大きな滞留が見られないため、就航船舶数を減少させる対象路線は隅田川の3路線、PR0,5,7のみとした。結果を所要時間と就航隻数の関係として図8に示す。図6と同様、極小値を持つ曲線が得られており、最適船

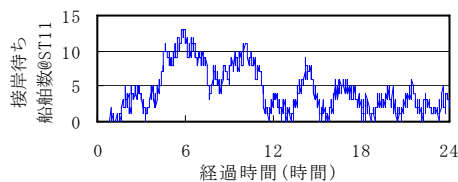
船舶数が 50～60 隻程度であることが示された。この時の平均所要時間は船舶数を半数以下にしたにもかかわらず、図 7 での 3.8 時間に対して 3.4 時間と短縮されている。就航船舶数を 57 隻としたときの各路線の船舶数を表 2 に、解析結果を図 9 にまとめる。図 9(b)の通り、ST11 での船舶の大きな滞留は改善され、図 9(a)では乗船者数が小さくなっている。これは、接岸待ち船舶の乗船者数の減少を意味している。また、図(c)の所要時間の頻度分布でも、短時間で目的地へ到着する人数が増加している。

表 2 最適な就航船舶数

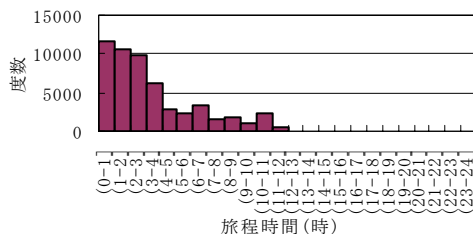
路線名	記号	独立航路解析 による就航隻数	最適就航隻数
日の出線	PR0	54	18
荒川遊園線	PR1	3	3
小松川線	PR2	7	7
秋ヶ瀬線	PR3	4	4
小名木川線	PR4	2	2
越中島線	PR5	36	11
小桜線	PR6	5	5
臨海線	PR7	21	7
計		132	57



(a) 到着、乗船待ち、乗船人数



(c) 接岸待ち船舶数



(d) 所要時間の度数分布

図 9 57 隻就航時の解析結果

4. まとめ

関東地方をモデルケースとして、54000 人あまりの被災者輸送シミュレーションを実施した。ここで設定した解析条件下では、全被災者の平均所要時間が 3～4 時間程度と推定され、定期船のような運航形態をとる場合、船舶の滞留が原因で被災者の所要時間が増加する現象が見られた。よって、むやみに船舶数を増加させても、かえって事態を悪化させることが判明し、最適な船舶数が存在することが明らかになった。現状の施設を利用することを前提とした場合、本条件下で平均所要時間が最小になる船舶数は東京湾周辺に在籍する、屋形船、水上バスなどで満足できる隻数であった。また、輸送を滞らせる大きな原因は隅田川下流域のリバーステーションに集中する被災者の乗降速度であることが分かった。

謝 辞

本研究は国土交通省技術研究開発委託費（総合政策局）により実施しました。関係各位に深く感謝します。

【参考文献】

- (1) 間島隆博, 勝原光治郎, 他, 複雑系マルチエージェントシミュレーションによる河川を利用した災害時緊急輸送能力の評価、日本造船学会論文集、192 号、pp. 465-474、(2002)
- (2) 間島隆博, 勝原光治郎, 他, 河川を利用した災害時緊急輸送シミュレーション、第 2 回海上技術安全研究所 研究発表会講演集、pp. 89-94、(2002)
- (3) 関東運輸局船舶部、災害時緊急輸送における河川航行船の利用可能性調査報告書、平成 14 年 3 月、(2002)
- (4) 中村 幸史、通勤電車運行スケジュールにおける遅延計算モデルの構築、<http://www.ise.chuo-u.ac.jp/ise-labs/taguchi-lab/pdf/a02n8100038.pdf>、(2005 年 4 月現在)