衝突回避を考慮した避難スケジューリング問題

2015SS095 吉田朱里

指導教員：佐々木美裕

1 はじめに

災害が発生すると多くの人が一斉に非難する．愛知県帰宅困難者対策実施要領では, 人々の混乱防止と安全で円滑な避難をするためにむやみに移動を開始しないことを基本原則としている [1]. 避難時にはただ迅速に移動するのではなく, 安全に人々が移動するために計画的なスケジュールに沿って移動することが必要であると考えられる.

三浦 [2] は, 交通網の整備や交通信号によって交通の流動を制御することでより円滑な交通の流動を実現するため, 格子状交通網モデルを用いて, 起点から終点間の経路の流動に左折や右折の流動配分ルールを設定し, OD 間の経路への流動配分と流動交差合流量の関係について解析している.

本研究では, 各避難者の避難開始地点 (起点) と目的地点 (終点), およびその避難経路を所与とし, 避難者の移動中の衝突を回避する避難のスケジューリング問題をネットワーク上で考える.

2 避難スケジューリング問題

ネットワーク上のノードは道路網の交差点, 枝は交差点間の道路に対応する．ネットワーク上の複数の起終点ペアと起終点間の経路 (パス) は所与とする．起点は学校や会社などに, 終点は避難地に対応し, 起点から終点へ避難する際の人々の流れを以下ではフローと呼ぶ．同じ時刻に複数のフローが同じノード上を通過することをフローが衝突するという．フローが衝突することなく移動し, 全フローの移動が完了する時刻を最小にするようなモデルを考える．ここで, 進み方の異なる 2 つのモデルとして待機ありモデルと待機なしモデルを提案する．待機ありモデルでは, 他のフローとの衝突を避けるために, 起点から終点へ移動する際に途中で待機することを許す．一方, 待機なしモデルでは, 途中で待機することを許さず, 起点を出発したら終点に向かって止まることなく進む.

3 定式化

衝突回避を考慮した避難スケジューリング問題 (待機ありモデルと待機なしモデル) を定式化するために, 以下の記号を定義する.

- N： ノードの集合.
- P： パスの集合.
- l_p： パス $p \in P$ に含まれるノードの数 (起点と終点を含む).
- T： 時刻, $T = \sum_{p \in P} l_p$
- M： 大きな値.

$$v_{pij} = \begin{cases} 1: & \text{パス } p \in P \text{ の } i \text{ 番目の} \\ & \text{ノード } (i = 1,\ \ldots,\ l_p) \text{ が } j \in N \text{ である.} \\ 0: & \text{上記以外.} \end{cases}$$

以下のように変数を定義する.

$$x_{pit} = \begin{cases} 1: & \text{パス } p \in P \text{ の } i \text{ 番目のノード上をフロ} \\ & \text{ーが時刻 } t \text{ に通過する.} \\ 0: & \text{上記以外} \end{cases}$$

$$y_{pt} = \begin{cases} 1: & \text{パス } p \in P \text{ 上のフローが時刻 } t \text{ に起点を} \\ & \text{出発する.} \\ 0: & \text{上記以外} \end{cases}$$

待機なしモデルは以下のように定式化できる.

$$\text{min.} \quad \sum_{p \in P} \sum_{i=1}^{l_p} \sum_{t=1}^{T} t x_{pit} \tag{1}$$

$$\text{s.t.} \quad \sum_{t=1}^{T-l_p+1} x_{pit} = 1,\ (p \in P,\ i = 1,\ \ldots,\ l_p) \tag{2}$$

$$\sum_{t=1}^{T-l_p+1} y_{pt} = 1,\ (p \in P) \tag{3}$$

$$\sum_{i=1}^{l_p} \sum_{k=t}^{t+l_p-1} x_{pit} \geq l_p y_{pt},\ (p \in P,\ t = 1,\ \ldots,\ T) \tag{4}$$

$$\sum_{k=i+1}^{l_p} \sum_{l=1}^{t} x_{pkl} \leq M(1 - x_{pit}),$$
$$(p \in P,\ i = 1,\ \ldots,\ l_p,\ t = 1,\ \ldots,\ T) \tag{5}$$

$$\sum_{p \in P} \sum_{i=1}^{l_p} v_{pij} x_{pjt} \leq 1,\ (t = 1,\ \ldots,\ T,\ j \in N) \tag{6}$$

$$x_{pit} \in \{0, 1\},\ (p \in P,\ i = 1,\ \ldots,\ l_p,\ t = 1,\ \ldots,\ T) \tag{7}$$

$$y_{pt} \in \{0, 1\},\ (p \in P,\ t = 1,\ \ldots,\ T) \tag{8}$$

(1) は各パス上のフローが各ノードを通過する時刻の総和を示し, これを最小にすることを目的とする．(2) は各パス上のフローが各パスのノード上を必ず 1 回のみ通過するという制約である．(3) は各パス上のフローがいずれかの時刻に出発することを示す制約である．(4) は各パス上のフローは出発後, 途中で待機することなく連続して進むことを表す制約である．(5) は時刻 t にパス $p \in P$ 上のフローが i 番目のノード上を通過するとき, このパス上の $i+1$ 番目以降のノードを時刻 t 以前に通過することはできないことを示す制約である．(6) は時刻 t においてノード $j \in N$ を通過することができるパス数は 1 以下を示す制約である．(7) は x_{pit} のバイナリ変数制約である．(8) は y_{pt} のバイナリ変数制約である.

待機ありモデルは以下のように定式化できる．

$$\begin{array}{ll} \text{min.} & (1) \\ \text{s.t.} & (2),\ (5)-(7) \end{array}$$

4　発見的解法

待機なしモデルに対する発見的解法について説明する．はじめに, 全てのフローが時刻 1 に出発すると仮定し, 各時刻において同じノード上を通過するフローがあれば, いずれかのパスの出発時刻を遅らせ, フローの衝突を回避する．すべての時刻においてフローの衝突が発生していないか確認し, フローの衝突が 1 回以上発生していた場合, 再び時刻 1 から各時刻においてフローの衝突が発生していないか確認する．各時刻においてフローの衝突が 1 回も発生してなければ終了．発見的解法を作成するために第 3 節で定義した記号に加え以下の記号を定義する．

$$flg = \begin{cases} 1 : \text{フローの衝突が 1 回以上発生した.} \\ 0 : \text{フローの衝突が 0 回.} \end{cases}$$

入力

u_{pi} ：パス $p \in P$ 上のフローを i 番目に通過するノード番号. $(i = 1,\ 2,\ \ldots,\ l_p,\ u_{pi} \in N)$

出力

h_{pi} ：パス $p \in P$ の i 番目のノード上を通過する時刻.

s_p ：パス $p \in P$ 上のフローが移動を開始する時刻.

e_p ：パス $p \in P$ 上のフローが時刻 t に通過するノード番号. $(n_{pt} \in N)$

待機なしモデルの発見的解法を以下のように記述できる．

ステップ 0　$h_{pi} := i,\ (p \in P,\ i = 1,\ 2,\ \ldots,\ l_p)$.
$s_p := 1,\ e_p := l_p,\ flg := 0,\ t := 0$.

$$n_{pt} := \begin{cases} u_{pt}, & (p \in P,\ t = 1,\ \ldots,\ l_p). \\ 0, & (p \in P,\ t > l_p). \end{cases}$$

ステップ 1　$t := t + 1,\ P_1 := P$.

ステップ 2　$k \in P_1$ を選択する．$P_2 := P$.

ステップ 3　$m \in P_2$ 選択する $(m > k)$.

ステップ 4　$n_{kt} = n_{mt}$ ならば, $s_m := s_m + 1,\ e_m := e_m + 1,\ h_{mi} := h_{mi} + 1(i = 1,\ \ldots,\ l_m),\ n_{mj} := n_{m(j-1)} + 1(j = e_m,\ \ldots,\ 2),\ n_{m1} := 1,\ flg := 1$. $n_{kt} \neq n_{mt}$ ならば $flg = 0$.

ステップ 5　$P_2 := P_2 \setminus \{m\}$ とし, $P_2 \neq \phi$ ならばステップ 3 へ．$P_2 = \phi$ ならば $P_1 := P_1 \setminus \{k\}$ としステップ 2 へ．$P = \phi$ かつ $t = T$ ならばステップ 6 へ．$P = \phi$ かつ $t \neq T$ ならばステップ 1 へ.

ステップ 6　$t = T$ のとき $flg = 0$ ならば $h_{pi},\ s_p,\ n_{pt}$ を出力し終了．そうでないなら $t := 0$ としてステップ 1 へ.

5　実行結果と考察

待機なしモデルの最適解を Gurobi Optimizer 7.5.1 を用いて求めた結果と Python で実装した発見的解法を用いて得られた解を比較する．使用したデータを表 1 に示す．このデータは 5 つのパスを含み, 「ノード番号」の列に記されている番号は, 各パスが通過する順に示している．例えば, パス 1 はノード 10 を起点とし, その後, 15, 14, 13 の順に通過し, 終点が 12 である．使用した計算機の計算環境は CPU : Intel(R)Core(TM)i7-6700 CPU @ 3.40GHz 3.40GHz, OS ; Windows10, メモリは 64.0GB である．計算時間は, 最適化を行った場合は 0.86 秒, 発見的解法を用いた場合は 0.031 秒であった．実行結果を, 表 2, 3, に示す．表 2, 3 は, 各時刻に対して各パス上のフローが通過するノード番号を示している．待機なしモデルの最適解において, 避難完了時刻が 10 であるのに対し, 提案した発見的解法で得られた解では 13 となった.

表 1　例題

	ノード番号								
パス 1	10	15	14	13	12				
パス 2	4	9	14	19	18	17	16	21	
パス 3	8	13	14	15	20	25			
パス 4	4	3	8	13	18	23			
パス 5	6	11	16	17	18	13	8	3	2

表 2　待機なしモデルの最適解

時刻	1	2	3	4	5	6	7	8	9	10	11	12	13
パス 1	10	15	14	13	12								
パス 2			4	9	14	19	18	17	16	21			
パス 3		8	13	14	15	20	25						
パス 4		4	3	8	13	18	23						
パス 5	6	11	16	17	18	13	8	3	2				

表 3　待機なしモデルを発見的解法を用いて解いた結果

時刻	1	2	3	4	5	6	7	8	9	10	11	12	13
パス 1	10	15	14	13	12								
パス 2		4	9	14	19	18	17	16	21				
パス 3				8	13	14	15	20	25				
パス 4			4	3	8	13	18	23					
パス 5					6	11	16	17	18	13	8	3	2

6　まとめ

ネットワークにおいてフローの衝突を回避するスケジュールの最適化問題の待機ありモデルと待機なしモデルを定式化し最適解を求めた．さらに待機なしモデルの発見的解法を提案した．そして, 待機なしのモデルについて最適解と発見的解法の解を比較した.

参考文献

[1] 愛知県：愛知県帰宅困難者対策実施要領（平成 27 年 3 月改訂）, 2015.

[2] 三浦英俊：格子状交通路モデルを用いた交通流動の交差と合流の解析, 日本オペレーションズ・リサーチ学会 2018 年秋季研究発表アブストラクト集, 2-A-1, 2018.