

旅行ユニットを考慮した 飛行機搭乗方式の性能評価



名古屋工業大学 工学部
中野 聡美

序論

私たちは、国内や海外への移動に飛行機を頻繁に利用する

しかし、飛行機に乗り込む際の問題はしばしば最適な方法で処理されていない

多くの航空会社は、搭乗時間が無駄に長くなっている

ターンアラウンドタイム

飛行機が空港に到着してから乗客を降ろすことや、機内の清掃、燃料の品質チェック・補給、乗客の搭乗などの全ての時間が含まれている

ターンアラウンドタイムの短縮→コスト削減

序論（過去の文献結果）

▪ Nyquist&McFadden(2008)

ターンアラウンドタイム1分短縮で毎回30ドルの節約

▪ Steiner.A,&Philipp.M.(2009)

1分あたり77ドルの節約

▪ Horstmeier.T.,&Haan,F.d.(2001)

1分あたり250ドルの節約

→環境や飛行機の大きさ等によって差は生じるが、コスト削減に直結

主な搭乗方法

① Random

搭乗順が定められていない

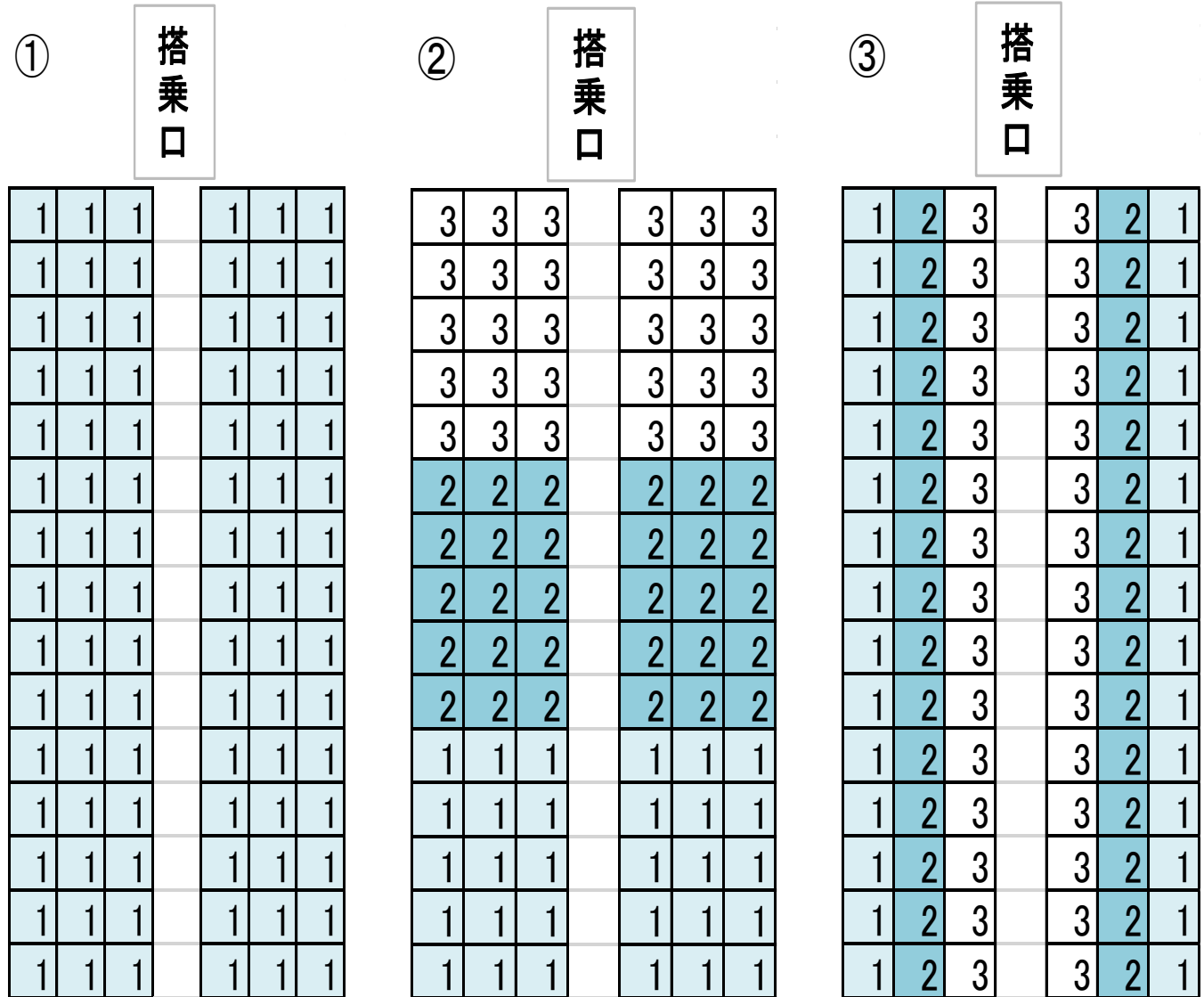
② Back-to-front

後ろから飛行機の前へ

グループ毎に搭乗

③ Outside-in

窓側、中央、通路側の順に搭乗



主な搭乗方法

Reverse-pyramid

Outside-inとback-to-frontを掛け合わせた

最近の研究では搭乗時間が一番早いとされている

搭乗順が複雑であり、

一緒に来た団体客が別々に搭乗することになるため

本研究では扱わないとする

			搭乗口			
3	4	5		5	4	3
3	4	5		5	4	3
3	4	5		5	4	3
3	4	5		5	4	3
2	3	5		5	3	2
2	3	5		5	3	2
2	3	5		5	3	2
2	3	5		5	3	2
1	3	5		5	3	1
1	3	5		5	3	1
1	2	4		4	2	1
1	2	4		4	2	1
1	2	4		4	2	1
1	2	4		4	2	1
1	2	4		4	2	1

搭乗問題の研究論文の本質的な目的・・・搭乗時間を短くする

しかし、ほとんどの研究では旅行客を個人単位としてしか扱っておらず、2, 3人などの旅行グループ(以下旅行ユニット)を考慮していない

→ 実用化が難しい

Steiner.A,&Philipp.M.(2009)

- ・旅行ユニット考慮しrandomとback-to-frontの2種類について研究
- ・「乗客の手荷物が多い」、「間違ったチケットの交換」または「遅刻した乗客を待つ」など、遅れの要因に焦点
- ・搭乗方法そのものの比較は行われていない

本研究の目的

① Back-to-frontにおいて、
搭乗グループのグループサイズが変化したときの搭乗時間の比較

② 旅行ユニットを考慮する場合としない場合の既存の搭乗方法についての比較

③ 旅行ユニットの割合を変化させたときの搭乗時間の比較

干渉の種類

搭乗時間の短縮 → 干渉数の最小化
干渉は主に、「座席干渉」と「通路干渉」

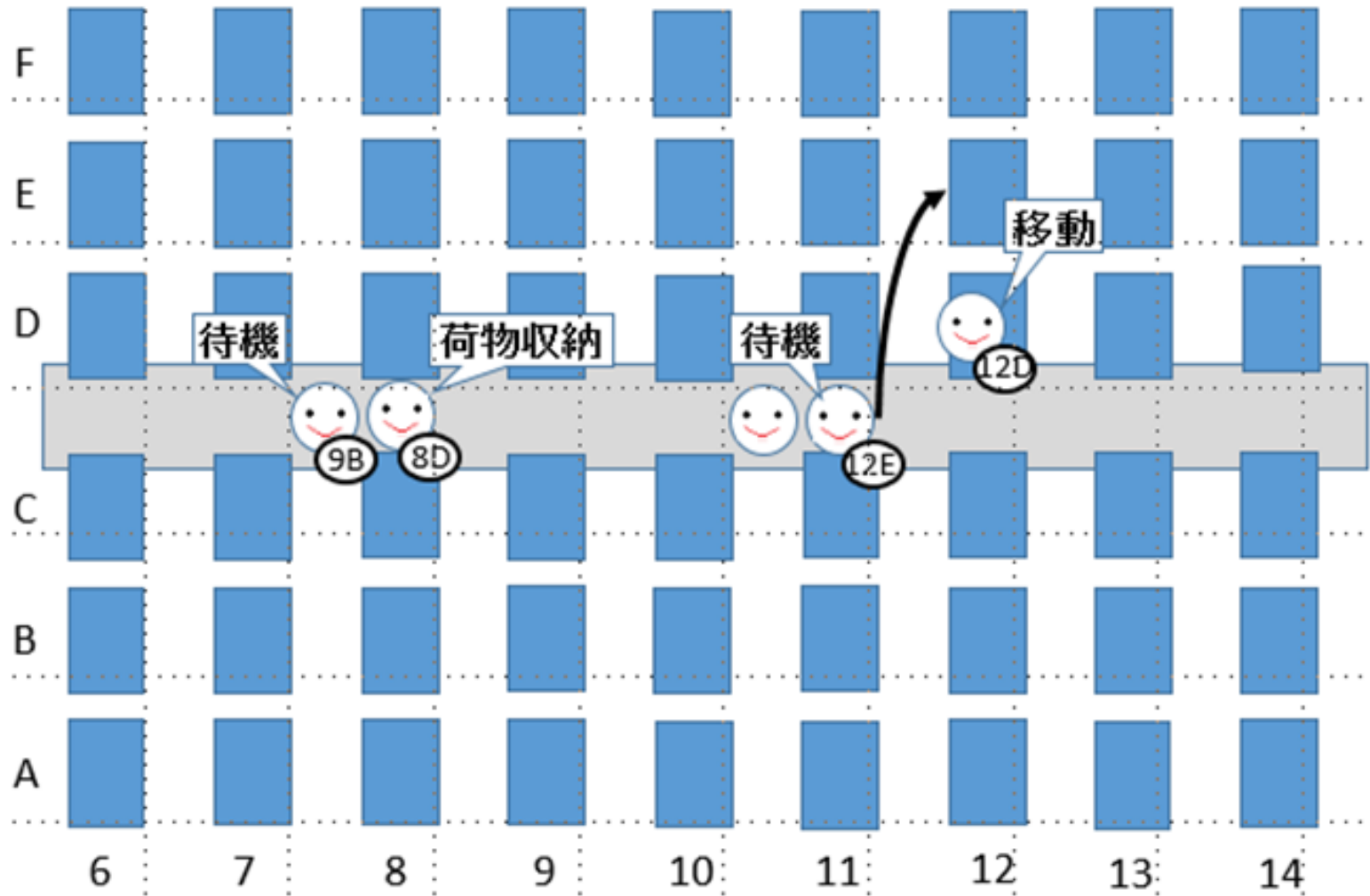
座席干渉

同じ側の同じ列である、窓側
また中央の座席の乗客が、
真ん中もしくは通路側の座席
の乗客よりも遅い時に起こる

通路干渉

2人以上の乗客を伴い、手荷物を詰める人を待ったり、ある乗客が席に座りたくて通路側に座っている他の乗客を立たせて移動する行為を待ったりすることにより通路がふさがれる時に起こる

図① 飛行機内の干渉モデル

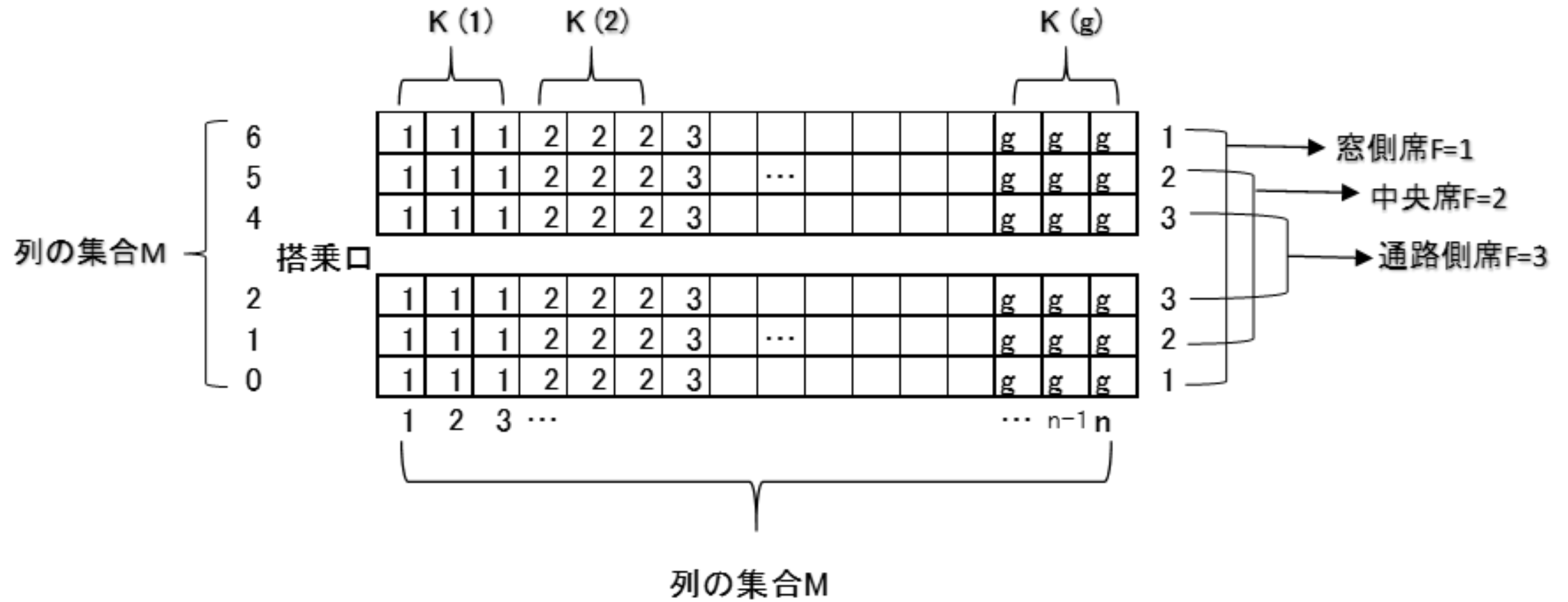


モデルの概要

飛行機のモデル

- エアバス319および320、ボーイング737および757のような通路の両側に3席を有する飛行機
- $N = \{0, 1, 2, \dots, n-1\}$ は列の集合, $M = \{0, 1, 2, 4, 5, 6\}$ は同じ列に並ぶ席の集合 ($M(3)$ は通路)
- $M = \{0, 6\}$ は窓側席に値するもので $F=1$ と定義、 $M = \{1, 5\}$ は中央席で $F=2$ 、 $M = \{2, 4\}$ は通路側席で $F=3$ とする
- $x \in N, y \in M$ とすると、座席位置は (x, y)
- 搭乗グループの集合を $K = \{1, 2, 3, \dots, g\}$

図② 飛行機のモデル図



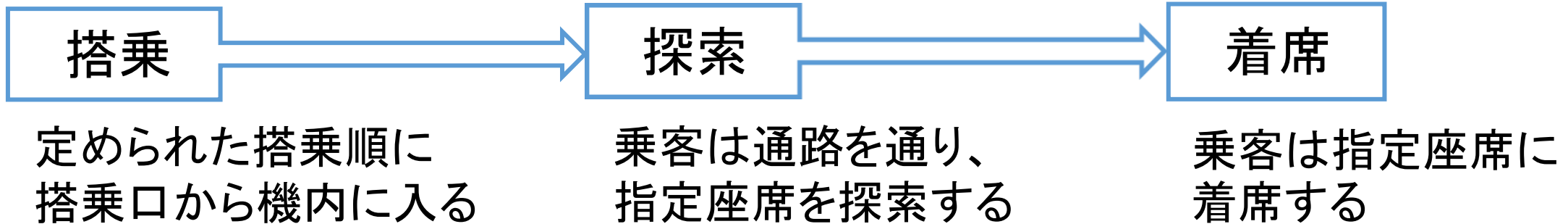
乗客のモデル

- ・各乗客に、座席位置(x軸とy軸)とランダムに振り分けられた旅行ユニットが与えられる
- ・旅行ユニットの割合[2]・・・

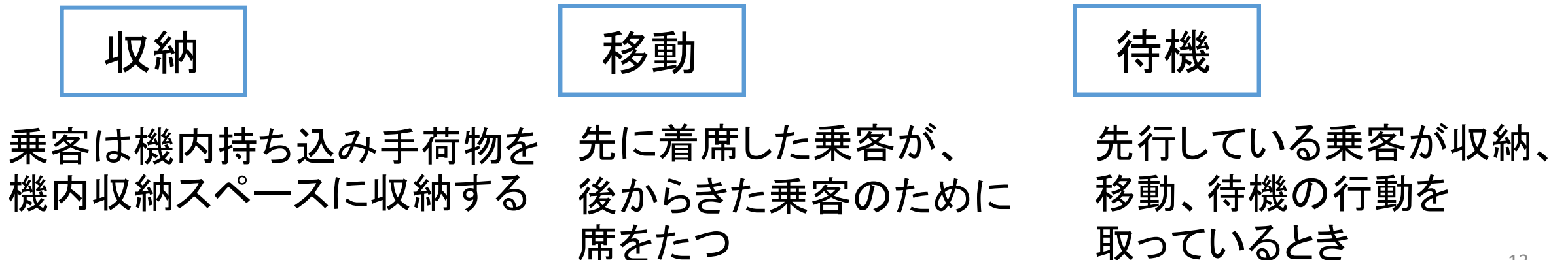
3人ユニット	7%	(9人)
2人ユニット	38%	(46人)
単身	55%	(65人)
- ・旅行ユニットは通路を挟まない隣り合った座席を与える
- ・全乗客の約6割が手荷物を持っている
(4割が1つ、2割弱が2個以上の手荷物をもつ)

乗客の行動

全乗客がとる行動[4]



一部の乗客がとる行動



シミュレーション

背景

- ・エアバス319および320、ボーイング737および757のような小型航空機を想定
- ・搭乗口は1つ
- ・座席は、列数を $N=20$ [列]、座席数を $M=6$ [席](通路を挟み3席ずつ)の計120席
- ・空席はない
- ・全乗客は搭乗ゲートにすでに集合しており、遅刻等の違反をする乗客はいない
- ・搭乗を開始した乗客は一定間隔で搭乗
- ・収納の行動をとるエージェントは全体の6割
- ・収納時間(15秒~100秒)(Steffen,J.H.,[5])
席を立ってもらう時間(5秒~10秒)の分布は一様分布で設定

各乗客に多種のIDを配列形式で指定、30パターン作成する

$((x, y), K, \text{secretID}, F, i)$

- x : x 座標
- y : y 座標
- K : 搭乗グループ
- secretID : 隠れ乗客ID、旅行グループをランダムで生成する際に使用
- F : 座席位置の区別(窓側、中央、通路側)
- i : 旅行ユニットごとのID

プログラム説明

```
# 乗客は60%の確率で荷物を持っている(確率を変える場合には数字0.6を変更)
if numpy.random.random() <= 0.6:
    self.carry=1 # 荷物を持っている状態は1が設定
    self.screenMarker='^' # 荷物を持っている状態のエージェントの形は三角形
else:
    self.carry=0 # 荷物を持っている状態は0が設定
    self.screenMarker='o' # 荷物を持っていない状態のエージェントの形は丸
```

毎回ランダムに6割の乗客が荷物を持っている状態にする

シミュレーション画面では荷物を収納するまで乗客を三角、収納し終わった乗客と元々荷物を持っていない乗客を丸で表す



前にいる乗客につづいて進んだり待ったりしている状態

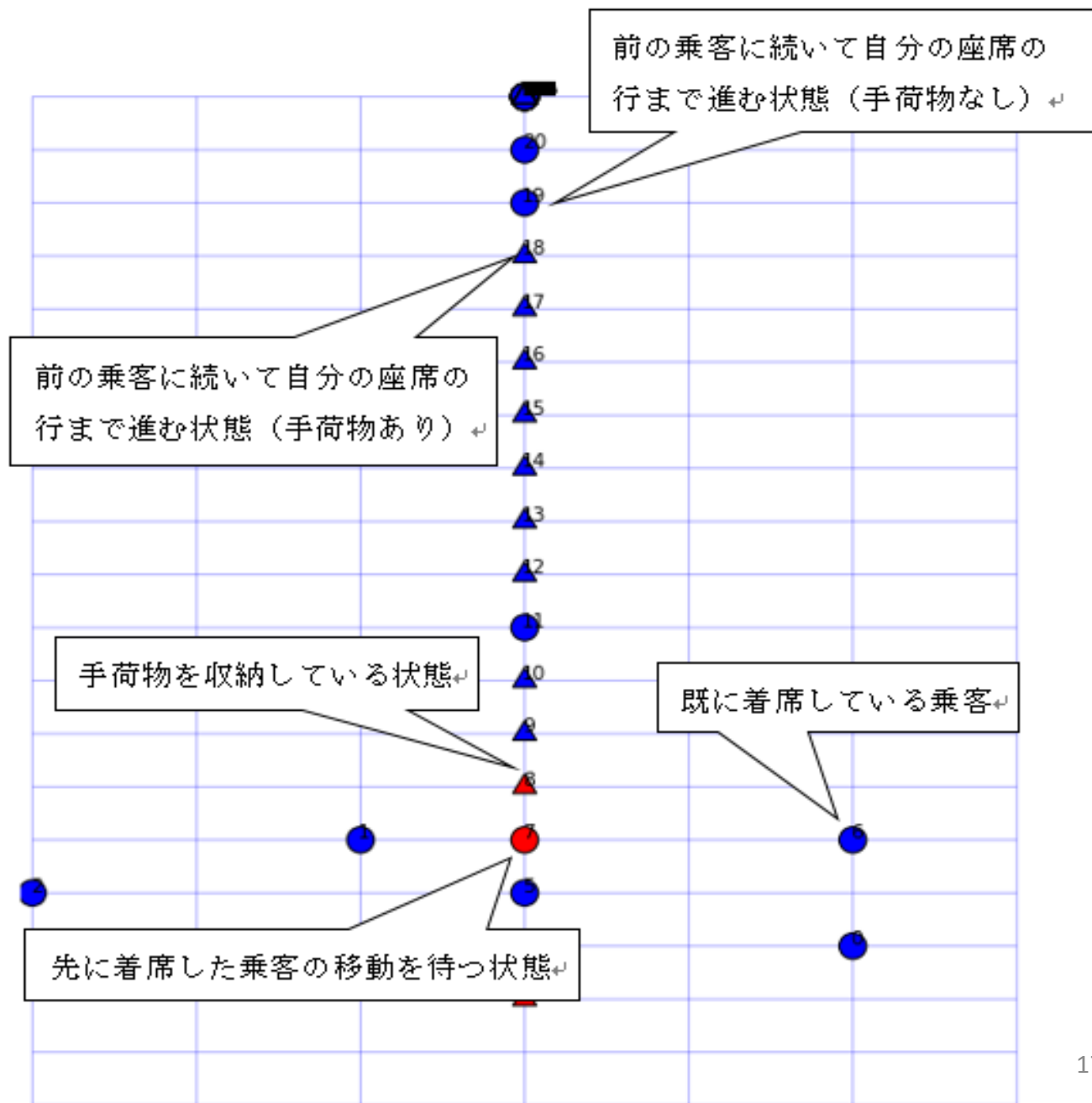


自分の座席の行に到着し、先に席についている自分よりも通路側にいる乗客が移動するのを待機している状態



手荷物を収納している状態

図③ シミュレーション画面



シミュレーション結果

結果①

比較対象となる通常のback-to-front , random , outside-inの平均搭乗時間
(back-to-frontの搭乗グループサイズは7:7:6)

搭乗方法	back-to-front	Random	outside-in
平均搭乗時間	1132.1	951.1	743.6

→ 一般的な論文結果の通り、
outside-inの平均搭乗時間は圧倒的に短い

結果②

back-to-front、搭乗グループサイズの割合変化
旅行ユニットは考慮なし

x,y,18	x,y,17	x,y,16	x,y,15	x,y,14	x,y,13	x,y,12	x,y,11	x,y,10	x,y,9	x,y,8	x,y,7	x,y,6	x,y,5	x,y,4	x,y,3	x,y,2	x,y,1	
859.8	903.1	927.1	942.9	962.2	970.6	970.1	964.1	970.8	953.8	1004.4	957.0	915.5	918.2	907.9	835.7	874.5	854.6	1,y,z
	887.7	927.6	969.6	978.8	981.7	1016.4	986.5	968.3	1014.8	986.6	979.4	979.0	937.4	910.1	856.0	864.8	859.7	2,y,z
		947.7	975.1	961.9	996.2	1012.9	1038.8	1024.7	1045.6	1003.2	1019.9	1036.8	999.7	974.6	862.5	894.8	866.3	3,y,z
			967.3	927.0	1009.0	1043.0	985.0	1166.0	1058.0	988.0	1039.5	1028.6	1047.0	986.6	951.5	872.8	950.0	4,y,z
				920.5	981.0	1070.0	1094.0	1084.0	1136.0	1088.0	829.0	956.0	1034.0	1083.0	923.0	985.0	955.0	5,y,z
					983.0	1062.0	1037.0	1072.0	1080.0	1139.0	1063.0	1028.0	935.0	1109.0	999.0	850.0	928.0	6,y,z
						979.0	1053.0	1058.0	1067.0	1003.0	972.0	1073.3	1063.6	1063.3	1008.9	972.6	958.9	7,y,z
							938.0	1056.0	1041.0	1004.0	1096.0	1099.0	1077.0	1075.0	1011.0	1118.0	948.0	8,y,z
								948.0	992.0	1065.7	1029.0	998.0	1044.6	846.0	1218.0	986.0	962.0	9,y,z
									951.0	1118.0	1074.0	1147.0	1026.0	1062.0	1028.0	1025.0	1020.0	10,y,z
										950.0	1020.0	902.0	1070.0	1152.0	1225.0	888.0	1060.0	11,y,z
											1069.0	1043.0	1139.0	936.0	1111.0	886.0	1056.0	12,y,z
												1043.0	1052.0	1180.0	1289.0	1066.0	936.0	13,y,z
													979.0	1042.0	1041.0	984.0	1062.0	14,y,z
														1009.0	1037.0	999.0	1195.0	15,y,z
															931.0	857.0	943.9	16,y,z
																972.0	1035.0	17,y,z
																	977.0	18,y,z

x、y、zのどれか1つのグループに搭乗客を偏らせる場合に搭乗時間を抑えられる

結果③

T分布による、95%信頼区間の推定結果

	標本平均 $\pm\alpha$ (信頼区間)
outside-in	743.6 \pm 21.7
back-to-front(1,16,3)	835.7 \pm 25.1
random	951.1 \pm 28.5
back-to-front(7,7,6)	1073.3 \pm 32.3

Outside-in < back-to-front (ex(1,16,3)) < Random < back-to-front (ex(7,7,6))

→ 95%信頼区間が被ることなく、
順番が前後することはほぼない

結果④

旅行ユニットを考慮した、outside-inとrandomの平均搭乗時間

	Outside-in	random
旅行ユニット 考慮なし	743.6 ↓	951.1 ↓
旅行ユニット 考慮あり	812.0	906.3

- outside-inは搭乗時間が長くなる
搭乗グループ1のサイズが大きくなり、
通路干渉が増えたことが原因
- randomは搭乗時間が短くなる
旅行ユニット内は座席干渉が起こらなくなり
全体的な干渉数の減少が原因

結果⑤

back-to-front、搭乗グループサイズの割合変化 旅行ユニットを考慮

x,y,18	x,y,17	x,y,16	x,y,15	x,y,14	x,y,13	x,y,12	x,y,11	x,y,10	x,y,9	x,y,8	x,y,7	x,y,6	x,y,5	x,y,4	x,y,3	x,y,2	x,y,1	
867.4	931.6	968.2	959.2	991.3	1011.6	1004.5	1015.1	1015.4	1000.8	1021.1	992.6	956.7	962.0	936.4	901.8	892.1	897.3	1,y,z
	952.2	987.7	1027.3	1028.9	1053.4	1022.6	1038.2	1057.9	1059.8	1036.7	1052.7	1013.9	967.3	969.1	946.0	903.8	890.7	2,y,z
		976.4	1036.3	1004.5	1062.6	1075.3	1103.6	1065.1	1076.9	1101.6	1063.4	1034.1	1039.4	968.2	971.6	960.4	935.4	3,y,z
			1008.7	993.5	1073.2	1102.6	1111.4	1099.3	1088.8	1058.9	1104.9	1095.9	1075.3	1022.7	1017.0	952.9	966.2	4,y,z
				1037.7	1055.2	1117.7	1076.2	1127.5	1128.4	1125.7	1098.7	1070.1	1098.0	1044.8	1038.0	977.7	977.8	5,y,z
					1039.4	1079.6	1093.4	1112.6	1088.3	1090.1	1122.5	1083.1	1059.2	1081.5	1050.6	1028.7	1035.8	6,y,z
						1031.6	1074.7	1106.1	1106.5	1128.2	1133.4	1087.3	1108.8	1039.2	1068.8	1009.9	1006.5	7,y,z
							1048.8	1099.9	1127.2	1107.3	1124.5	1101.8	1121.1	1118.7	1047.6	1025.7	1017.8	8,y,z
								1025.6	1086.7	1134.5	1113.9	1105.9	1110.7	1072.9	1054.5	1042.8	1025.3	9,y,z
									1060.9	1085.2	1069.6	1082.8	1101.5	1095.9	1028.1	1042.6	1021.2	10,y,z
										1041.5	1069.4	1087.9	1073.1	1107.2	1075.1	1045.0	1012.7	11,y,z
											1060.6	1041.5	1057.2	1078.6	1066.6	1048.3	1023.1	12,y,z
												1037.0	1020.2	1102.3	1061.1	1046.7	1012.3	13,y,z
													997.6	1022.3	1048.0	1033.3	1007.2	14,y,z
														1035.1	1002.2	990.2	969.9	15,y,z
															973.7	988.5	992.4	16,y,z
																945.4	964.8	17,y,z
																	934.5	18,y,z

結果⑥

T分布による、95%信頼区間の推定結果

	標本平均± α (信頼区間)
outside-in	811.9±26.5
back-to-front(1,1,18)	867.4±44.2
Random	906.3±28.3
back-to-front(7,7,6)	1087.3± 179.9

Outside-in < Back-to-front ex (1,1,18) < random < Back-to-front ex (7,7,6)

→ 旅行ユニットを考慮しない場合と
搭乗時間の早い順番は変わらない
信頼区間が被る部分が出てくる

旅行ユニットの割合変化

- ① 3人ユニットが多いケース(卒業シーズンなど、団体の旅客が多い場合)

1人:2人:3人=37人:44人:39人

- ② 2人ユニットが多いケース(クリスマスシーズンなど、ツインの旅客が多い場合)

1人:2人:3人=45人:66人:9人

- ③ 1人ユニットが少ないケース(ツアーなど団体で申し込むことが多い場合)

1人:2人:3人=36人:66人:18人

結果⑦

旅行ユニットの割合を変化させた場合の、
random,outside-inの平均搭乗時間

	ran	OI
①	921.8	851.1
②	914.5	834.8
③	974.8	843.8

→ outside-inはどの割合でも
搭乗時間はあまり変化が見当たらない
Randomは割合によっての変化が大きい

結論

飛行機の搭乗方式について、
S4シミュレーションシステムを用いて比較を行った。

飛行機の搭乗方式のうちback-to-frontについて、
3つのグループの客数比により
ターンアラウンドタイムが大きく異なることが分かった

旅行ユニットを考慮する場合、
outside-inが圧倒的に優れているとはいえ、
1つの搭乗グループに乗客を偏らせたback-to-frontとの差が
小さくなることがわかった。

今後の課題

① 空席がある場合、それぞれの搭乗方法で変化があるのかを調べる

② もっと大きな車体、通路が二つある場合など、想定環境を変える

③ 旅行ユニットを考慮した、outside-inよりも平均搭乗時間の短い新しい搭乗方法の探索

参考文献

Nyquist,D.C.,&McFadden,K.L(2008)

A study of the airline boarding problem.journal of Air Transport Management,14(4),197-204.

Steiner.A,&Philipp.M.(2009)

Speeding up the airplane boarding process by using pre-boarding areas.In:9th Swiss Transport Research Conference,September 9-11,2009,Monte Verita/Ascona.

Horstmeier.T.,&Haan,F.d.(2001)

Influence of ground handling on turn round time of new large aircraft.Aircraft Engineering and Aerospace Technology,73(3),266-271.

搭乗順作成のアルゴリズム

```
#乗客の隠れIDで搭乗順をシャッフル
import random
tozyo=range(1,72)
random.shuffle(tozyo)
print tozyo

#リスト作成
attrmx=[]

for l in range(0,71):
    for mm in range(0,120):
        if tozyo[l]==attr[mm][4] and attr[mm][3]==1:
            print attr[mm]
            attrmx.append(tuple(attr[mm]))
        else :
            pass
```

1. それぞれの旅行ユニットにユニットID(U1,U2,,,Un)を振り分け
2. ユニットIDと同量の数字(l1,l2,,,ln)をシャッフル
ex) l = { l1 ,l2 ,l3,...l75} → l = { l5 ,l32 ,l25,...l65}
3. 新しくリストattrmx[]を作成
4. シャッフルした数字の先頭の数と全乗客の隠れIDを照合、かつ窓側の席(F=1)の乗客である場合は、その乗客を作成したリストに先頭から入れる
 - ・ex) l(1)=l5
 - ・ l5=U5 かつ F=1
 - ・元リストattr[]から ((x,y) , K , secretID , 1 , l5)を探し、attrmx[]へ
 - ・存在しない場合はpass

```

⊖ for mm in range(0,120):
⊖     if tozyo[1]==attr[mm][4] and attr[mm][3]==2:
⊖         print attr[mm]
⊖         attrmx.append(tuple(attr[mm]))
⊖     else :
⊖         pass
⊖ for mm in range(0,120):
⊖     if tozyo[1]==attr[mm][4] and attr[mm][3]==3:
⊖         print attr[mm]
⊖         attrmx.append(tuple(attr[mm]))
⊖     else :
⊖         pass

#attrmxからattrに移動
t=0
⊖ for nn in range(0,120):
⊖     attr[t]=attrmx[nn]
⊖     print attr[t]
⊖     t=t+1
print attr

attr=sorted(attr, key=lambda k:k[1])
print attr

```

5. 中央席(F=2)、通路側(F=3)も同様に行う

ex) 中央席の場合 I(1)=I5

I5=U5 かつ F=2

通路側の場合

I5=U5 かつ F=3

6. シャッフルした数字の二番目(I(2))、三番目(I(3))、、、とループ

7. 全ての乗客がリストattrmx[]に入ったら、元のリストattr[]に順番を変えずに置き換える

8. 搭乗グループKでソート(K=1から順に並び替える)