

# Dynamic ridesharing : Is there a role for dedicated drivers?

Alan Lee, Martin Savelsbergh

Transportation Research Part B, Vol. 81, pp. 483–497, 2015.

2016/5/18(水)

理論談話会#4

M1 三木真理子

1. Introduction: 問題設定
2. Ridesharing system: ライドシェアのフレーム
3. Literature review : ライドシェア/ルーティング問題
4. Formulation: 問題定式化
5. Heuristics: 解法の設計
6. Computational results: 数値計算

どちらかというと・・・

定式化や解法アルゴリズムの新規性というより  
フレームが新しいという印象

## ■背景

ライドシェアリングがなかなか根付かない

：“新たなサービス”に必然的に伴う不確実性が原因？

## ■既往研究から見るサービス成立要件

参加者が十分に多ければサービスは安定

→導入期に十分に多くの人を満足させ、  
利用者から信頼を得ることが必要.

### ➤ マッチング不成立の場合・・・

- ドライバー：ひとりで移動できる
- ライダー：代替手段を見つけなければ移動できない

→ライダーに対する影響が大きい

## ■ 仮説

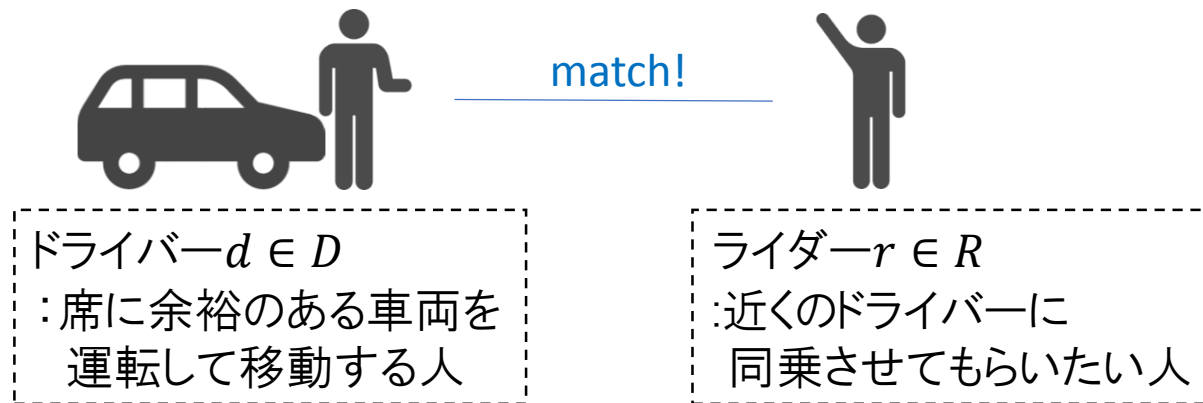
ドライバーを雇うことによってライダーの移動の確実性を高めることができるのではないか。

## ■ 本論文の目的

ライドシェアリングシステムに専属ドライバーを導入する際の利点・困難な点・コストの変化を調査する

参加者同士のマッチング＋専属ドライバーのルーティングを同時に考える問題

### ■従来型のライドシェアのフレーム



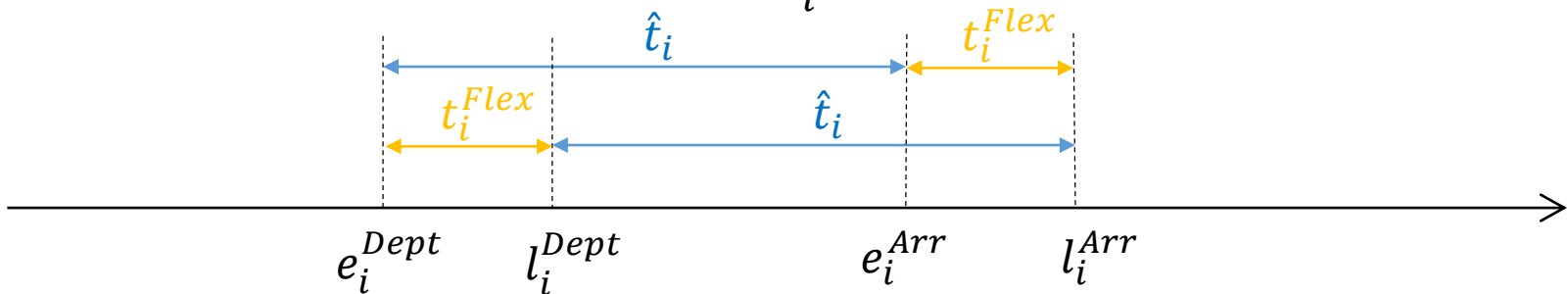
#### ● マッチングの成立条件

1. Time-feasible: 両者のtime window制約が満たされる
2. Cost-feasible: 両者にとって1人で移動するよりも低コスト

## ■ Time-feasible

参加者  $i \in D \cup R$  を次のように特徴づける

- 出発地と目的地
- 出発地から目的地までの移動時間  $\hat{t}_i$
- 目的地到着時刻の下限  $l_i^{Arr}$
- 出発地出発時刻の上限  $e_i^{Dept}$



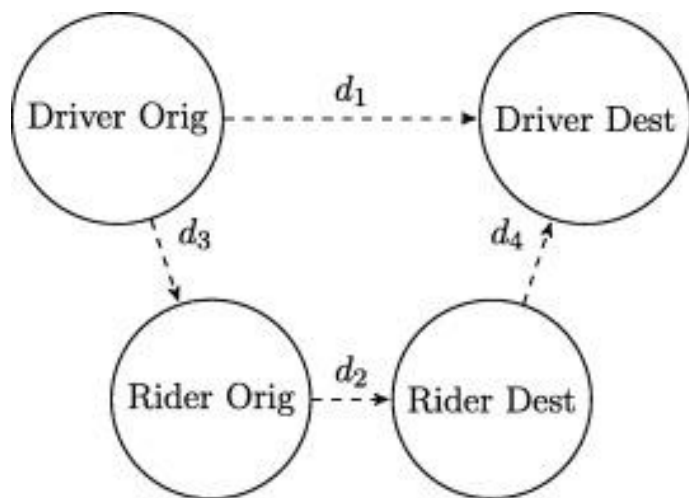
出発時刻および到着時刻のtime windowと柔軟性(time flexibility)は次のように計算できる

$$\begin{aligned}l_i^{Dept} &= l_i^{Arr} - \hat{t}_i \\e_i^{Arr} &= e_i^{Dept} + \hat{t}_i \\t_i^{Flex} &= l_i^{Arr} - e_i^{Arr} - \hat{t}_i\end{aligned}$$

### ■ Cost-Feasible

双方にとって、1人で移動するよりも低コストになるようなマッチングのみ成立する

[仮定]コストが移動距離に比例する



左図のマッチング成立条件は  
 $c(d_3) + c(d_2) + c(d_4) < c(d_1) + c(d_2)$

仮定より, これは以下のときに成立  
 $d_3 + d_2 + d_4 < d_1 + d_2$

■ ライダーの支払額  $f_r$ , ドライバーの負担額  $f_d$  を定める

$$f_r = \frac{c(d_2)}{c(d_1) + c(d_2)} \cdot (c(d_3) + c(d_2) + c(d_4)) < c(d_2)$$
$$f_d = \frac{c(d_1)}{c(d_1) + c(d_2)} \cdot (c(d_3) + c(d_2) + c(d_4)) < c(d_1)$$

### ■ 目的関数の設定

マッチングの望ましさを評価する観点が多様

- 成立したペア数が多い
- 時間調整が少ない
- 中心部に行く車の台数が少ない
- 総移動距離が少ない

→それぞれについて重み付けし、スコアを評価する

### ■ ライドシェアシステムを定式化

$$\begin{aligned} & \max \sum_{i \in D} \sum_{j \in R} \widehat{c}_{ij} x_{ij} \\ & \text{s. t. } \sum_{i \in D} x_{ij} \leq 1, \quad \forall j \in R \\ & \quad \sum_{j \in R} x_{ij} \leq 1, \quad \forall i \in D \end{aligned}$$

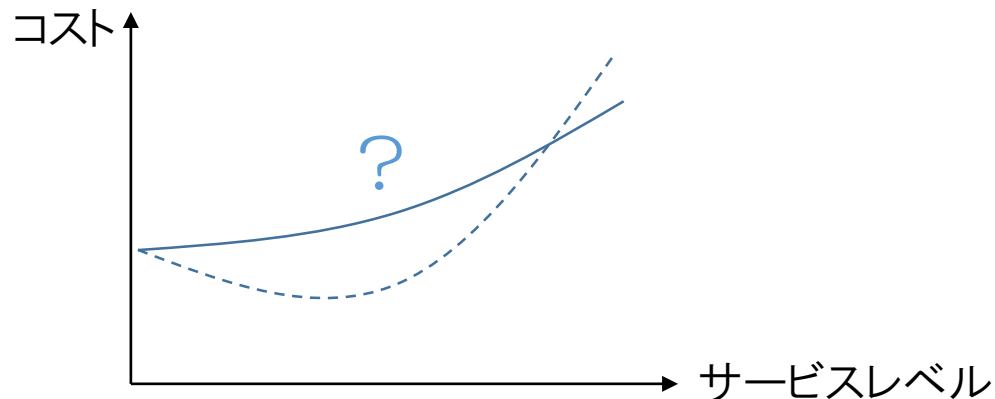
$\widehat{c}_{ij}$ : リンク  $ij$  のスコア  
 $x_{ij}$ : リンク  $ij$  のマッチング有無



## ■本研究の目的

専属ドライバーを雇うことによるコストと、達成されるサービスレベルとの関係を見る

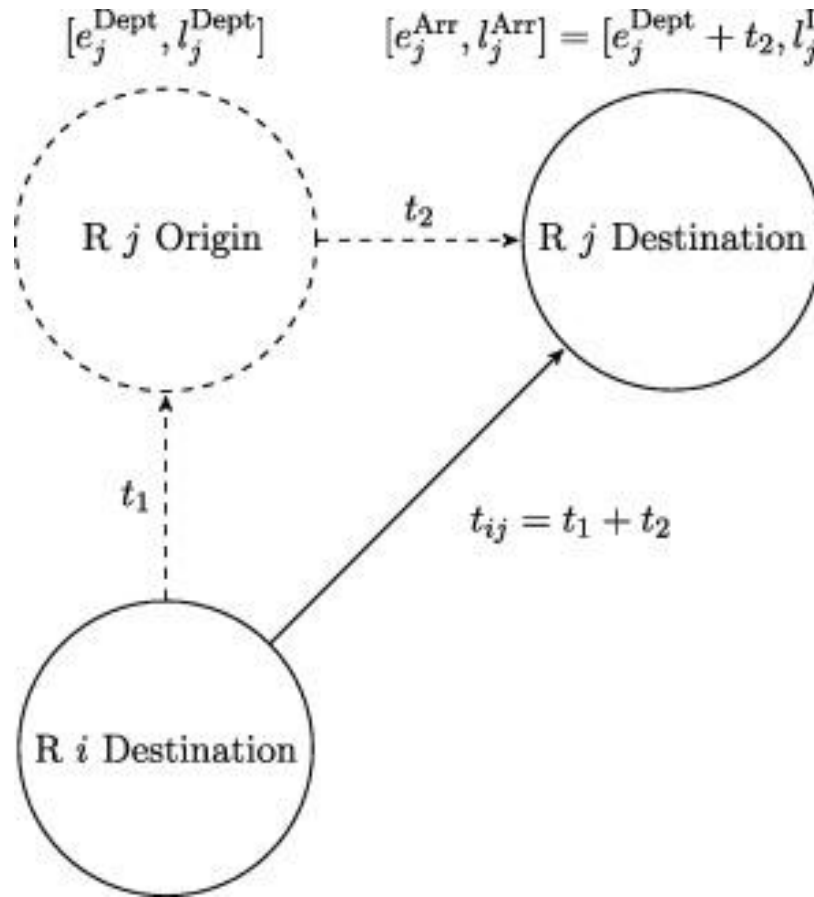
※サービスレベル＝ライダーのマッチング成功率



## ■問題

1. ライダーの支払額の設定
2. 専属ドライバーを雇うのに必要なコストの設定

## ■ 専属ドライバーのルーティングの表現方法



グラフ  $G = (N, A)$  により表現

$N$ : ライダー  $i \in R$  を表すノード

$A$ : ライダー  $i, j \in R$  を結ぶリンク

専属ドライバーの移動

(  $Destination^i \rightarrow Origin^j$

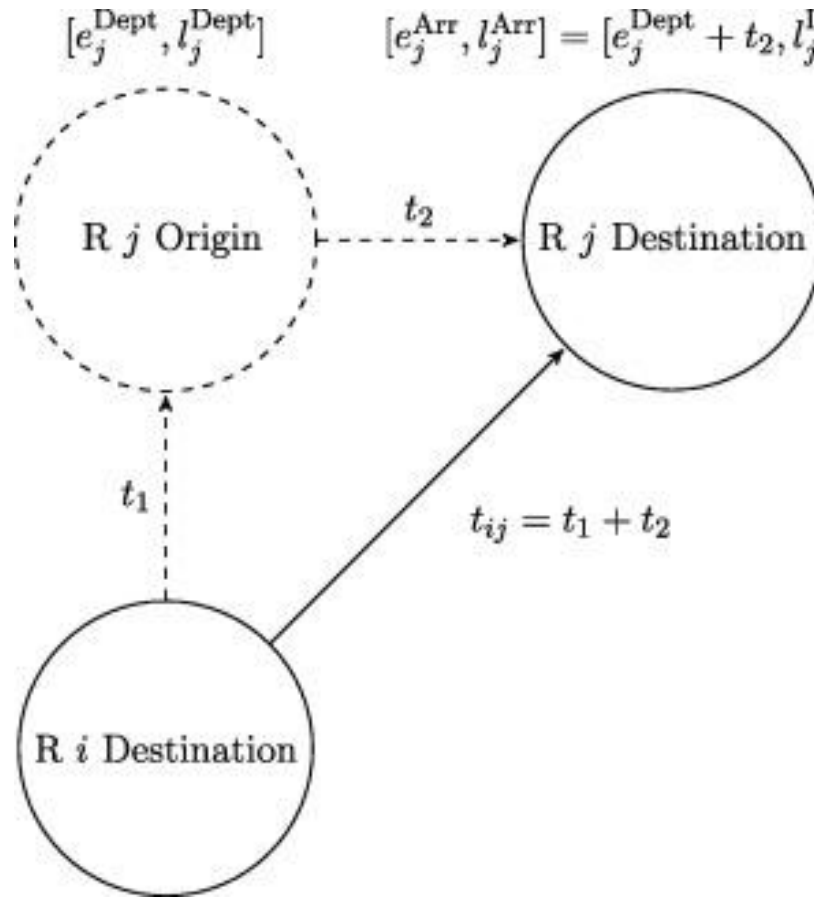
$\rightarrow Destination^j$  ) を表す

$t_{ij}$  : ノード  $i$  からノード  $j$  までの所要時間

$T_j$  : ノード  $j$  の到着時間

$$e_j^{\text{Arr}} \leq T_j \leq l_j^{\text{Arr}}$$

## ■ 専属ドライバーのルーティングの表現方法



専属ドライバーノード  $k \in K$  を  $N$  に追加

[各ノード  $k \in K$  のもつ情報]

$e_k$  : 出発可能時刻の上限

$l_k$  : 帰宅時刻の下限

$T_k^e$  : 実際の出発時刻

$T_k^l$  : 実際の到着時刻

制約条件は次のようにかける

$$e_k \leq T_k^e, \quad T_k^l \leq l_k$$

運行時間制約は、上限を  $L$  とすると、

$$T_k^l - T_k^e \leq L, \quad \forall k \in K$$

$x_{ij}$ :  $i \in D$ と $j \in R$ が同乗

$y_{ij}^k$ : 専属ドライバー $k$ が $i \in R$ の次に $j \in R$ を送る

$y_{0j}^k$ : 専属ドライバー $k$ が最初に $j \in R$ を送る

$y_{j0}^k$ : 専属ドライバー $k$ が最後に $j \in R$ を送る

$y_{00}^k$ : 専属ドライバー $k$ は使わない

[係数]

$c_{ij}$ : リンク $ij$ にかかるコスト

$\hat{s}$ : サービス保証レベル

(ライダーの移動達成率)

$$\min \sum_{k \in K} \left( \sum_{j \in R} c_{0j} y_{0j}^k + \sum_{i \in R} \sum_{j \in R} c_{ij} y_{ij}^k \right)$$

$$\sum_{j \in R} x_{ij} \leq 1, \quad i \in D$$

(1) ドライバーのマッチング相手は1人以下

$$\sum_{i \in D} x_{ij} + \sum_{k \in K} \left( y_{0j}^k + \sum_{i \in R} y_{ij}^k \right) \leq 1, \quad j \in R$$

(2) ライダーのマッチング相手は1人以下

$$\sum_{j \in R} \left( \sum_{i \in D} x_{ij} + \sum_{k \in K} \left( y_{0j}^k + \sum_{i \in R} y_{ij}^k \right) \right) \geq \lceil \hat{s} * |R| \rceil \quad (3) \text{ マッチング成功率は } s \text{ 以上}$$

$$\sum_{i \in R \cup \{0\}} y_{ij}^k - \sum_{i \in R \cup \{0\}} y_{ji}^k = 0, \quad j \in R, k \in K$$

(4) 専属ドライバーはノードを訪れたら必ずそのノードから出て行く

$$\sum_{j \in R} y_{0j}^k + y_{00}^k = 1, \quad k \in K$$

(5) 専属ドライバーは誰かを送るか移動しないかのいずれかである

$$T_k^e \geq e_k, \quad k \in K \quad (6) \text{ 専属ドライバーの出発時刻制約}$$

$$T_j \geq e_j^{\text{Arr}}, \quad j \in R \quad (7) \text{ ライダーの到着時刻制約}$$

$$T_j \geq \sum_{k \in K} (T_k^e + t_{0j}^k) y_{0j}^k, \quad j \in R \quad (8) \text{ トリップの接続性の制約}$$

$$T_j \geq (T_i + t_{ij}) \sum_{k \in K} y_{ij}^k, \quad j \in R, i \in R \quad (9) \text{ トリップの接続性の制約}$$

$$T_j \leq l_j^{\text{Arr}}, \quad j \in R \quad (10) \text{ ライダーの到着時刻制約}$$

$$T_k^l - T_k^e \leq L, \quad k \in K \quad (11) \text{ 専属ドライバーの労働時間制約}$$

$$T_k^l \geq \sum_{j \in R} (T_j + t_{j0}^k) y_{j0}^k, \quad k \in K \quad (12) \text{ トリップの接続性の制約}$$

$$T_k^l \leq l_k, \quad k \in K \quad (13) \text{ 専属ドライバーの到着時刻制約}$$

$$x_{ij} \in \{0, 1\}, \quad i \in D, j \in R$$

$$y_{ij}^k \in \{0, 1\}, \quad i \in R \cup \{0\}, j \in R \cup \{0\}$$

- 制約条件が決定変数に関する二次式となるため、非線形整数計画問題となる
- 問題の規模が大きくなると厳密解を求めるのが困難なため、(メタ)ヒューリスティクスを適用
  
- 発見的解法(Heuristics)
  - 解の精度の保証はないが、経験的に近似解が求められるとわかっている手法
    - 構築法: 何もない状態からあるルールに基づいて解を構築
    - 改善法: 実行可能初期解からあるルールに基づき解を改善
  
- メタヒューリスティクス
  - ヒューリスティクスにパラメータを追加して計算を効率化するテクニック

## ■ 解の表現方法

matching と schedule で表現

- ・ matching : 通常ドライバーとライダーのペアの集合
- ・ schedule : ある専属ドライバーが送るライダーの順序つきリスト



解は、ライダーが重複しないようなmatchingと専属ドライバーの人数分のschedulesの組合せで表される

## ■ 解法の流れ

1. 複数の初期解の構築
2. 局所探索法による初期解の改善
3. Shakingによる解の改変(確率的な改善)

## 1. 複数のmatchingを生成

ライドシェアシステムの最適化問題(2章で紹介)を解くことによって得る

## 2. Sequential insertion heuristicにより, 残りのライダーを輸送するschedulesを作る

### ■ Sequential insertion heuristic

- 各ライダーについて, 当該ドライバーに対する最も低コストな挿入を決め, その中で最も低コストなライダーを実際に挿入する
- 当該ドライバーについて挿入できるライダーがいなくなったら, サービスレベルに達するまで新たなドライバーについてscheduleを作る.



1. 2章の整数計画問題を  $\widehat{c}_{ij} = 1 \quad \forall i \in D, \forall j \in R$  として解く

matching-IP

$$\begin{aligned} & \max \sum_{i \in D} \sum_{j \in R} \widehat{c}_{ij} x_{ij} \\ & \text{s. t. } \sum_{i \in D} x_{ij} \leq 1, \quad \forall j \in R \\ & \quad \sum_{j \in R} x_{ij} \leq 1, \quad \forall i \in D \end{aligned}$$

この解を  $x^*$ , ペア数を  $z^*$  と表す.  
以降, マッチングのペア数を  $z^*$  と制限する

2.  $\widehat{c}_{ij} = 1 - x_{ij}^* + \delta_{ij}$  とする.  
 $\delta_{ij}$  はランダム項.  $[-0.5, 0.5]$  の区間で一様に分布
3. コスト  $\widehat{c}_{ij}$  が更新された matching-IP を解く.  
解  $\hat{x}$  を matching とする.

十分な数の matching が得られるまで, 2番以降を繰り返す.

近傍解として次の2種類を使う

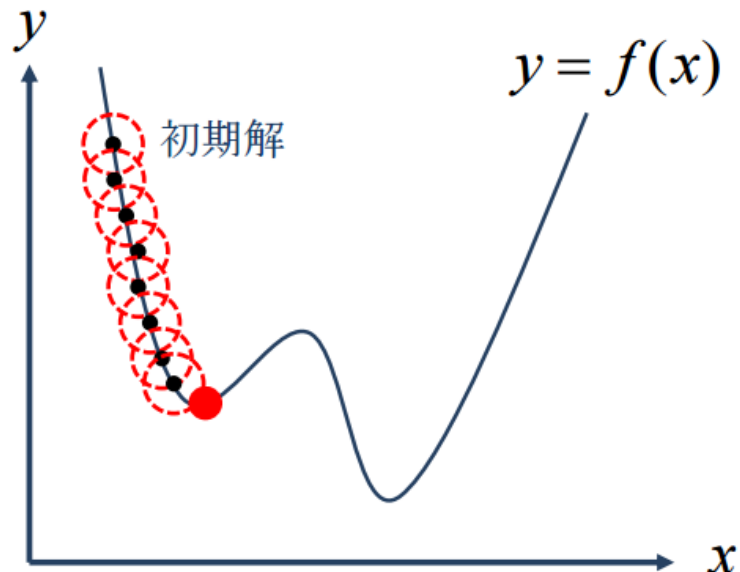
■ schedule を変更した解

■ あるライダーを送るドライバーを変更した解

## ● 局所探索法 (Local Search)

現在の解の近傍が目的関数の値を改善する場合はその近傍を新たな解入れ替え、そうでなければ現在の解を維持する

※別名: 丘登り法 (Hill Climbing Method)



- 現在の解の近傍の範囲
- 現在の解
- 局所的最適解

メタヒューリスティクスゼミ(2013)#1  
今泉さんのスライドより引用

専属ドライバーによって輸送されているライダーについてより低いコストで輸送できるようなscheduleの組合せを探す

変更の仕方は次の5通り

1. 各ライダーについて、ドライバーを変えずに順番を変更する
2. あるドライバーのschedule内にいるすべてのペアについて、ペアを入れ替える
3. あるライダーについて、別のスケジュール内のどこかに移動させる
4. あるライダーについて、別のスケジュール内のライダーと入れ替える
5. 同一schedule内の一部の順列について、schedule内の全員が輸送される限りで、逆順にする

- 通常ドライバー  $d \in D$  が、matchingによりライダー  $r \in R$  と同乗することになっている場合
  - 専属ドライバーに送ってもらうライダーの集合を  $R_d$  とする
  - 各ライダー  $\hat{r} \in R_d$  について、次の2点を検証
    - ✓ 専属ドライバーがライダー  $\hat{r} \in R_d$  の代わりにライダー  $r \in R$  を送れるか
    - ✓ 送るライダーを入れ替えた場合に専属ドライバーのコストが下がるか

↓

2つとも満たす場合、ライダー  $\hat{r} \in R_d$  と  $r \in R$  を入れ替える
- 通常ドライバー  $d \in D$  が matching に含まれていない場合
  - 別の通常ドライバーと同乗することになっているが、ドライバー  $d \in D$  と同乗可能なライダーの集合  $S_d$  を作る
  - $S_d \neq \phi$  なら、 $S_d$  中のライダーをランダムに選び、ドライバーを入れ替える

schedulesから確率的にライダーを消去し, 再び挿入する

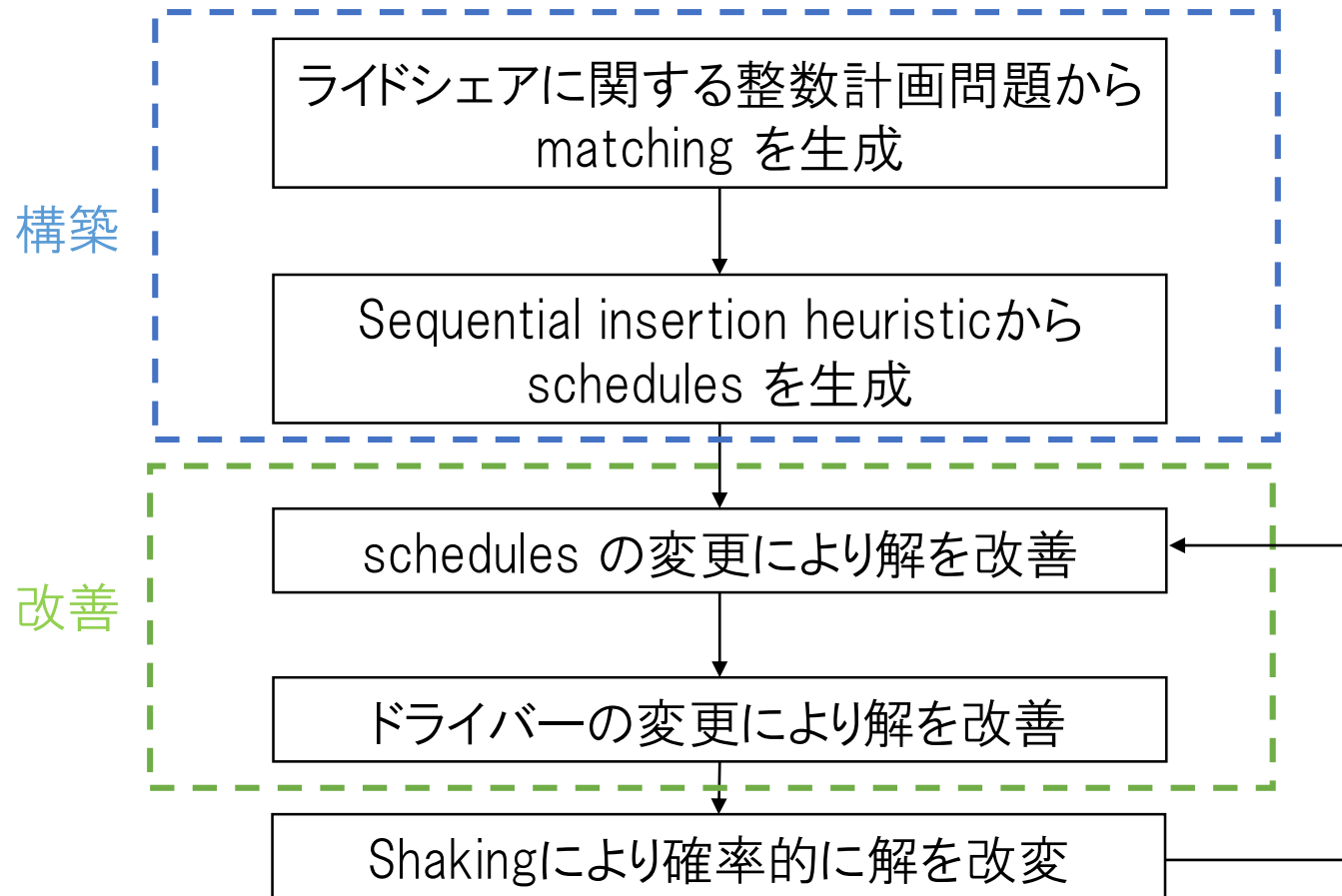
次の4つのパラメータを使う(いずれも0以上1以下)

[消去]

- deleteFraction  
: schedule中で消去するライダーの割合
- probLargest  
: コスト低下をもたらすようなライダーの選択確率  
(確率的にコスト増も許容する)

[再挿入]

- probIncludeDeleted  
: 消去したライダーが再び戻される確率
- probLeast  
: コストが最低となるような挿入を行う確率  
(確率的に最低とならない解も許容)



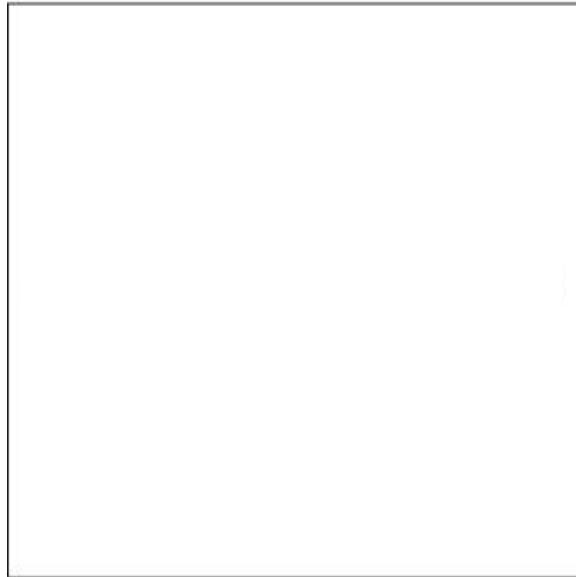
## ■収束判定

Shakingを2度繰り返しても、最良解が改善されなければ終了

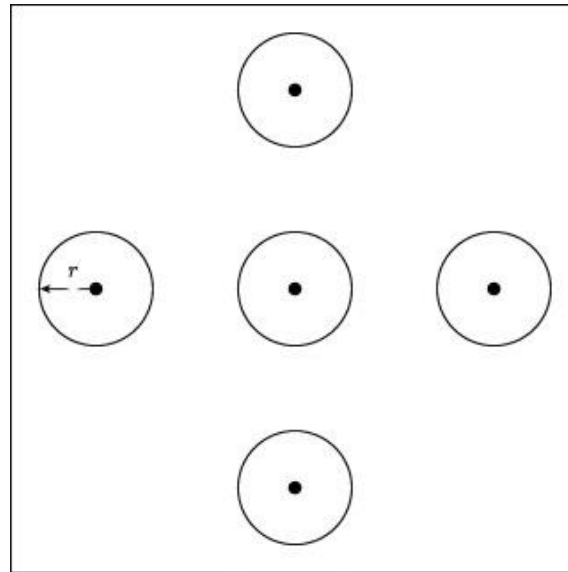
### ■ 需要側のシナリオを次のパラメータの組合せで生成

- numParticipants  
: サービス参加者の人数(ドライバー/ライダーは半々)  
{100, 200, 300, 400, 500}の5通り
- spreadParticipants  
: 参加者の0またはDの分布  
{uniform, 5-hubs, 2-hubs}の3通り
- meanFlexibility  
:  $t_i^{Flex}$ の従う正規分布の平均値  
{24分, 36分}の2通り

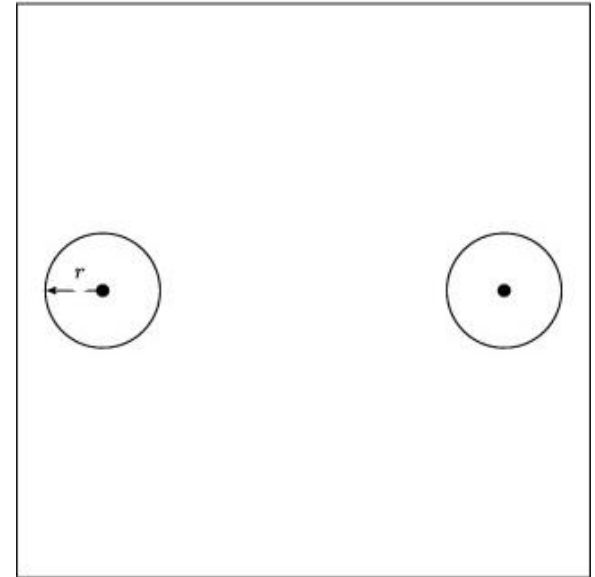
## ■spreadParticipants : トリップパターン(0とDの位置)に関する設定



Uniform dist  
サービスエリア内に  
一様に分布



5-hubs  
サービスエリア内の  
5つのハブの周りに集中



2-hubs  
サービスエリア内の  
2つのハブの周りに集中

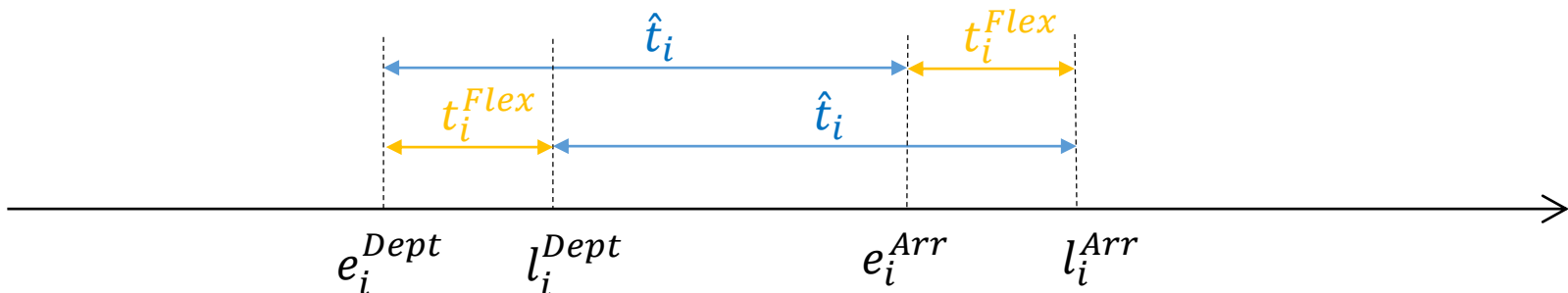
※ハブ周りに集中: 半径 $r$ の円内に一様に分布



■ meanFlexibility : time-window制約の柔軟さを表す

➤ 参加者*i*のTime-window制約の表現

- $t_i^{Flex}$  : 平均{24分, 36分}・標準偏差4分の切断正規分布からサンプリング
- $l_i^{Arr}$  : 平均240分・標準偏差45分の切断正規分布からサンプリング
- $\hat{t}_i$  : ODをもとに計算でもとめる
- $l_i^{Dept} = l_i^{Arr} - \hat{t}_i$
- $e_i^{Arr} = l_i^{Arr} - t_i^{Flex}$
- $e_i^{Dept} = l_i^{Dept} - t_i^{Flex}$



### ■ サービス提供者の意思決定原理

[設定] 専属ドライバーを雇うコストは,  
移動距離に比例する(1\$/mile)

[目的] サービスレベルを維持しながら  
オペレーションコストを最小化する

### ■ 計算

- 需要側の各シナリオについて10例を生成
- サービスレベル(達成率)が,  
{90%, 95%, 98%}のそれぞれの場合について計算

## ■参加者50人，達成率98%で厳密解を求めて比較

	Uniform distribution	Five hubs	Two hubs
Average difference in objective value	0.84%	0.72%	0.04%
Maximum difference in objective value	2.04%	3.15%	0.28%
Average difference in vehicles used	0.3	0.3	0.1
Maximum difference in vehicles used	1	1	1
Instances where Opt was achieved	2/10	5/10	8/10

■最適解との差は平均して1%未満．最大でも3.15%

■30例のうち半数で最適解に一致し，5例では使用車両が1台増えた．

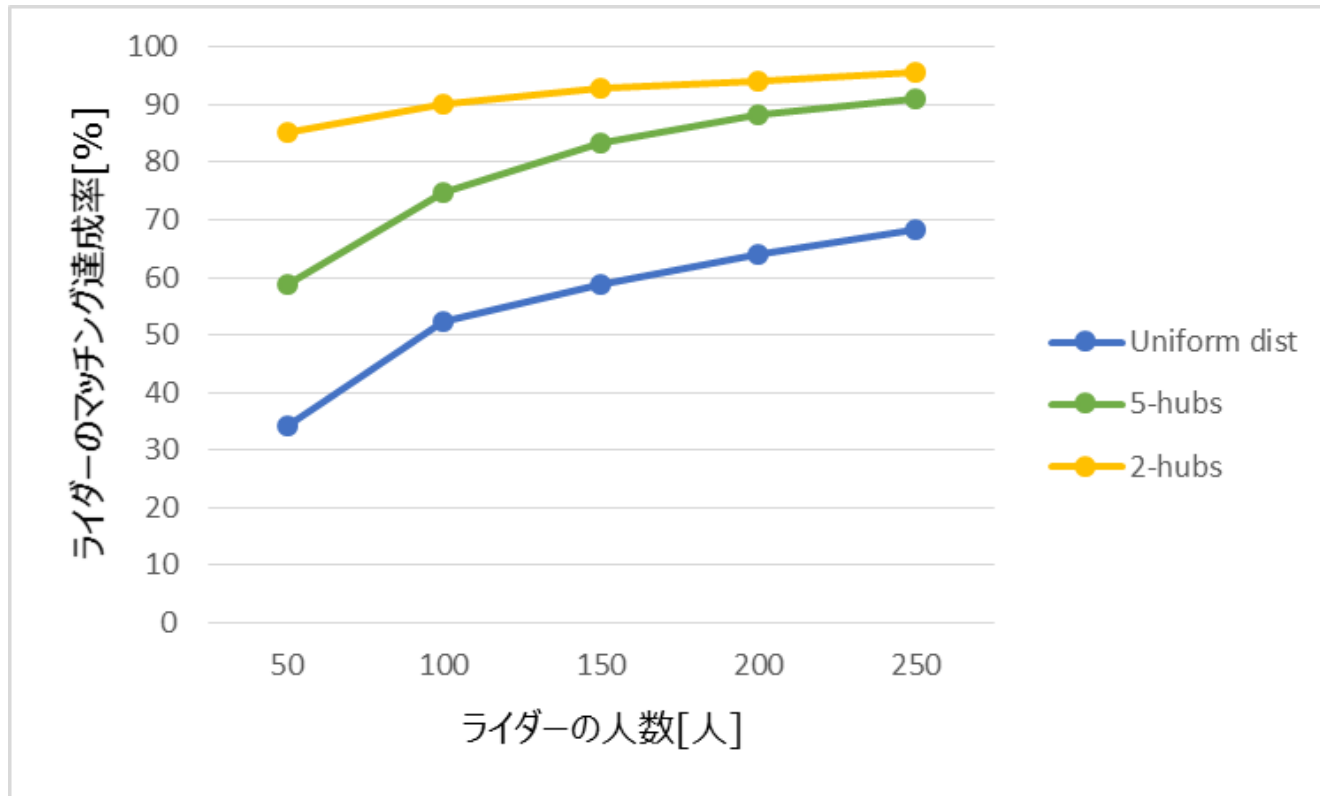


十分信頼できるといえるだろう

## ■参加者同士のマッチング可能率

Fraction of riders that can be served by ad hoc drivers.

Num participants	Num riders	Uniform dist.	Five hubs	Two hubs
100	50	34.20% (17.10)	58.80% (29.40)	85.20% (42.60)
200	100	52.20% (26.10)	74.70% (37.35)	90.10% (45.05)
300	150	58.87% (29.43)	83.40% (41.70)	92.80% (46.40)
400	200	64.10% (32.05)	88.20% (44.10)	94.05% (47.03)
500	250	68.28% (34.14)	90.92% (45.46)	95.52% (47.76)



## 専属ドライバーの効果

Performance statistics of dedicated drivers.

	Total cost (per-rider cost)			# ded drivers			% served by ded drivers		
	90%	95%	98%	90%	95%	98%	90%	95%	98%
<i>Uniform distribution</i>									
100	527.11 (18.90)	622.21 (20.15)	658.27 (20.65)	5.90	6.90	7.10	62.00%	64.38%	65.10%
200	631.92 (16.73)	769.00 (17.97)	862.89 (18.85)	7.30	9.10	10.10	42.00%	45.05%	46.73%
300	728.62 (15.56)	942.18 (17.20)	1059.62 (18.02)	8.60	10.90	12.20	34.59%	38.25%	39.93%
400	759.99 (14.62)	991.33 (16.00)	1158.35 (17.04)	8.60	11.50	13.70	28.78%	32.53%	34.59%
500	754.09 (13.84)	1037.52 (15.37)	1246.87 (16.74)	8.90	12.10	14.00	24.13%	28.28%	30.33%
<i>Five hubs</i>									
100	224.59 (14.32)	289.70 (15.55)	312.83 (15.94)	3.10	4.00	4.30	34.67%	38.75%	40.00%
200	193.16 (12.56)	279.21 (13.74)	336.72 (14.45)	3.00	4.00	5.00	17.00%	21.37%	23.78%
300	120.15 (11.74)	244.67 (13.53)	323.63 (14.68)	2.00	4.00	5.00	7.33%	12.52%	14.90%
400	58.51 (12.13)	169.73 (12.40)	276.56 (14.14)	1.29	2.80	4.80	2.11%	7.16%	10.00%
500	47.08 (10.56)	121.46 (11.40)	229.76 (13.08)	1.50	2.10	3.90	0.40%	4.50%	7.22%
<i>Two hubs</i>									
100	49.62 (22.03)	110.31 (20.23)	137.97 (21.47)	1.10	1.90	2.60	5.33%	11.25%	13.06%
200	140.35 (23.39)	130.55 (23.57)	184.51 (22.96)	1.67	1.89	3.00	2.00%	5.26%	8.06%
300	122.07 (26.76)	179.29 (24.06)	191.47 (23.38)	1.50	2.33	2.80	0.67%	2.94%	5.31%
400	107.59 (26.90)	136.17 (24.12)	195.89 (24.47)	1.00	1.83	2.60	0.22%	1.74%	4.03%
500	25.52 (25.52)	237.39 (23.00)	176.43 (24.03)	1.00	3.00	2.22	0.04%	0.80%	2.65%

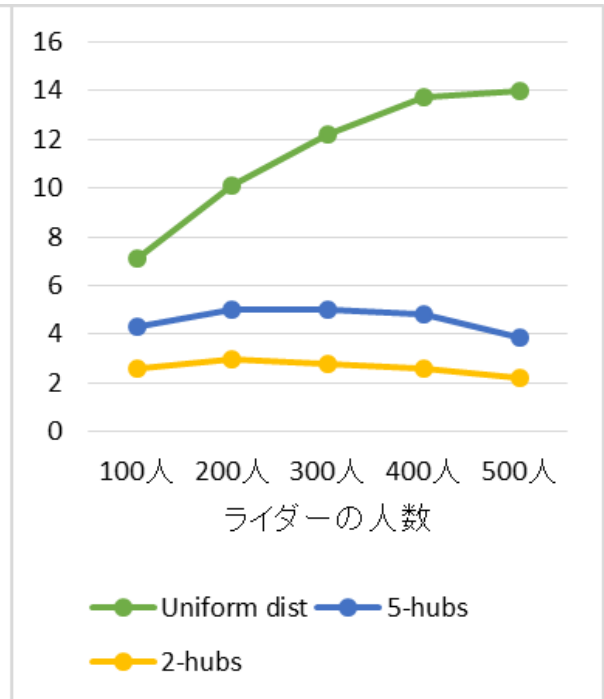
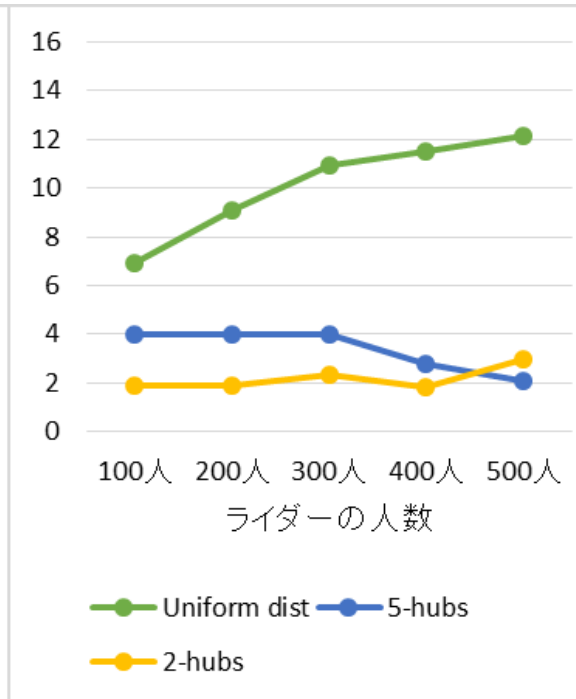
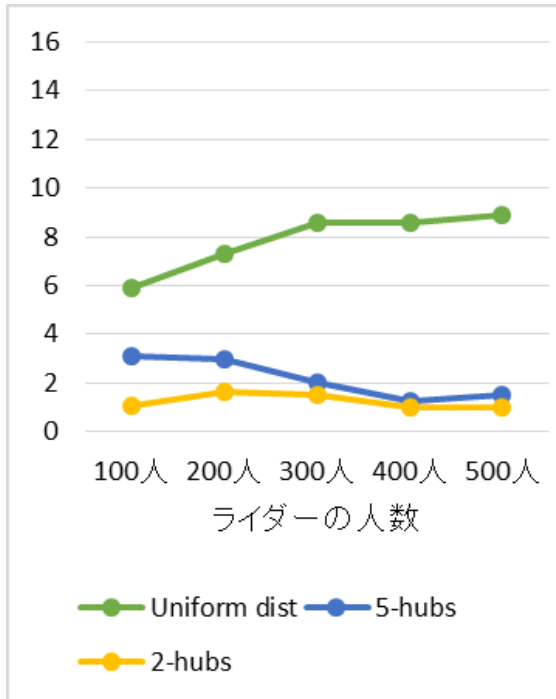
Number of instances for which no dedicated drivers were needed.

	Uniform distribution			Five hubs			Two hubs		
	90%	95%	98%	90%	95%	98%	90%	95%	98%
100	-	-	-	-	-	-	-	-	-
200	-	-	-	-	-	-	7	1	-
300	-	-	-	-	-	-	8	4	-
400	-	-	-	3	-	-	9	4	-
500	-	-	-	8	-	-	9	8	1

達成率90%

達成率95%

達成率98%



5-hubs, 2-hubs

: 専属ドライバーの人数にピークが見られる  
→ 参加者が増えればサービスが安定する

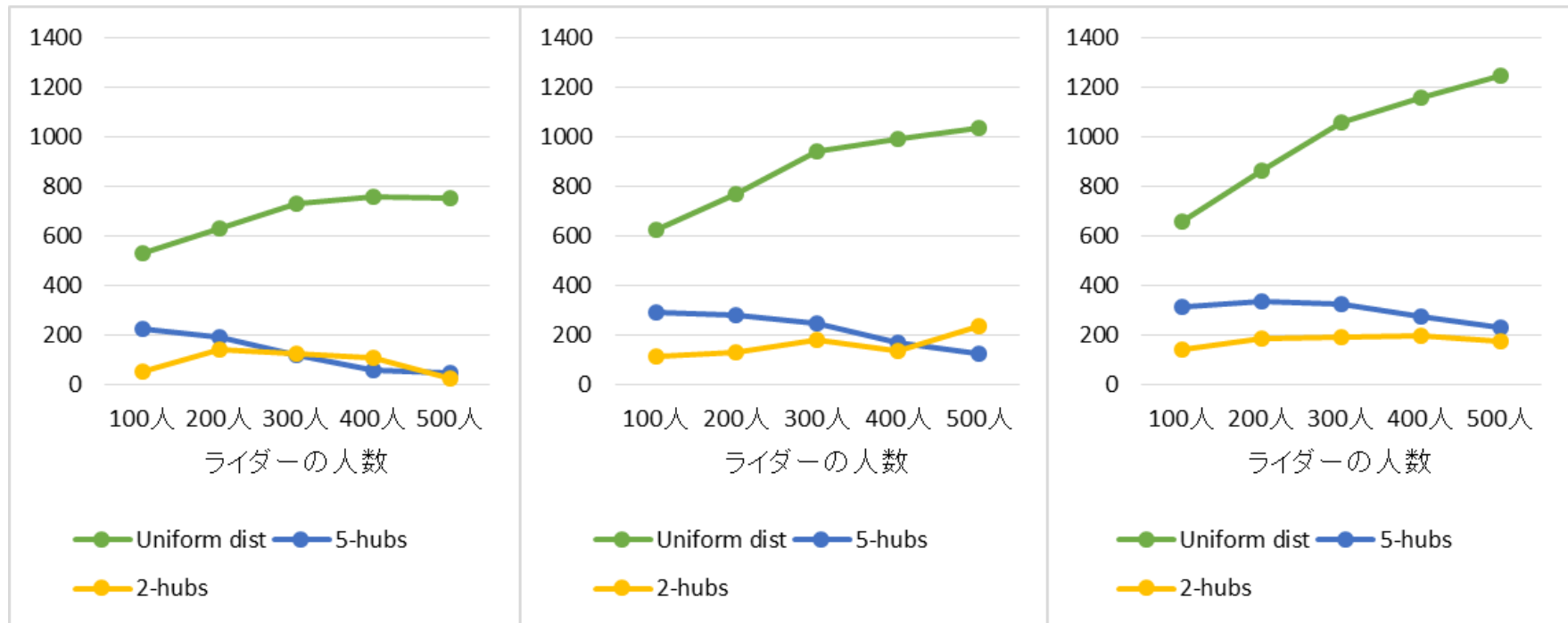
Uniform dist.

: 参加者が増えるにつれて必要なドライバー数も増える

達成率90%

達成率95%

達成率98%



5-hubs, 2-hubs

:参加者が増えるほどコストは下がる

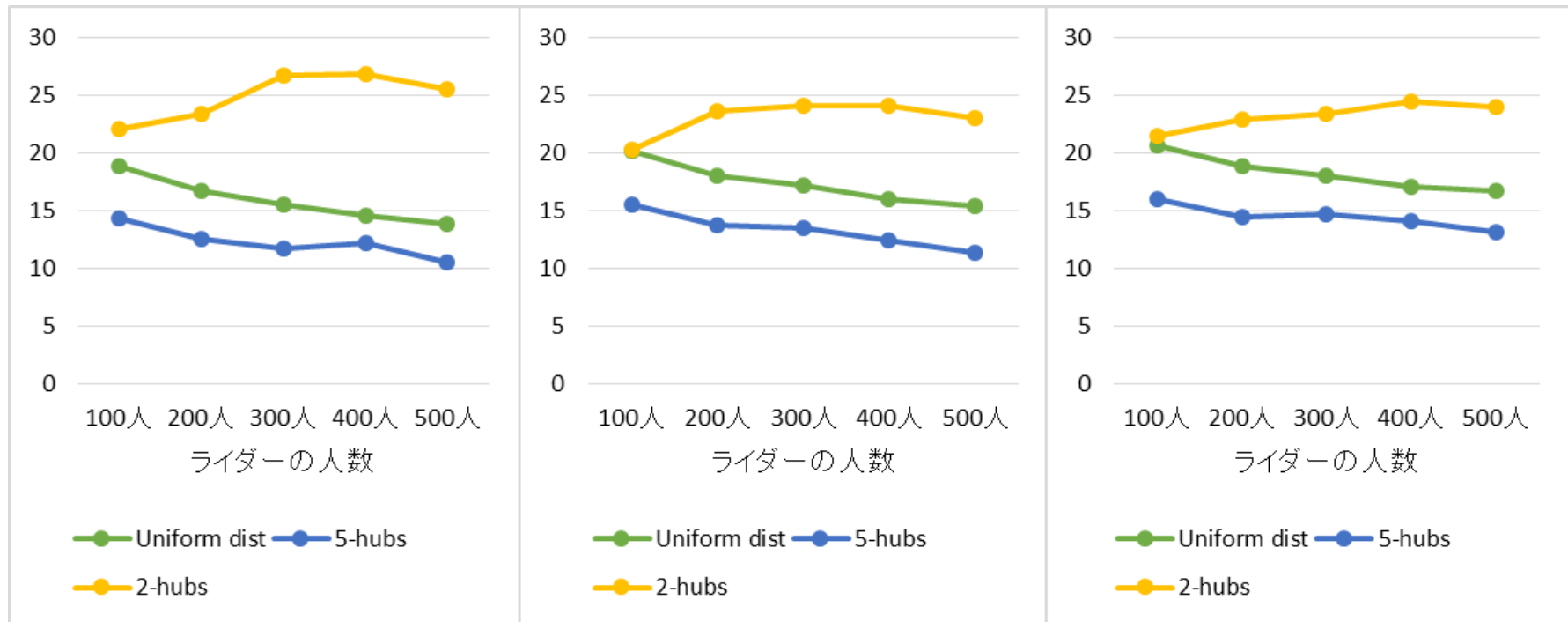
Uniform dist.

:参加者が増えるにつれて必要なドライバー数も増える

達成率90%

達成率95%

達成率98%



ライダーの人数が少ないほど、専属ドライバーの  
回送距離が長くなりやすい



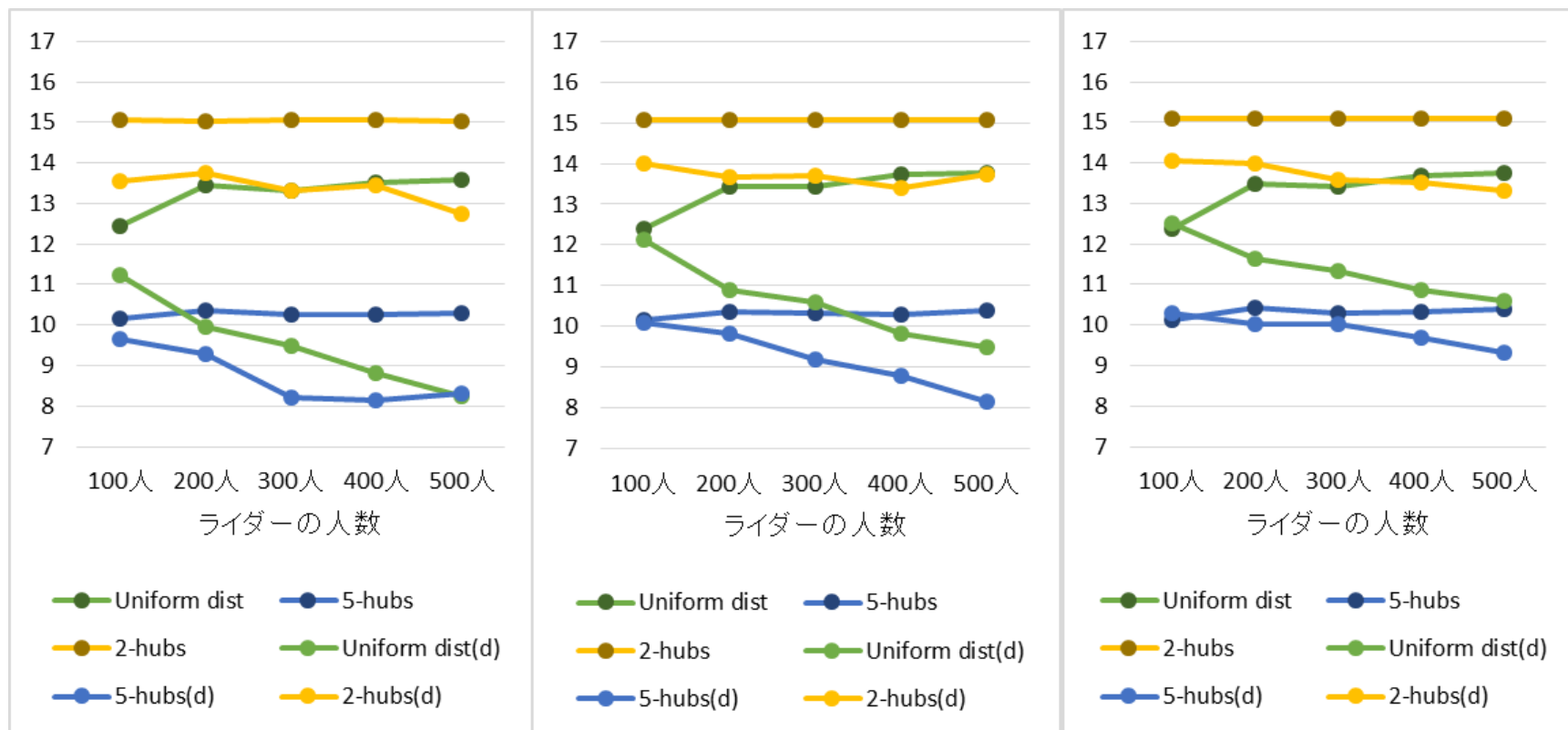
参加者が多いほどコストは少なくなる



達成率90%

達成率95%

達成率98%



移動距離の短いライダーを専属ドライバーが送っている

専属ドライバーの送るライダーの平均距離は、サービスレベルが上がるにつれて長くなる

※緩和しても専属ドライバーを必要とするケースのみで計算

Decrease in total cost from lowering the required service level.

	Uniform distribution		Five hubs		Two hubs	
	98%-95%	95%-90%	98%-95%	95%-90%	98%-95%	95%-90%
100	5.51% (10)	15.38% (10)	7.53% (10)	22.98% (10)	21.29% (10)	57.29% (10)
200	11.02% (10)	18.00% (10)	17.25% (10)	31.39% (10)	42.24% (9)	57.33% (3)
300	11.10% (10)	22.34% (10)	25.33% (10)	53.38% (10)	41.77% (6)	69.96% (2)
400	14.36% (10)	23.50% (10)	39.84% (10)	71.44% (7)	59.38% (6)	71.93% (1)
500	15.93% (10)	27.57% (10)	48.56% (10)	76.75% (2)	46.97% (2)	93.30% (1)

■参加者が多いほど削減率は大きい傾向

※緩和前のコストは参加者が多いほど小さい  
→削減幅は小さくても削減率は大きくなる

※緩和しても専属ドライバーを必要とするケースのみで計算

## ■ライダーひとりあたりのコスト削減率

Effect of lowering the required service level on the average per-rider cost.

	Uniform distribution		Five hubs		Two hubs	
	95%-90%	98%-95%	95%-90%	98%-95%	95%-90%	98%-95%
100	2.42% (10)	6.17% (10)	2.48% (10)	7.95% (10)	5.96% (10)	-12.06% (10)
200	4.68% (10)	6.93% (10)	4.84% (10)	8.53% (10)	-8.44% (9)	5.53% (3)
300	4.55% (10)	8.91% (10)	8.09% (10)	13.44% (10)	-0.60% (6)	1.70% (2)
400	6.02% (10)	8.69% (10)	12.20% (10)	3.87% (7)	1.33% (6)	1.75% (1)
500	7.14% (10)	10.12% (10)	12.76% (10)	6.15% (2)	2.44% (2)	6.20% (1)

## ■専属ドライバーの送るライダーの平均移動距離の増減

Effect of lowering the required service level on the average origin-destination distance of riders served.

	Uniform distribution		Five hubs		Two hubs	
	95%-90%	98%-95%	95%-90%	98%-95%	95%-90%	98%-95%
100	2.88% (10)	7.63% (10)	1.83% (10)	4.38% (10)	0.33% (10)	3.42% (10)
200	6.19% (10)	8.61% (10)	2.04% (10)	5.57% (10)	1.78% (9)	0.59% (3)
300	6.49% (10)	9.61% (10)	8.61% (10)	10.64% (10)	-0.03% (6)	2.16% (2)
400	8.92% (10)	10.61% (10)	9.43% (10)	10.23% (7)	1.88% (6)	1.92% (1)
500	8.73% (10)	13.84% (10)	11.85% (10)	-0.57% (2)	0.25% (2)	6.22% (1)

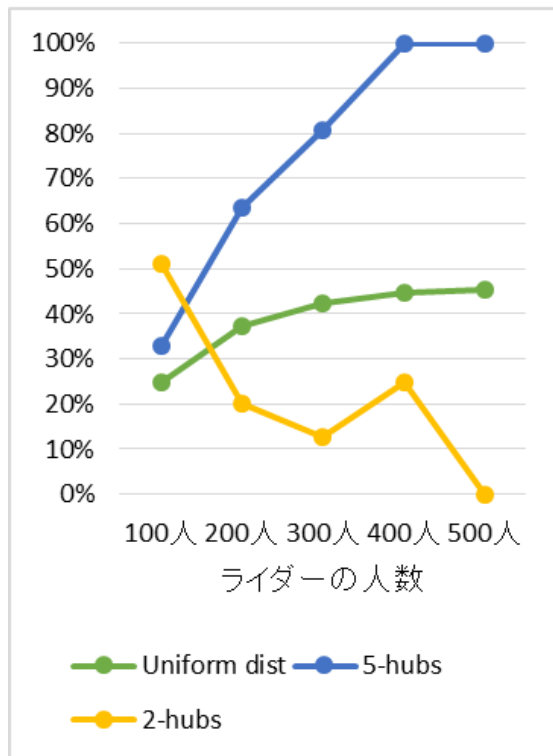
## ■ Time flexibilityの平均値を24分から36分に変化させて計算

※参加者のtime window制約が緩くなった,  
参加者に譲歩が生まれたという解釈ができる

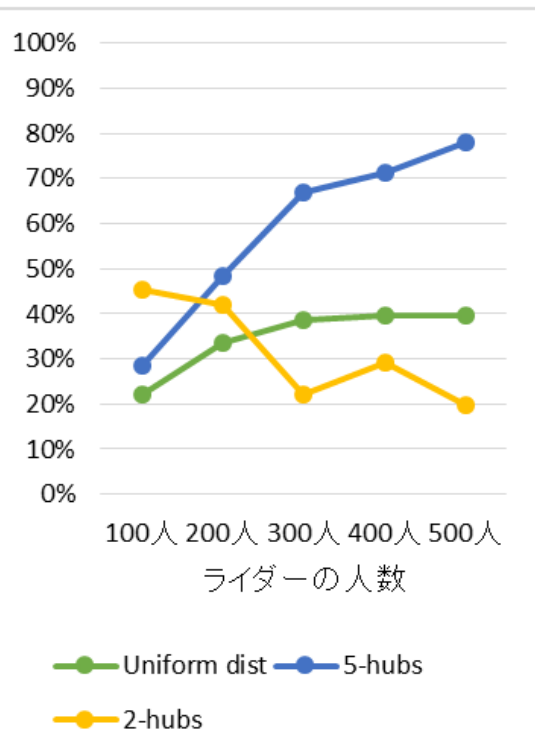
Percentage reduction resulting from greater flexibility.

	Total cost			Num of ded drivers			% served by ded drivers		
	90%	95%	98%	90%	95%	98%	90%	95%	98%
<i>Uniform distribution</i>									
100	24.71% (10)	22.11% (10)	21.74% (10)	30.90%	24.94%	18.21%	18.31%	16.44%	15.90%
200	37.13% (10)	33.55% (10)	31.60% (10)	35.63%	35.16%	34.34%	27.44%	24.10%	22.47%
300	42.33% (10)	38.76% (10)	36.35% (10)	39.56%	38.93%	34.32%	31.98%	27.22%	25.34%
400	44.67% (10)	39.71% (10)	37.31% (10)	42.30%	37.45%	37.23%	32.87%	27.53%	25.08%
500	45.28% (10)	39.74% (10)	37.64% (10)	40.42%	39.64%	31.83%	33.90%	27.32%	24.74%
<i>Five hubs</i>									
100	32.94% (10)	28.54% (10)	26.86% (10)	32.50%	26.33%	23.33%	31.23%	26.05%	24.70%
200	63.73% (10)	48.34% (10)	41.13% (10)	53.33%	42.00%	36.17%	59.91%	44.80%	38.95%
300	80.93% (4)	66.95% (10)	56.14% (10)	58.33%	57.67%	50.33%	79.43%	61.08%	49.42%
400	N/A	71.24% (7)	61.46% (10)	N/A	57.14%	61.50%	N/A	68.15%	56.74%
500	N/A	77.99% (4)	68.51% (9)	N/A	52.08%	58.52%	N/A	74.89%	63.85%
<i>Two hubs</i>									
100	51.09% (2)	45.47% (9)	42.22% (10)	25.00%	31.48%	46.67%	50.00%	52.90%	47.90%
200	20.09% (1)	42.04% (5)	37.29% (8)	33.33%	17.33%	48.96%	25.00%	44.20%	44.12%
300	12.68% (1)	22.23% (4)	24.56% (7)	0.00%	20.83%	38.10%	12.50%	20.59%	27.96%
400	24.98% (1)	29.07% (4)	25.87% (7)	0.00%	16.67%	20.24%	25.00%	25.83%	27.17%
500	0.00% (1)	19.70% (2)	19.58% (8)	0.00%	25.00%	15.62%	0.00%	20.00%	23.81%

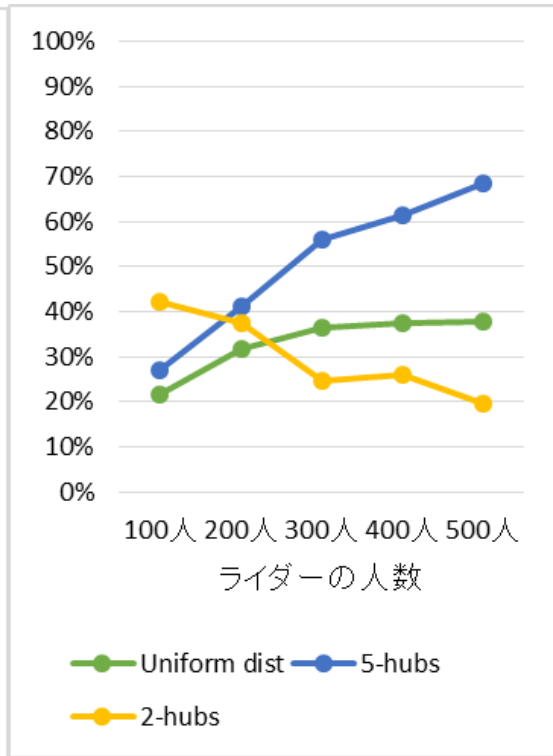
達成率90%



達成率95%

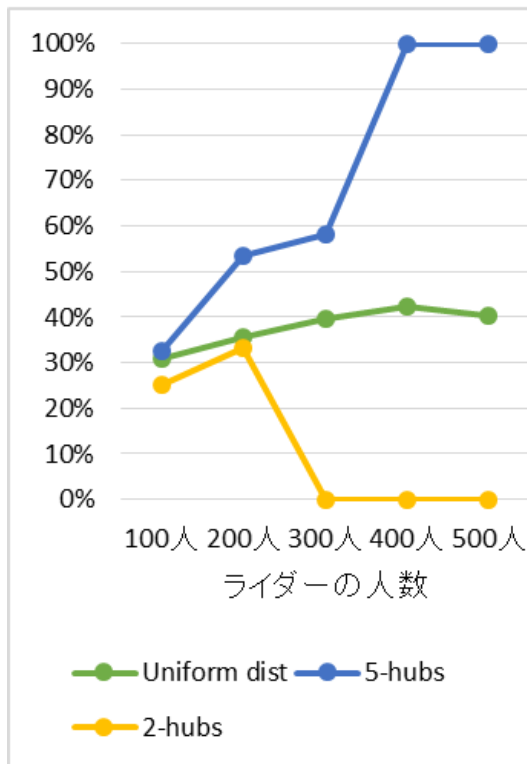


達成率98%

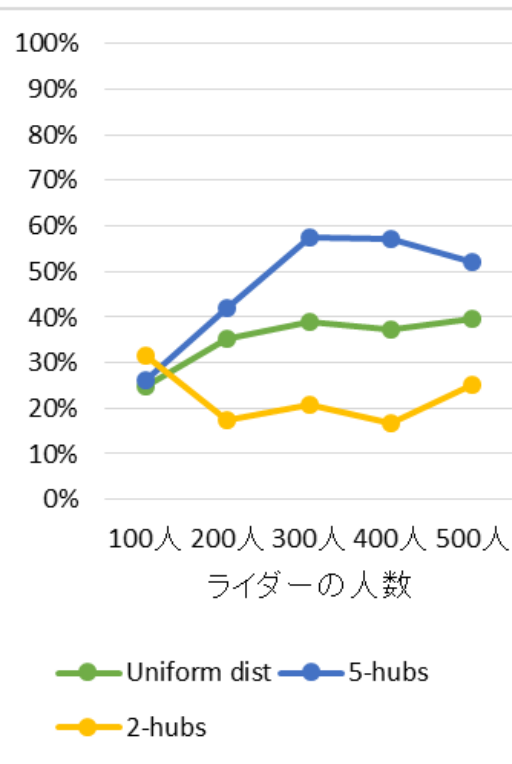


- 2-hubsを除き，参加者が増えるほどコスト削減効果が高い
- 参加者の調整幅が大きくなることによるコスト削減効果は，5-hubs > Uniform dist. > 2-hubs

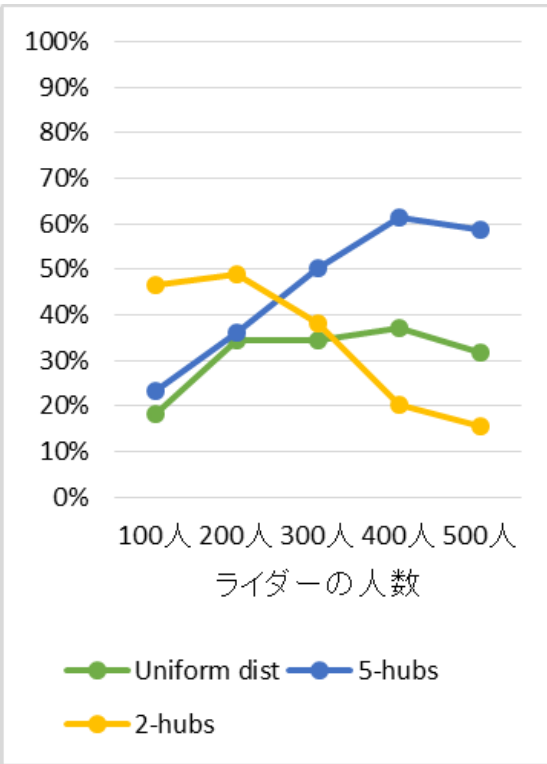
達成率90%



達成率95%

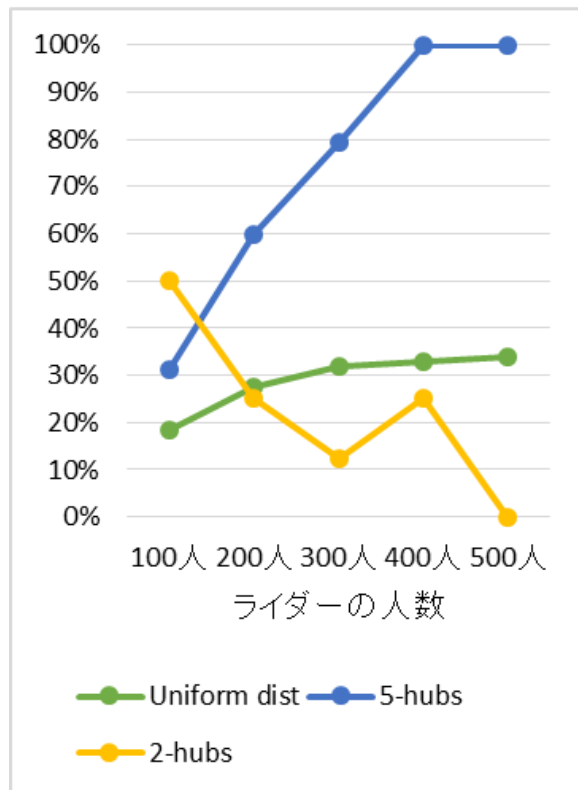


達成率98%

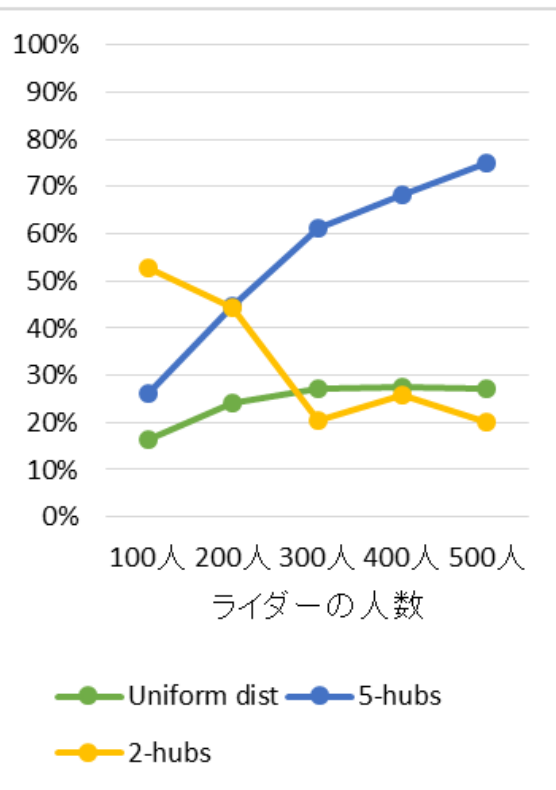


- 2-hubsを除き，参加者が増えるほどドライバー削減効果が高い
- 参加者の調整幅が大きくなることによるドライバー削減効果は，5-hubs > Uniform dist. > 2-hubs の傾向がある

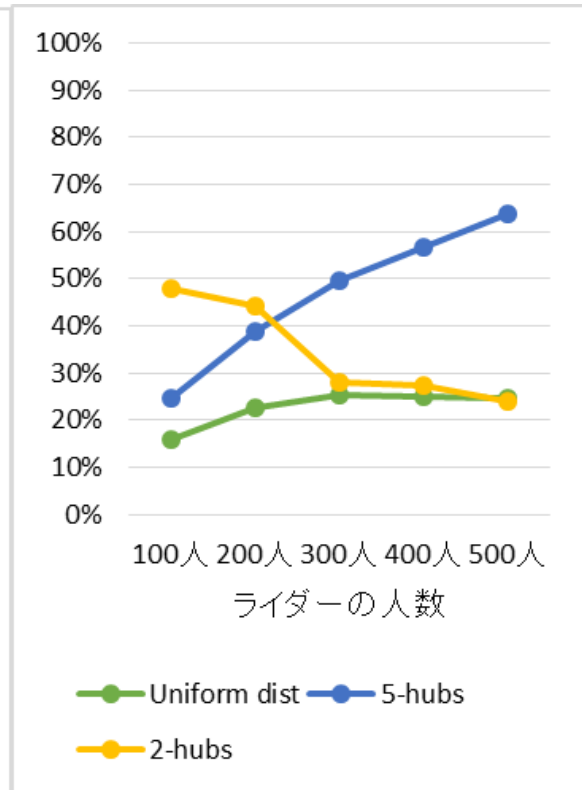
達成率90%



達成率95%



達成率98%



- 5-hubsの場合，参加者の調整幅が広がることにより，参加者同士のマッチング率が高くなっている
- サービスレベルが上がるほど，シェアの削減率は下がる（依然として専属ドライバーが必要とされている）

## ■本研究の成果

- ライドシェアシステムに専属ドライバーを導入した場合の効果について分析ができるような定式化を行った
- 非線形整数計画問題について解法の提案を行い、その有効性を示した
- 参加者の密度・トリップパターンの類似度・時刻制約の柔軟性に依存して、オペレーションコスト・必要なドライバーの人数が変わることを示した

## ■拡張の方向性

- ドライバーの譲歩に対する補償を記述することで、cost-feasibleなペア以外でマッチングさせる
- マッチングが何度も失敗した場合に補償する（ライダーの差別化を行う）