

閉鎖型待ち行列ネットワークの 最適拠点配置への適用と シミュレーションとの連携

静岡理工科大学総合情報学部コンピュータシステム学科

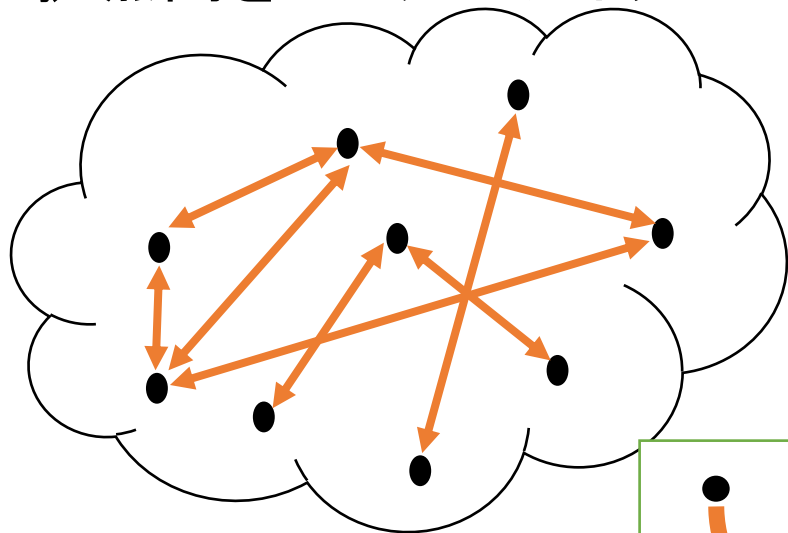
鈴木宏

池田清花

指導教員 水野信也

乗り捨て型レンタルサイクル(1)

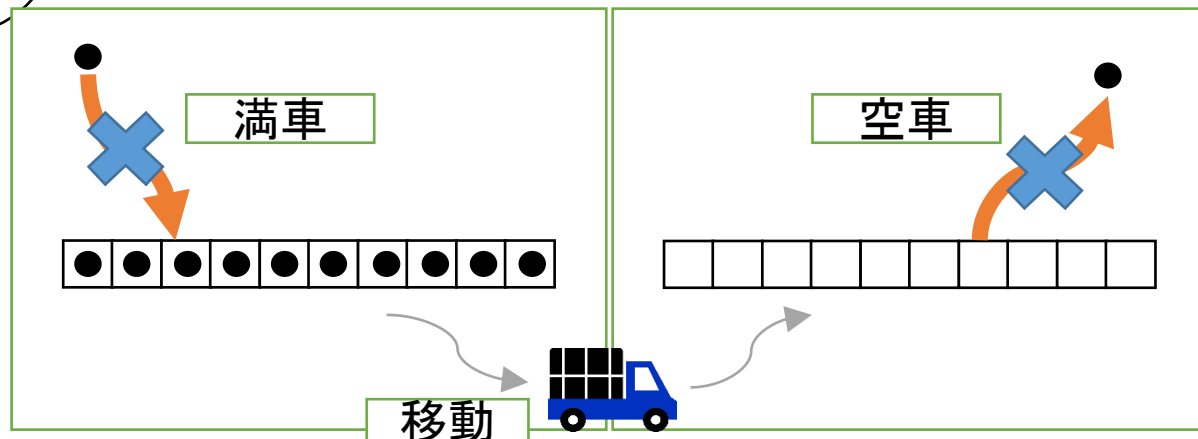
- 自転車を共同利用するシステム(シェアサイクル)
- 借りた場所とは違う場所で返却できる
- 拠点間をレンタルサイクルで簡単に移動可能



問題点

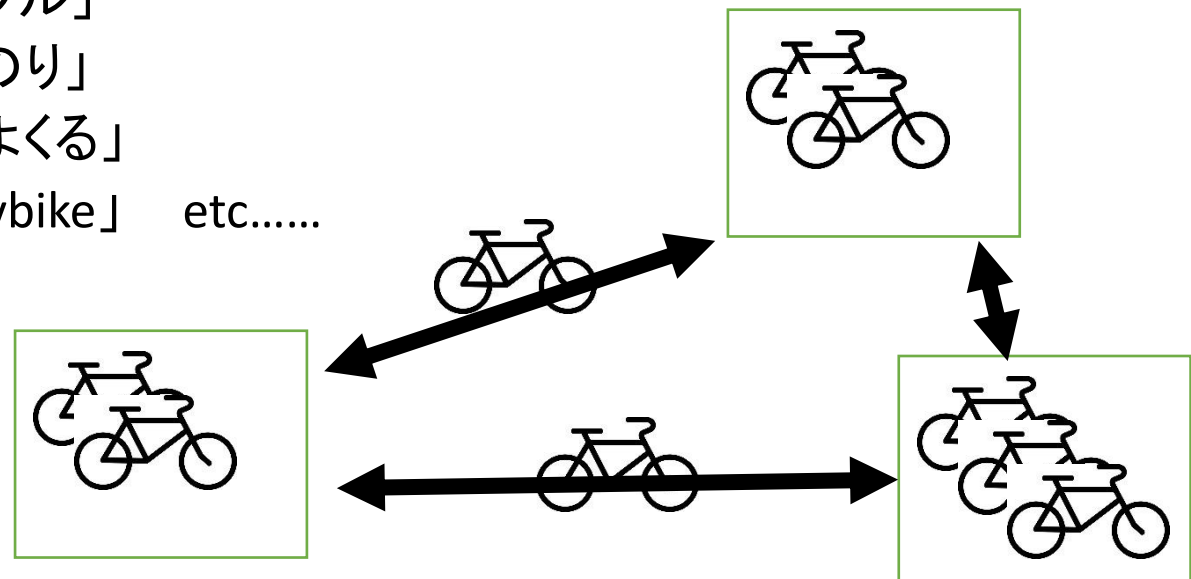
- 移動用具の拠点集中
- 移動用具の呼損

→利用率の低下



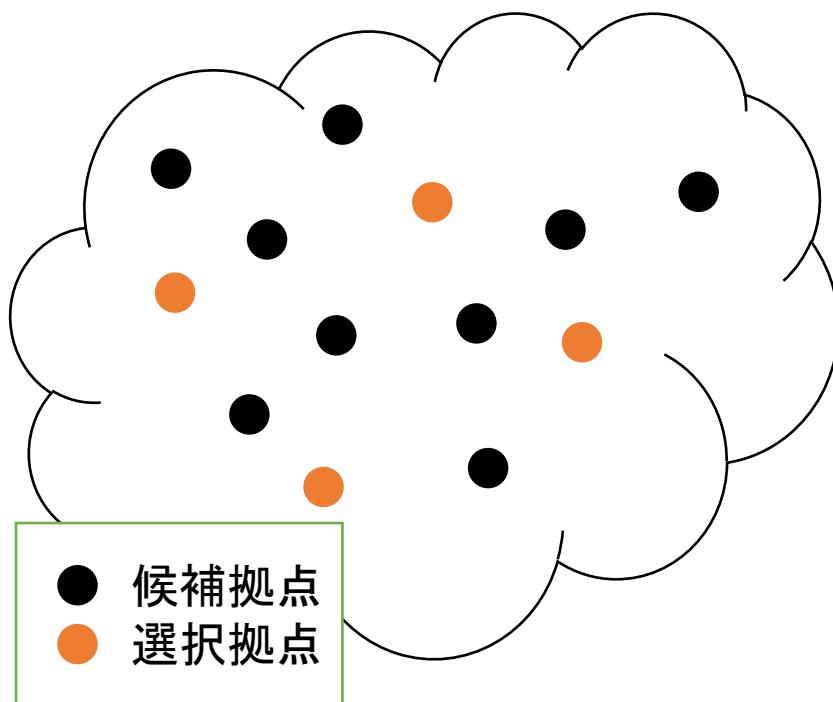
乗り捨て型レンタルサイクル(2): 事例

- 現在欧米を中心に世界中の500 都市以上で運用されている
- 欧米: 平均 移動用具5,000台、1,000拠点の規模で実施
 - イギリス ロンドン「Barclays Cycle Hire」
 - フランス パリ「velib'」 etc.....
- 日本: 平均 移動用具100台、20拠点程度での実施
 - 北海道札幌市「ポロクル」
 - 石川県金沢市「まちなのり」
 - 東京都千代田区「ちよくる」
 - 神奈川県横浜市「baybike」 etc.....



研究概要

- 候補拠点の中から系内台数が分散される拠点を選択する
- 選択拠点の理論値に対し、シミュレーションの実行
- 計算では得られないデータをシミュレーションで求める
- 拠点再選択を行うためのパラメータを再設定



理論値の計算と拠点の選択
↓
選択拠点でのシミュレーションを実行
↓
シミュレーション結果の考察と
パラメータ, 目的関数の再設定

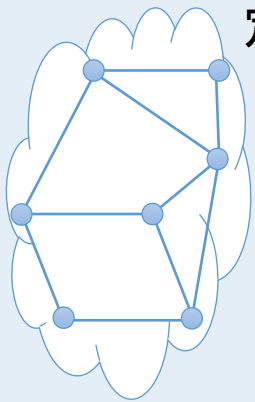
最適配置解析基盤:どこに何台配置する？

モデル化

◎閉鎖型待ち行列ネットワーク

→各拠点の平均系内台数,
スループット算出

→各拠点の平均ノード滞在時間の算出



定常分布

$$\pi(n_1, n_2, \dots, n_K) = \frac{1}{G(N, K)} \prod_{k=1}^K \sigma_k^{n_k}$$

$$\sigma_k = \alpha_k / \mu_k$$

$$G(N, K) = \sum_{n_1 + n_2 + \dots + n_K = N} \prod_{k=1}^K \sigma_k^{n_k}$$

推移確率行列の算出

重力モデルを利用

$$f_{ij} = C \frac{p_i^a q_j^b}{d_{ij}^c}$$

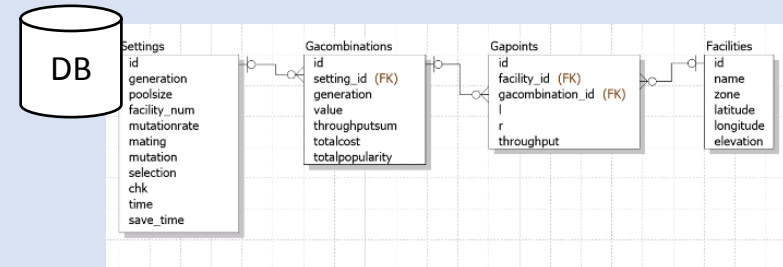
拠点パラメータ

→拠点人気度、拠点間距離

最適化計算:各拠点の平均系内台数を分散

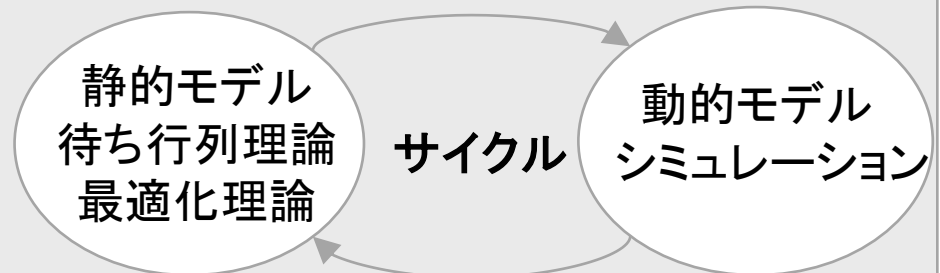
目的関数

$$\text{最小化} \sum_{k=1}^K (|L_k - CP_k| + PT_k \cdot I(\bar{A}_k))$$



シミュレーション:

動的モデルからモデル整合性チェック



整合性確認

静的モデルと動的モデルのサイクル

閉鎖型待ち行列ネットワークの定義

- 網内でサービスされる移動台車は一種類
- 網内の拠点数: K 個
- 移動台車の総数: $N = \sum_{k=1}^K n_k$ 拠点 k の移動台車数: n_k
- サービス率: μ_k
 - サービス率 μ_k が高いほど利用率は高くなる
 - サービス率 μ_k は人気度に比例する
 - サービス時間はサービス率 μ_k の指数分布に従う
- 拠点 k に網内から到着する移動台車の全到着率: α_k
- トラフィック方程式:

$$x_1 = 1, \quad x_k = \sum_{j=1}^K x_j p_{j,k}, \quad (k = 2, 3, \dots, K)$$

- $x_k = \alpha_k$ となる
- 拠点 $i \rightarrow j$ の推移確率:

$$p_{i,j} (1 \leq i, j \leq K, p_{i,j} > 0, \sum_{j=1}^K p_{i,j} = 1)$$

推移確率 p_{ij}

- 各拠点ごとの移動推移確率は重力モデルに従って求める
- f_{ij} : 拠点*i*から拠点*j*への移動量
- p_i : 拠点*i*を出発地とする移動総量
- q_j : 拠点*j*を出発地とする移動総量
- d_{ij} : 拠点*i, j*間の距離

$$f_{ij} = C \times \frac{p_i^a q_j^b}{d_{ij}^c},$$

移動距離による影響を減らすため
 $a = b = 1$, $c = 0.5$, $C = 0.1457$
で計算

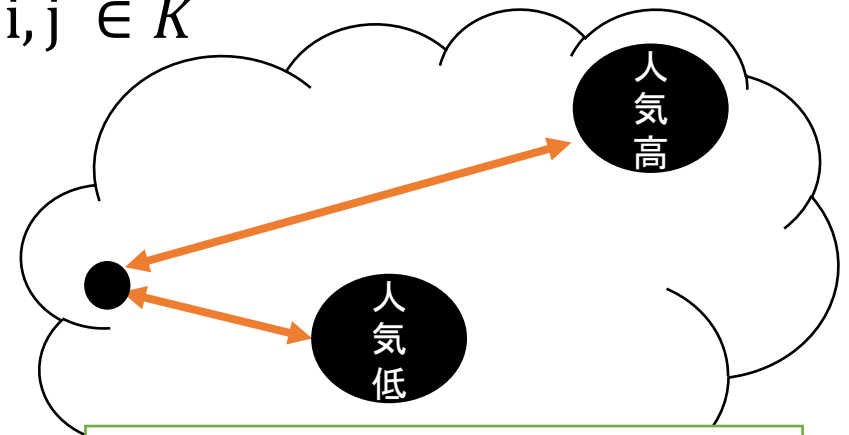
今回は

p_i : 拠点*i*の人気度

q_j : 拠点*j*の人気度

とする

$i, j \in K$



近くても遠くても人気度によって
推移確率は同じくらいとなる

定常分布

- $\sigma_k = \frac{\alpha_k}{\mu_k}$ とおくと、拠点1に n_1 台, $\dots$, K に n_k 台いるという定常分布は

$$\pi(n_1, n_2, \dots, n_K) = \frac{1}{G(N, K)} \prod_{k=1}^K \sigma_k^{n_k}$$

- ただし

$$G(N, K) = \sum_{n_1 + n_2 + \dots + n_K = N} \prod_{k=1}^K \sigma_k^{n_k}$$

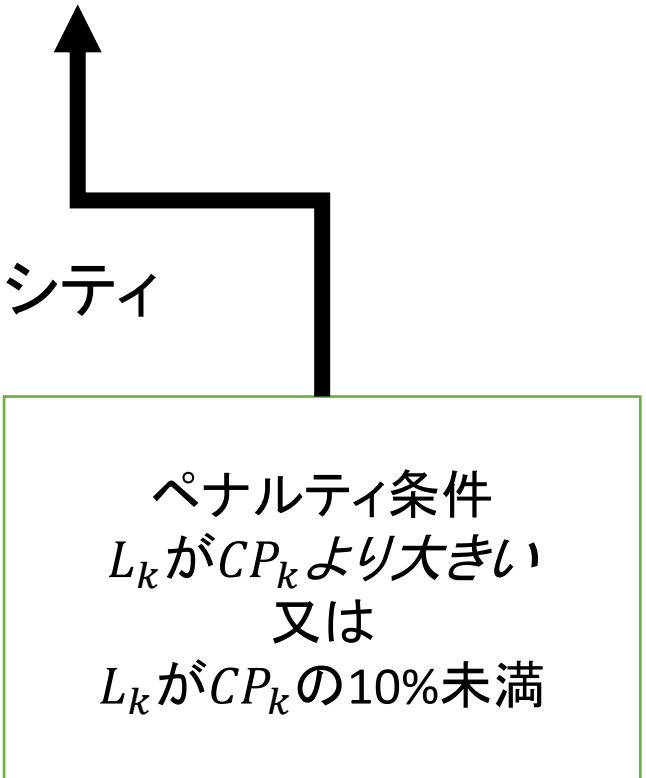
- リトルの公式と平均値解析法を用いて、以下の値を求めた
 - スループット(単位時間当たりの移動台数) : λ_k
 - 平均ノード滞在時間 : R_k
 - 平均系内台数 : L_k

拠点選択の目的関数

- 各拠点にある移動用具を分散させる目的関数

$$\text{最小化} \quad \sum_{k=1}^K (|L_k - CP_k| + PT_k \cdot I(\overline{A_k}))$$

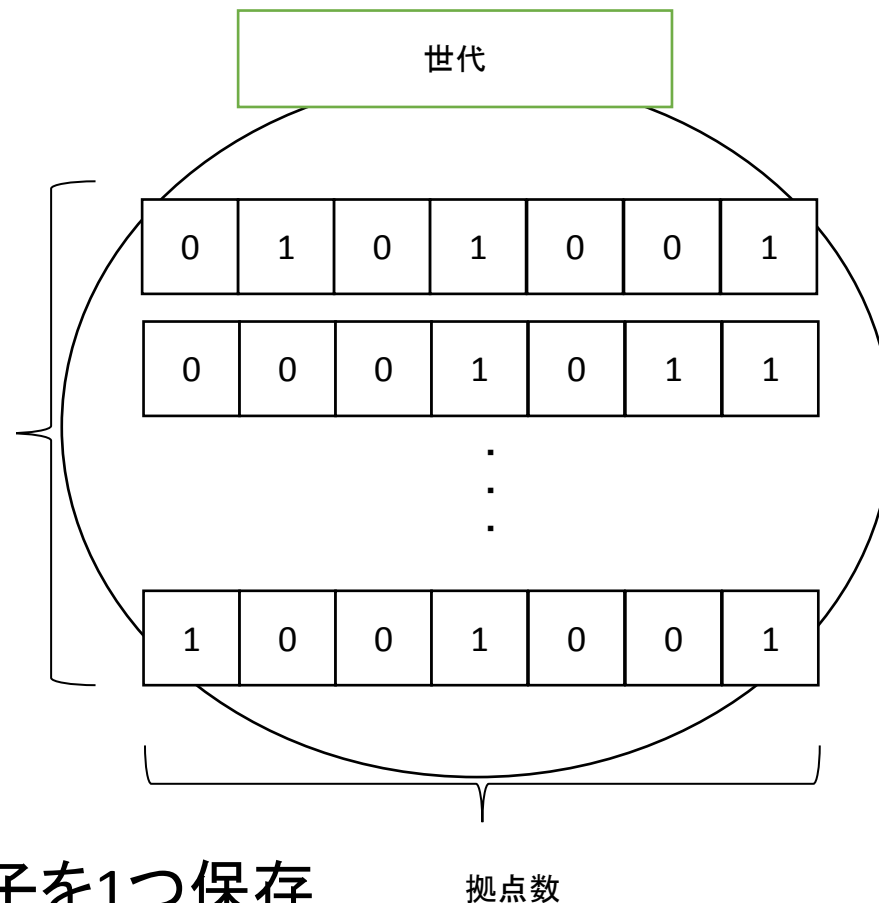
- L_k : 拠点 k における平均系内台数
- CP_k : 拠点 k における移動用具台数のキャパシティ
- PT_k : 拠点 k におけるペナルティ
 - ペナルティはある程度大きな数(今回は50)
- A_k : 拠点 k において安定している事象
 - 多すぎても少なすぎても駄目
- $I(\overline{A_k})$: 事象 A_k における指示関数



ペナルティ条件
 L_k が CP_k より大きい
又は
 L_k が CP_k の10%未満

最適化法：遺伝的アルゴリズム(GA)

- 遺伝子は0/1で選択拠点を表現
- 数値の設定
 - 反復数
 - プールサイズ(世代遺伝子数)
 - 突然変異率
 - 交叉法
 - 選択法



- 初期世代はランダムで生成
- 世代を回し、各世代の最良遺伝子を1つ保存

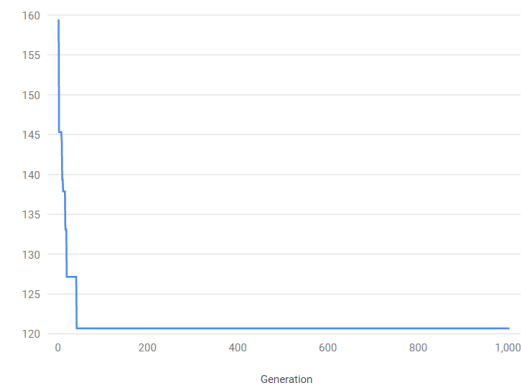
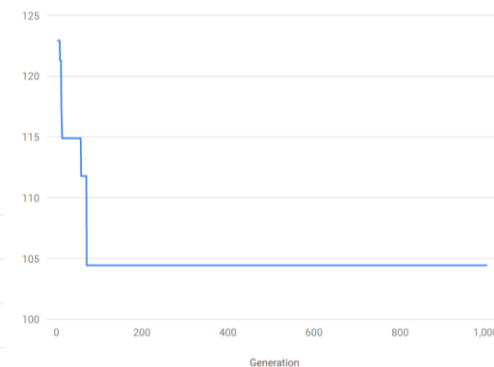
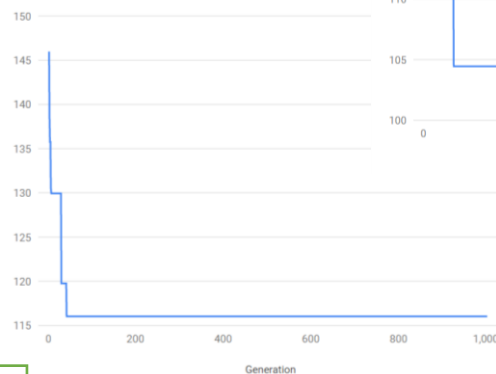
最適化法(2): GA

- プールサイズ:50 反復数:1000で何度か実行
- ほとんどが100世代以内で収束した

実行時間短縮のため、設定は
以下のようにした

設定

- 反復数 : 100世代
- プールサイズ : 50
- 突然変異率 : 0.3
- 交叉法 : 一点交叉
- 突然変異 : 交換
- 選択法 : ルーレット選択



モデル地：浜名湖ガーデンパーク(1)

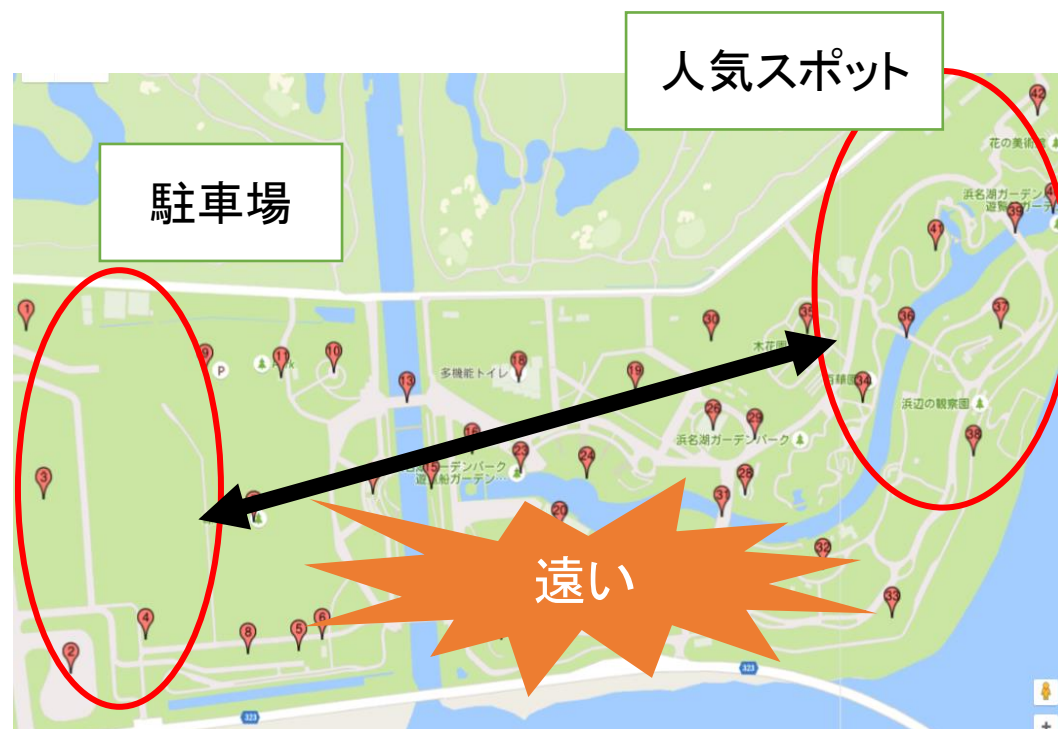
- 浜名湖花博の会場として使われていた場所
- 庭園や自然があり、四季を通じて楽しむことができる
- 子どもからお年寄りまで幅広い年齢の人が訪れる
- 面積が56ヘクタール(560,000m²→東京ドーム約12個分)
 - ディズニーランドより広い



モデル地：浜名湖ガーデンパーク(2)

- ・ 駐車場から人気スポットが遠い → 遠距離へ楽に移動したい
→ 自動運転車を設置しよう → 設置場所を考える

- ・ 候補拠点数：42
- ・ 選択拠点数：12
- ・ 人気度：5 または 10
- ・ サービス率：
人気度に比例
- ・ キャパシティ：10
- ・ 移動用具台数：100



人の集まる場所の人気度を10としている(駐車場、展望塔、庭園)
それ以外の場所は5としている

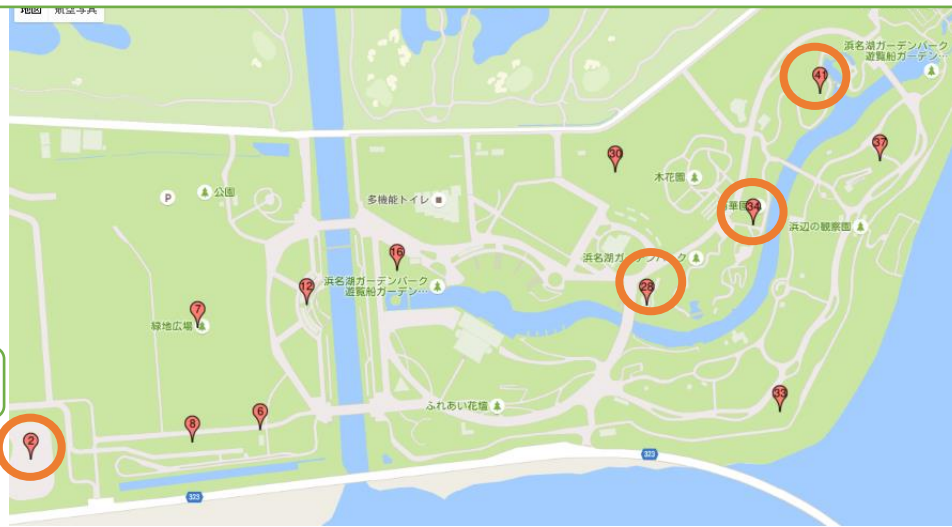
最適化計算結果(1) 選択拠点数 $K = 12$

- 同じ条件で何度かGAを実行した

ID	世代数	目的関数	スループット合計	コスト合計	人気合計
1	100	106	55	127	65
2	100	98	72	119	90
3	100	95	56	112	65
4	100	86	69	130	80
5	100	104	62	130	80

目的関数が最も最小 → この結果からシミュレーションを行う

駐車場



人気スポット

← 選択拠点

最適化計算結果(2) 選択拠点数 $K = 12$

- 人気度の高い場所は網内からの客の到着率が高い

推移確率行列

2	6	7	8	12	16	28	30	33	34	37	41
0.001	0.099	0.111	0.119	0.088	0.077	0.121	0.061	0.055	0.111	0.051	0.105
0.139	0.000	0.112	0.127	0.112	0.081	0.106	0.054	0.046	0.094	0.042	0.086
0.159	0.115	0.000	0.129	0.103	0.076	0.102	0.052	0.045	0.091	0.041	0.086
0.170	0.130	0.129	0.000	0.090	0.071	0.100	0.051	0.044	0.090	0.040	0.084
0.125	0.113	0.102	0.089	0.000	0.111	0.115	0.060	0.049	0.100	0.044	0.093
0.112	0.085	0.077	0.072	0.114	0.000	0.139	0.073	0.055	0.116	0.050	0.105
0.088	0.055	0.052	0.051	0.059	0.069	0.001	0.117	0.090	0.202	0.069	0.149
0.081	0.051	0.048	0.047	0.056	0.067	0.215	0.000	0.067	0.168	0.062	0.138
0.079	0.047	0.045	0.044	0.049	0.054	0.177	0.073	0.000	0.199	0.080	0.153
0.078	0.047	0.045	0.044	0.050	0.056	0.195	0.089	0.098	0.001	0.092	0.207
0.072	0.042	0.040	0.040	0.044	0.048	0.135	0.065	0.079	0.185	0.000	0.249
0.080	0.047	0.045	0.045	0.050	0.055	0.156	0.079	0.082	0.225	0.135	0.001

理論値計算結果(1) 選択拠点数 $K = 12$

- 平均系内台数はうまく分散されている
- 人気度の高い場所もサービス率が高いためスループットが高くなり、系内台数はキャパシティ内に収まっている

ID	拠点名	人気度	到着率	平均系内台数	平均滞在時間	スループット
2	南ロータリー	10	1.000	1.769	0.276	6.398
6	南車椅子貸出し場	5	0.715	9.663	2.113	4.572
7	緑地広場	5	0.697	7.773	1.742	4.462
8	南駐車場	5	0.699	7.958	1.778	4.475
12	ほほえみの森	5	0.708	8.891	1.962	4.532
16	水遊び広場	5	0.686	6.828	1.556	4.389
28	展望塔	10	1.381	7.168	0.811	8.834
30	風の森	5	0.753	18.17	3.772	4.818
33	湖岸の森	5	0.701	8.071	1.801	4.482
34	百華園1	10	1.428	9.573	1.048	9.135
37	里の見晴台	5	0.709	9.021	1.987	4.539
41	百華園2	10	1.313	5.114	0.609	8.403



- この拠点だけ平均系内台数がキャパシティオーバーとなっている

理論値計算結果(2) 選択拠点数 $K = 12$

- 平均系内台数がキャパシティオーバーとなっている拠点は人気度の高い拠点から等しく距離が近い



シミュレーションの設定(1)

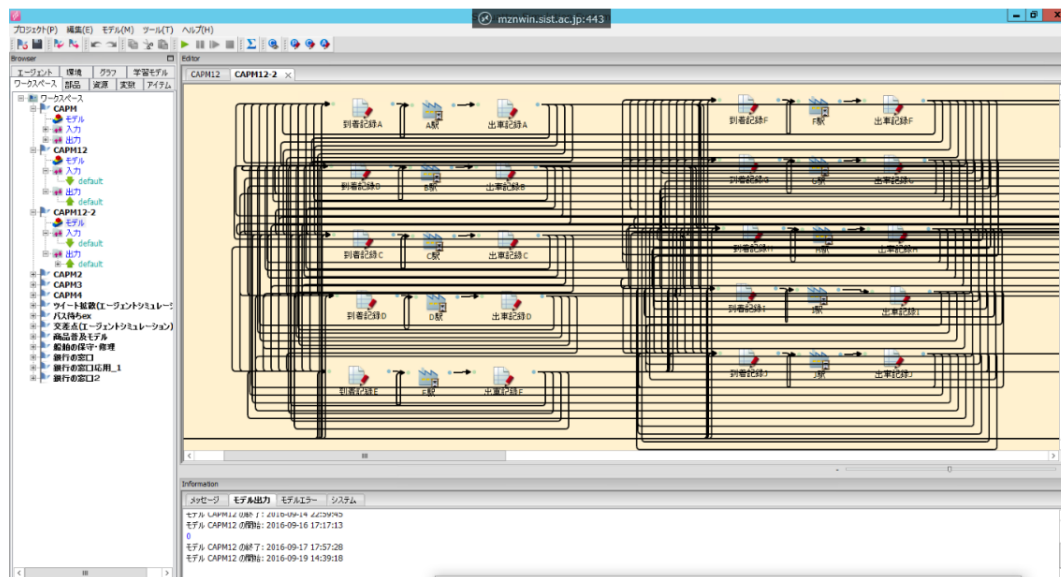
- 最適化、理論値計算の結果からシミュレーションを実施する

最適化計算
理論値計算

閉鎖型待ち行列
ネットワーク

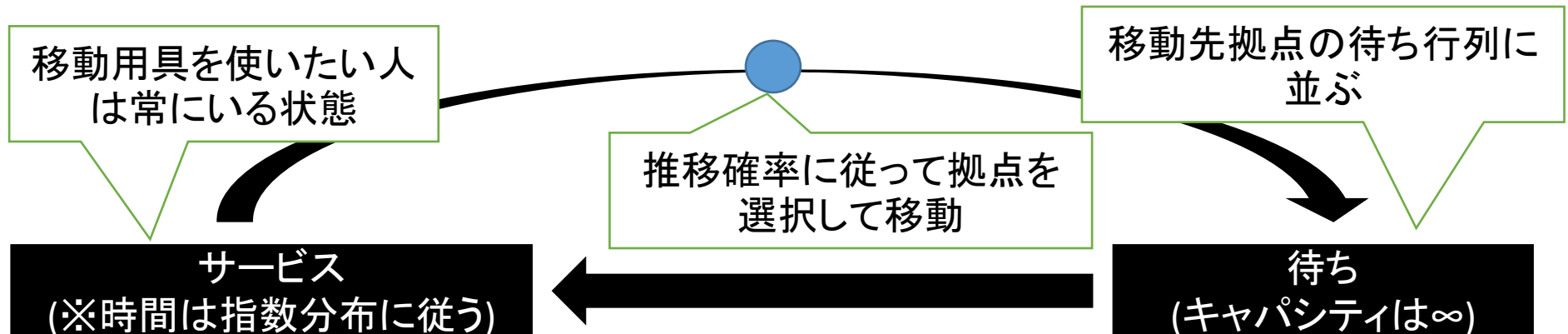
シミュレーションソフト
 S^4 に数値を設定

選択拠点



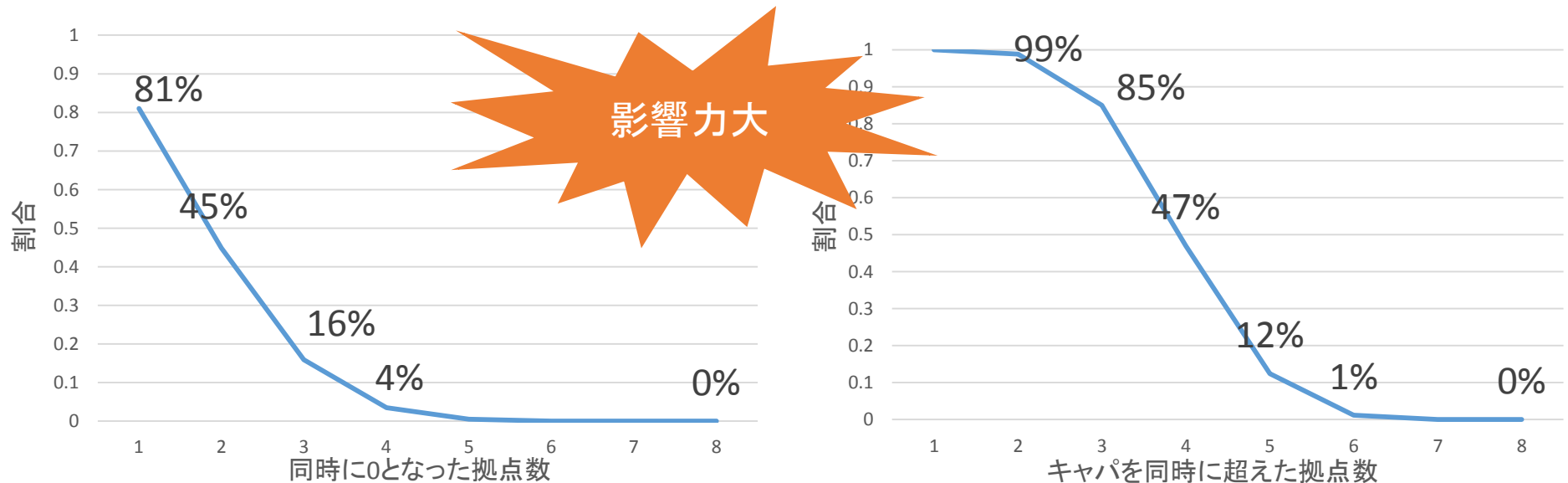
シミュレーションの設定(2)

- 利用者は充分にいるものとする
- 移動用具(アイテム)は一種類
- 各拠点は独立しているものとする
- 初期配置は全拠点到一様に配置する
- 移動時間は0
- 全ての拠点到最適化と同じパラメータを入力して実行
 - 理論値で算出された推移確率を各拠点到設定
 - サービス時間(サービス率 μ_k の指数分布に従う)を設定



系内台数が0 または キャパを超える時間割合

- 系内台数が同時に0となってしまう時間割合(左)
 - 1拠点以上が同時に0となってしまう時間割合は81%
 - 2拠点でも45%と非常に高い
- 系内台数が同時にキャパシティを超えてしまう時間割合(右)
 - 2拠点以上が同時にキャパオーバーとなる時間割合は99%
 - 3拠点でも85%と非常に高い
- シミュレーションを実施し、利用者にとって不利益な状態が明確に！



各拠点同士の系内台数の相関関係(1)

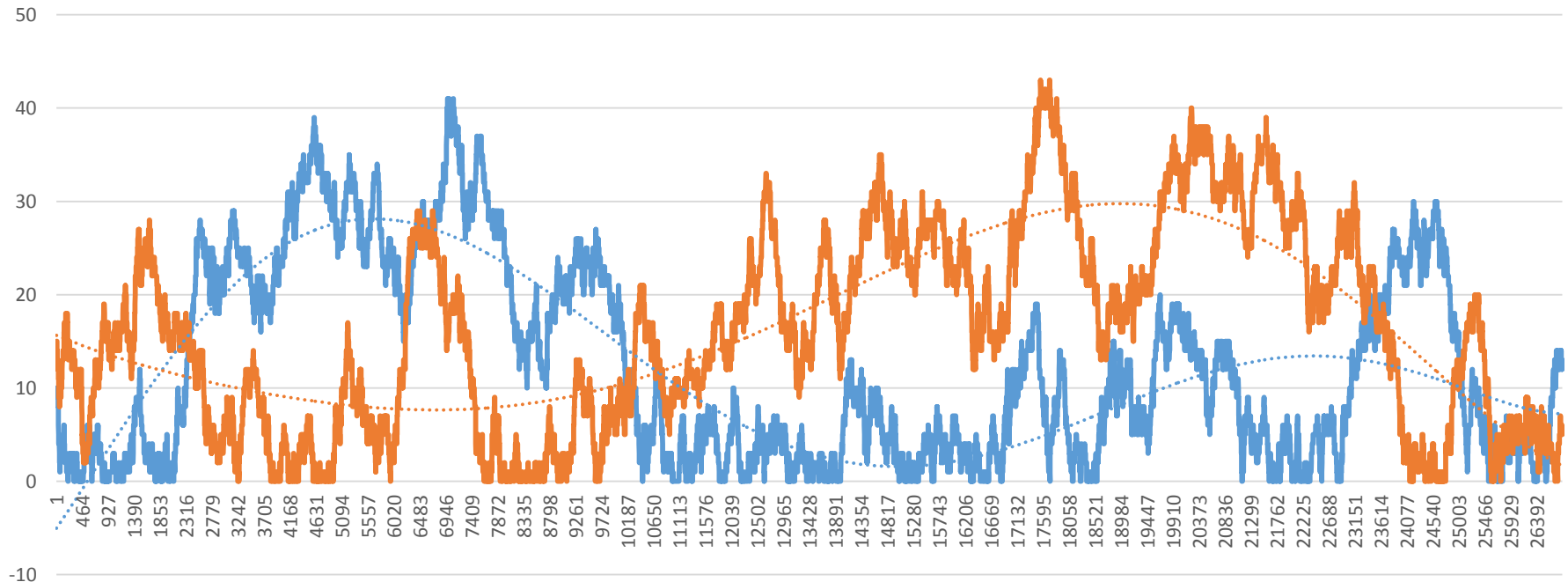
- 平均系内台数がキャパシティオーバーとなっていた拠点30に各拠点との負の相関がある
- その中で最も負の相関のある拠点は30,12の-0.24である

	2	6	7	8	12	16	28	30	33	34	37	41
2	1	-0.03192	-0.04529	-0.0256	-0.01774	0.002004	0.010847	-0.04936	-0.01997	-0.03281	0.002434	-0.02679
6	-0.03192	1	-0.12631	-0.08612	-0.09084	-0.0017	-0.06448	-0.19908	-0.0767	-0.05278	-0.14516	-0.09406
7	-0.04529	-0.12631	1	0.038357	0.017712	-0.10159	-0.07181	-0.17958	-0.13451	-0.09614	-0.10031	-0.07254
8	-0.0256	-0.08612	0.038357	1	-0.02302	-0.10712	-0.04289	-0.16814	-0.12565	-0.11479	-0.07573	-0.07871
12	-0.01774	-0.09084	0.017712	-0.02302	1	-0.06803	-0.10472	-0.23646	-0.06886	-0.14619	-0.17598	-0.08627
16	0.002004	-0.0017	-0.10159	-0.10712	-0.06803	1	-0.08852	-0.13525	-0.08406	-0.09625	-0.08479	-0.05687
28	0.010847	0.01084	0.01084	0.01084	0.01084	0.01084	1	-0.10808	-0.04995	-0.07318	-0.09109	-0.05204
30	-0.04936	-0.19908	-0.17958	-0.16814	-0.23646	-0.13525	-0.10808	1	-0.17212	-0.12975	-0.18398	-0.12771
33	-0.01997	-0.0767	-0.13451	-0.12565	-0.06886	-0.08406	-0.04995	-0.17212	1	-0.16416	-0.0615	0.026586
34	-0.03281	-0.05278	-0.09614	-0.11479	-0.14619	-0.09625	-0.07318	-0.12975	-0.16416	1	-0.09413	-0.05233
37	0.002434	0.00243	0.00243	0.00243	0.00243	0.00243	0.00243	0.00243	0.00243	0.00243	1	-0.01632
41	-0.02679	-0.09406	-0.07254	-0.07871	-0.08627	-0.05687	-0.05204	-0.12771	0.026586	-0.05233	-0.01632	1

各拠点同士の系内台数の相関関係(2)

- 時間系列に系内台数の変化をグラフとした
- 拠点12 拠点30(キャパオーバー)
- 一方が増えたと一方が減る、負の相関関係がある

拠点12と拠点30の系内台数



- 問題のある拠点がどの拠点かわかる

各拠点同士の系内台数の相関関係(3)

- ・負の相関がある拠点の位置関係
 - ・ 人気度の高い拠点との中間にある拠点との相関があった



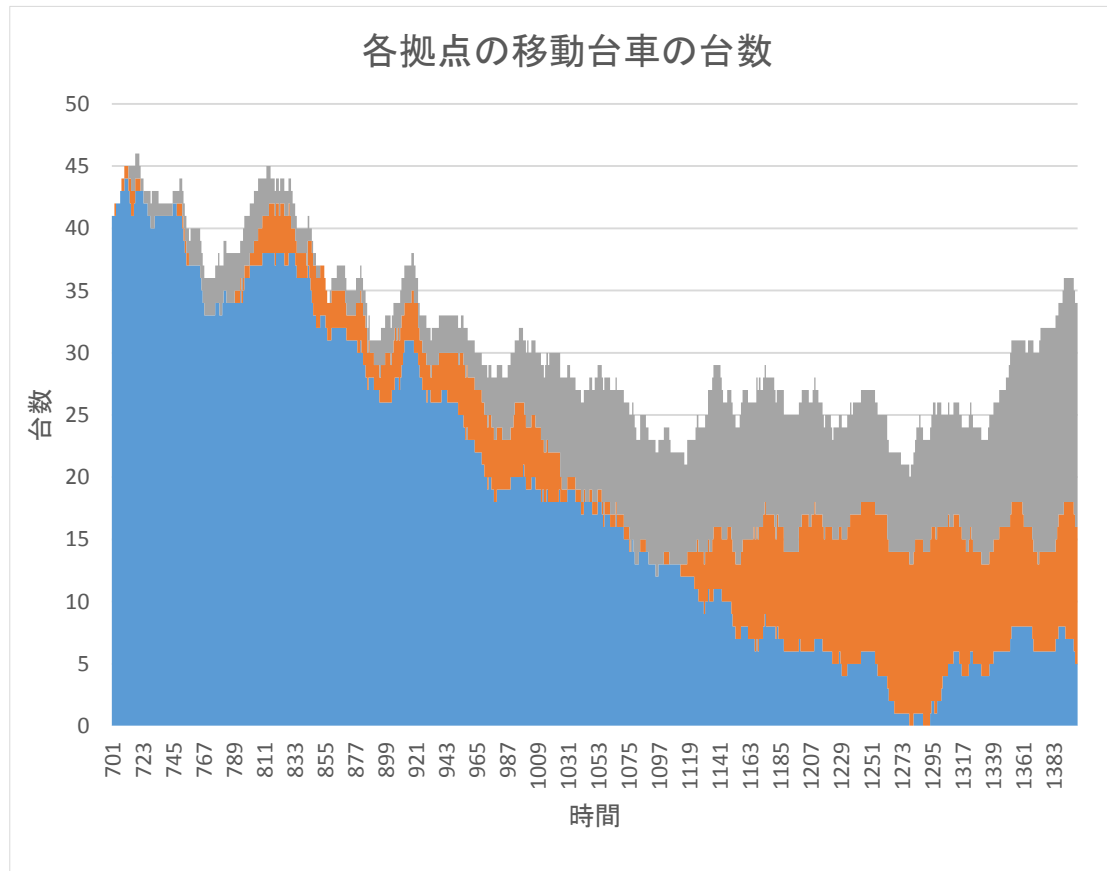
立ち上がり考察のための設定

- 基本的な設定は同じ
- 初期配置を全ての人が最初に来る駐車場へ配置
- シミュレーション時間1500秒までの結果を考察する



シミュレーションの立ち上がりの考察(1)

- 拠点間距離が近いほど移動するものが多かった
- 人気拠点との中間拠点にも多く移動した

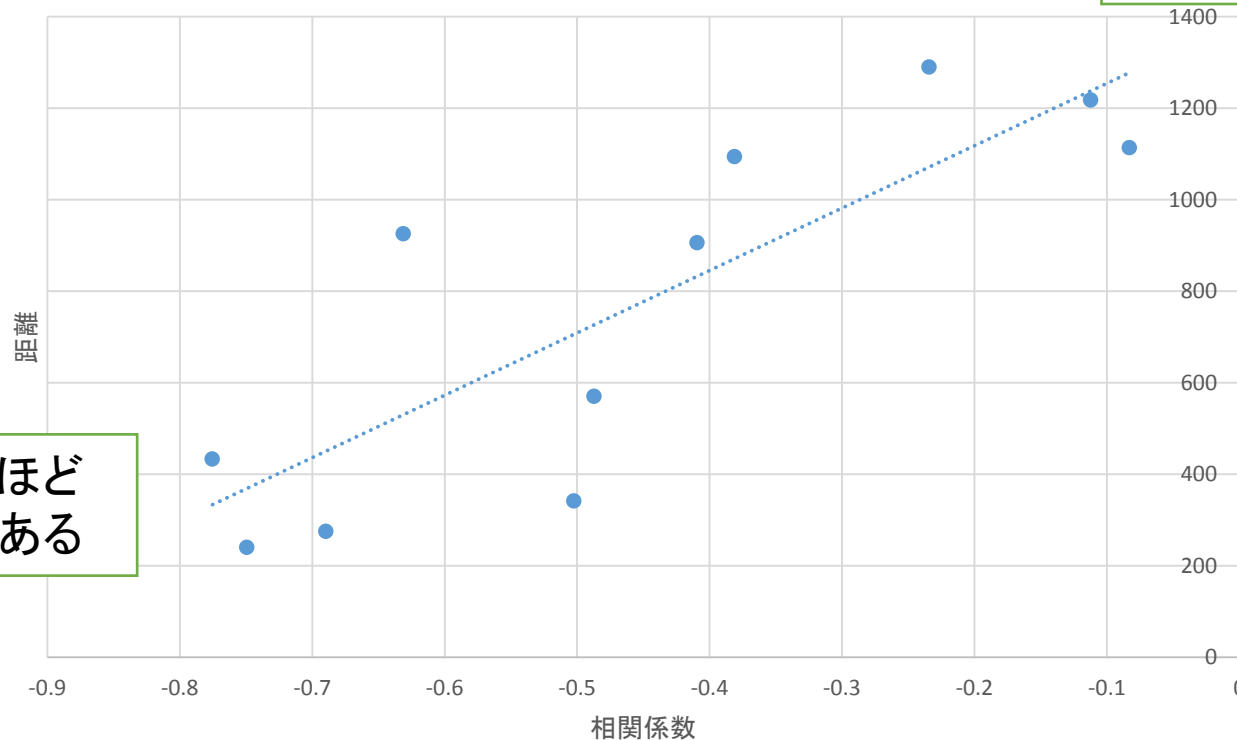


シミュレーションの立ち上がりの考察(2)

- 距離と各拠点の台数の相関の分布

駐車場からの距離と台数の相関係数の分布

距離が遠いほど
相関がなくなる



距離が近いほど
負の相関がある

- 近くへ移動する人が多すぎるため推移確率の求め方を考え直す必要がある

まとめ・今後の課題

- 閉鎖型待ち行列ネットワークの理論値に対し、シミュレーションを行った
- シミュレーション
 - 呼損やキャパシティオーバーの時間割合を算出した
 - 希少な事象が起こる確率を算出した
 - 影響度の高い拠点の組み合わせを算出し、可視化した
 - 立ち上がりを見ることで重力モデルの問題を提起した
- 最適なパラメータ設定
 - 人気度、サービス率、キャパシティを現実に近いものとする
 - 推移確率で扱う距離が直線距離となっているため道のりに変更する
- 適切な目的関数の設定
 - キャパシティによるペナルティのみ→待ち時間やコストを組み込む
- シミュレーション結果の反映
 - 今回は理論値→シミュレーションのみのため、今後、シミュレーション→理論値へ戻していく方法を考えていく
- 顧客モデルを作成し、シミュレーションに組み込む