

利用者の不満度を考慮した デマンドバスシステムに関する研究

東京理科大学工学部経営工学科

沼田研究室

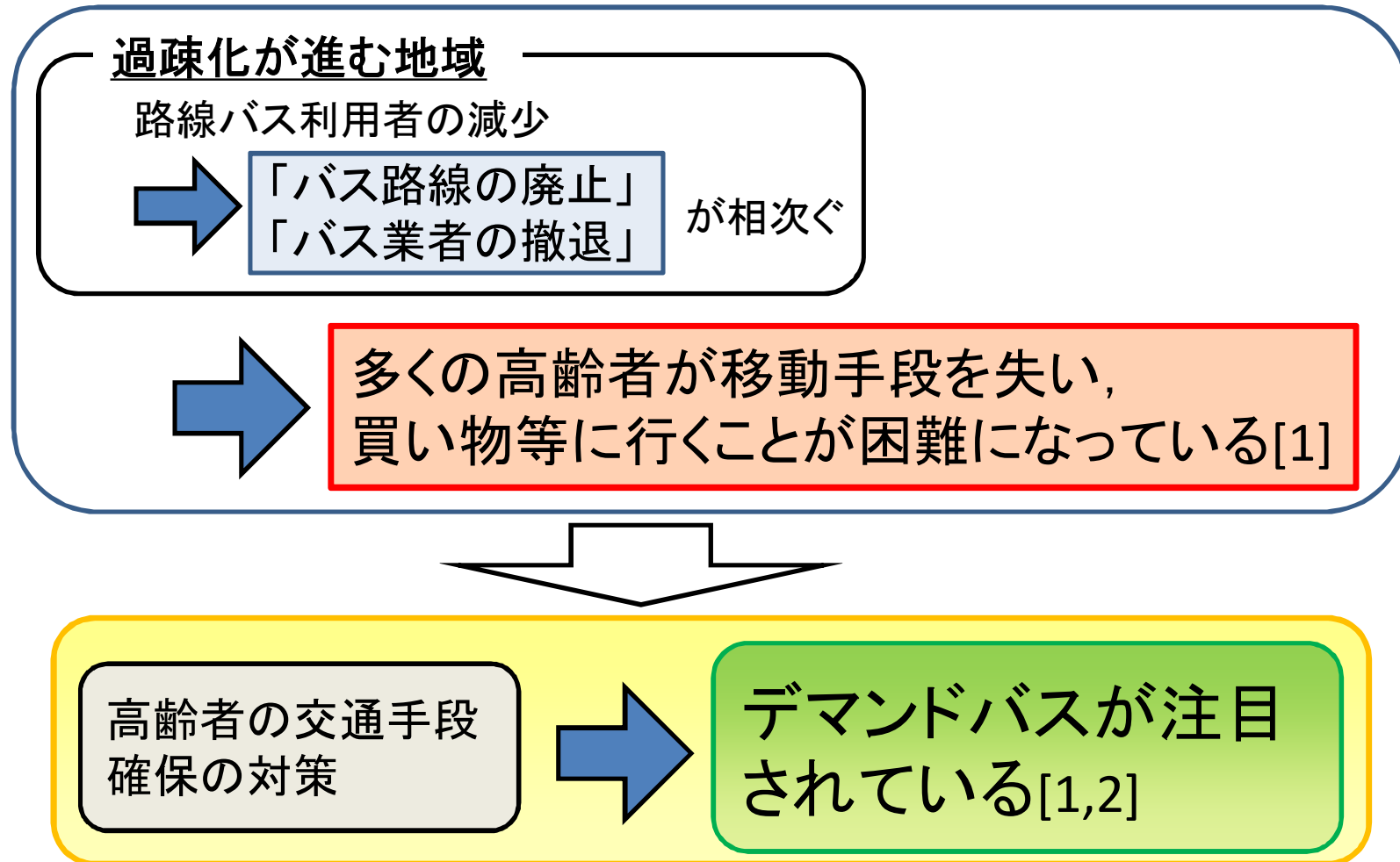
4409005 井上和弥

目次

1. はじめに
2. 本研究で扱う問題
3. 提案解法
4. 数値実験
5. まとめと今後の課題

1.はじめに

1.1 研究背景(1)



1.1 研究背景(2)

デマンドバス

電話等による事前予約を受け, 利用者が希望する出発地から目的地まで送迎する乗り合い型の交通サービス

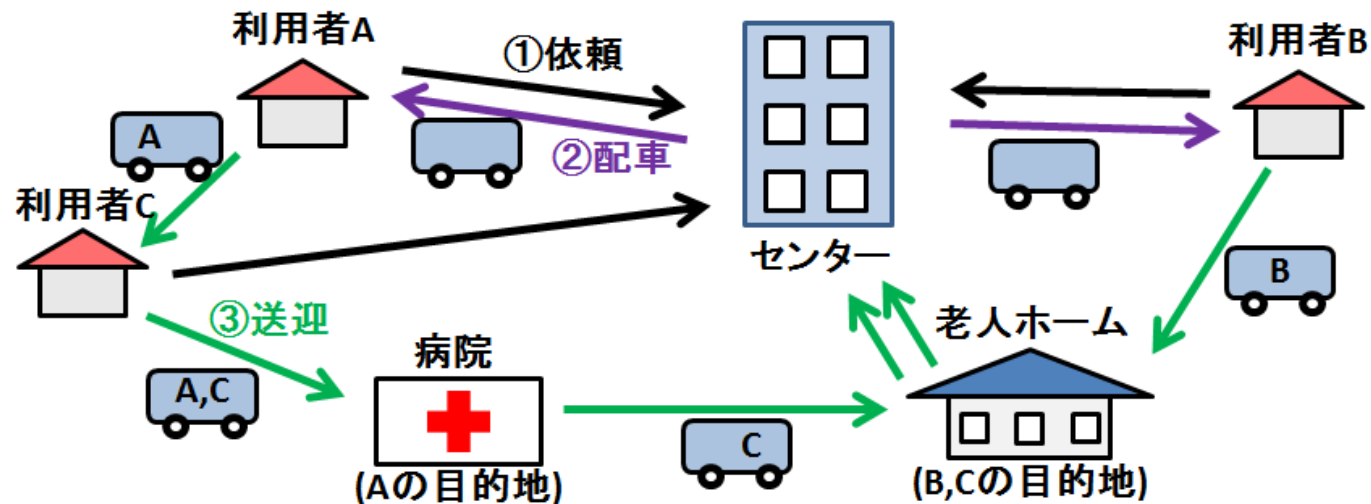
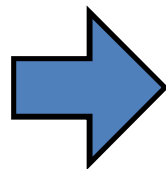


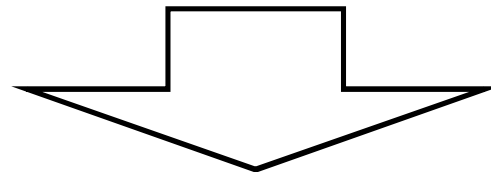
図1: デマンドバスシステム

1.1 研究背景(3)

利用者の要求(希望乗車時間, 希望到着時刻)を満たす送迎を行うには多くの車両が必要[1]



**車両維持に莫大な費用
がかかっている**

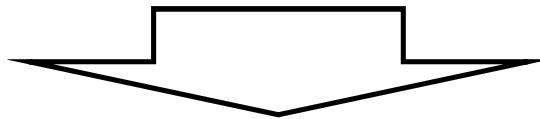


継続的な運営を
おこなうために

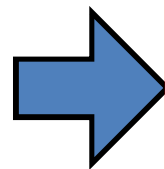
**少ないバスの台数で
効率的な送迎を行うことが必要**

1.1 研究背景(4)

- バスの台数の削減により, 利用者の乗車/降車希望時刻に間に合わない



- 希望した到着時刻までに目的地に到着できないことは, 「待ち合わせ時間」, 「病院などの予約時間」への遅れにつながる



利用者が大きな不満を抱く

その為

少ないバスの台数で, 遅延時間を最小にするバスの運行計画を決定することが重要

1.2 研究の目的

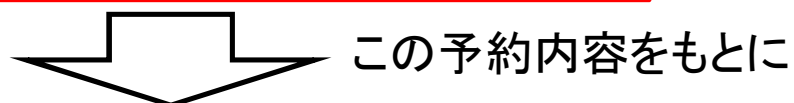
バスの台数を制限し，到着場所へのバスの遅延時間を最小化するデマンドバスの運行計画問題を提起し，その解法を提案する

2. 本研究で扱う問題

2.1 問題の概要

- 利用者が前日までに連絡すること

①乗車希望時刻 ②乗車場所
③降車希望時刻 ④降車場所



- 複数のバスがデポを出発し、各利用者の送迎を行って再びデポに戻る

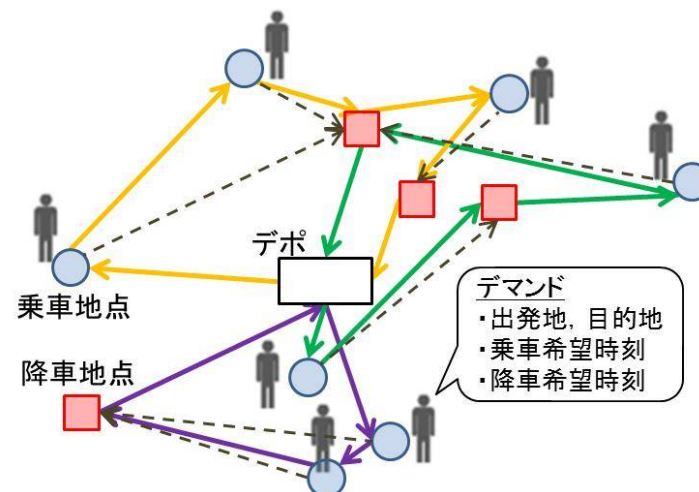
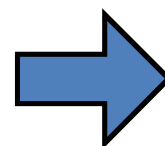


図2: 問題の概要

バスが利用者の降車希望時刻までに目的地に到着できない



不満発生

<問題の目的>

利用者の不満が最小になるバスの運行計画の決定

2.2 不満度の設定

利用者*i*の降車場所にバスが到着する際、

➤ 降車希望時刻 b_i に間に合う場合 ➡ 利用者*i*は満足する

➤ 降車希望時刻 b_i より遅れる場合 ➡ 利用者*i*に不満が生じる



利用者*i*の不満度を
$$p_i(t) = \max(0, u_i - b_i) \dots (1)$$
とする.

(u_i : 利用者*i*が降車場所に到着する時刻)

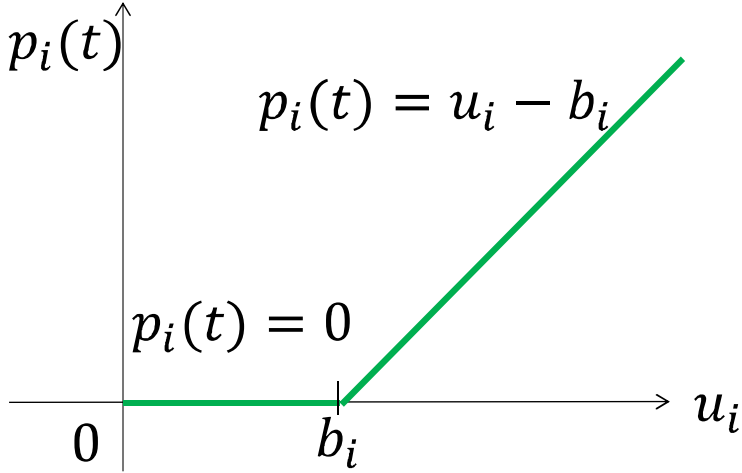


図3: 不満度関数

2. 本研究で扱う問題

2.3 前提条件

- デポ, 各地点間の移動時間は既知
- 利用者の乗車・降車する時間は考慮しない
- 追加予約は発生しない
- 途中下車, 他のバスへの乗り換えは行わない
- 乗車地点において, バスが利用者の乗車希望時刻よりも早く到着しても, 乗車希望時刻になるまで出発できない
- 降車地点において, バスが利用者の降車希望時刻よりも早く目的地に到着した場合, 即利用者を降ろし, 次の利用者の送迎を行うことができる

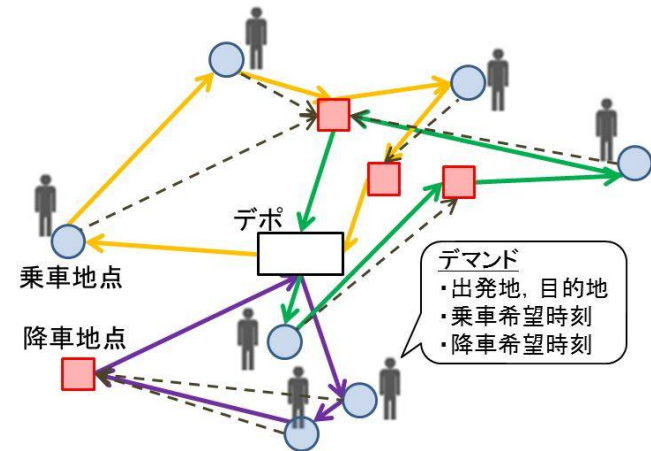


図2: 問題の概要(再掲)

2. 本研究で扱う問題

2.4 記号化(1)

$i (i = 1, 2, \dots, n)$: 利用者

$k (k = 1, 2, \dots, m)$: 送迎バス

q : バスの最大定員数

v_i : 利用者 i の乗車地点

v_{n+i} : 利用者 i の降車地点

$\{v_0\}$: デポ

$O = \{v_1, \dots, v_n\}$: 乗車地点の集合

$D = \{v_{n+1}, \dots, v_{2n}\}$: 降車地点の集合

$V = \{v_0, v_1, \dots, v_n, v_{n+1}, \dots, v_{2n}\}$:

デポ, 利用者の乗車地点・降車地点の集合

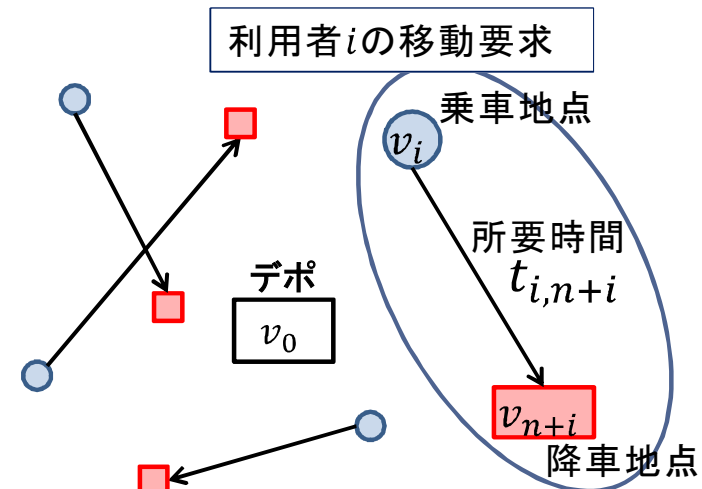


図4: 利用者 i の移動要求

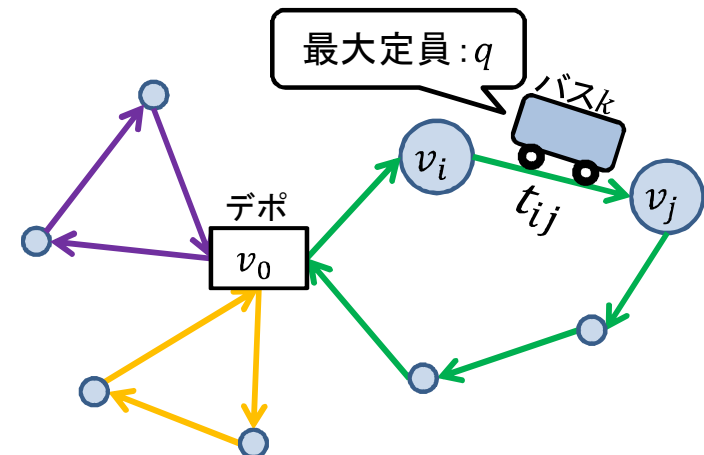


図5: 送迎の流れ

2.4 記号化(2)

t_{ij} : 地点 v_i から地点 v_j まで直行するときの走行時間

a_i : 利用者 i の乗車希望時刻

b_i : 利用者 i の降車希望時刻

$$(a_i + t_{i,n+i} \leq b_i)$$

u_i : 利用者 i が降車希望地点に到着する時刻

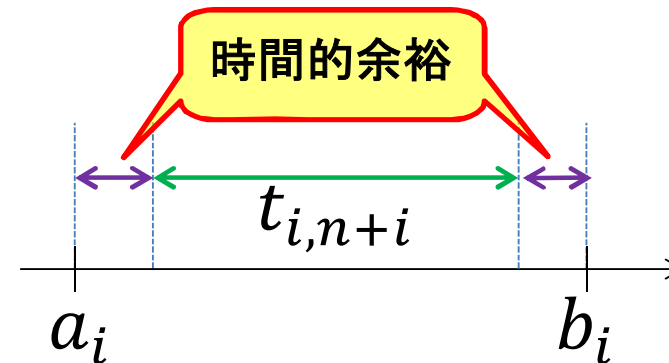


図6: 利用者 i の予約時間

2.5 目的・制約条件

<目的> 総不満度を最小化

$$\min \sum_{i=1}^n \max(0, u_i - b_i) \quad \dots (2)$$

<制約条件>

- 各バスはデポから出発しデポに戻る
- 各利用者はいずれかのバスに乗車して、乗車地点から降車地点まで送迎してもらう
- 一度にバスに乗車している人数は q 人以下

3.提案解法

3.1 提案解法の概要

- 初期解を構築する
- 初期解(全バスの訪問順序)を,「乗客の交換」,「乗客の移動」で改善する(局所探索法[4])

<探索方法>

best move

乗客の交換(移動)によって得られる全ての交換(移動)近傍の中で最も良い解に遷移する方法

first move

乗客の交換(移動)により得られる全ての交換(移動)近傍を調べるのではなく,より良い解が見つかり次第すぐにその解に遷移する方法

交換, 移動する乗客をランダムに選択して改善操作を行った.

3.2 初期解の構築(1)

次に示す構築法を用いて, 各利用者のバスへの割り当てと全バスの訪問順序を決定する.

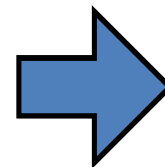
Step 1: 利用者の番号付けを, 乗車希望時刻の早い順(同値の場合, 降車希望時刻の早い順)に行う.

Step 2: 利用者1をバス1に割り当てる.

バス1の訪問順序を

0	v_1	v_{n+1}	0
---	-------	-----------	---

に更新



更新後の
各車両の訪問順序

バス1:	0	v_1	v_{n+1}	0
バス2:	0			
バス3:	0			
⋮	⋮			
バスk:	0			

3. 提案解法

3.2 初期解の構築(2)

Step 3:

利用者2~利用者 n を順番にバスに割り当てる.

既に乗客が割り当てられているバスの経路に、利用者 i の乗車/降車地点を挿入する全組み合わせを調べる.

制約: 最大定員 q 以下

総不満度の
最小値

0より大きい

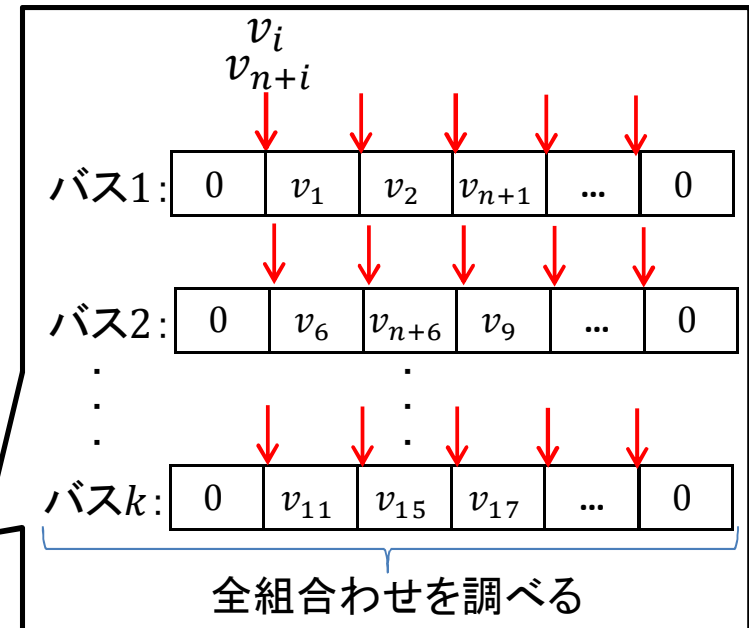
0

使用して
いるバスが
 m 台未満

m 台使用

未使用のバスに利用者 i
を割り当てる

総不満度が最小となる箇所
に乗車/降車地点をバスの
経路に挿入.



Step 4: 全利用者が割り当てられるまでStep3を繰り返す.

3.3 解の改善方法（乗客の交換）

Step 1 :

別々のバスに割り当てられている利用者の乗車/降車地点を経路から取り除く。

Step 2 :

総不満度が最小となるように、割り当てるバスをそれぞれ変えて乗車/降車地点を経路に挿入する。

Step 3 :

総不満度が小さくなる、または総不満度が同値であっても総移動距離が短くなれば解を更新する。

Step 4 :

Step1~3を更新できなくなるまで繰り返し、Step5へ移る。

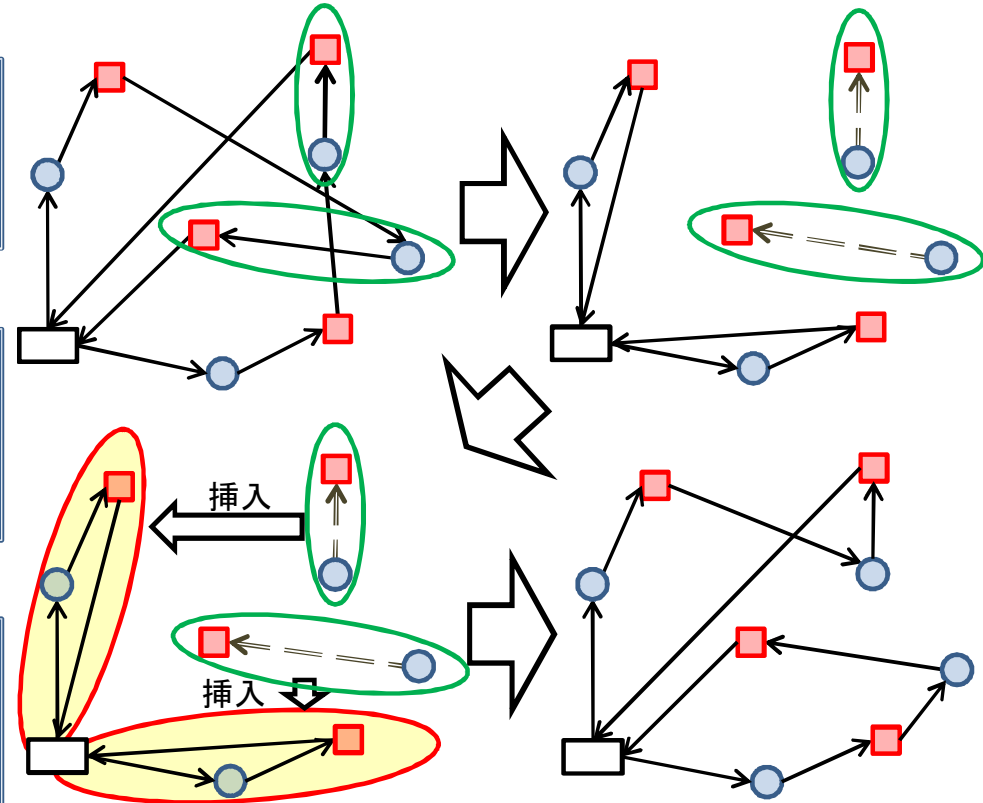


図7: 交換法実行例

3.4 解の改善方法(乗客の移動)

Step 5 :

利用者1人の乗車/降車地点を経路から取り除く.

Step 6 :

総不満度が最小となるように, 別のバスの経路に利用者の乗車/降車地点を挿入する.

Step 7 :

総不満度が小さくなる, または総不満度が同値であっても総移動距離が短くなれば解を更新する.

Step 8 :

Step5~7を更新できなくなるまで繰り返す.

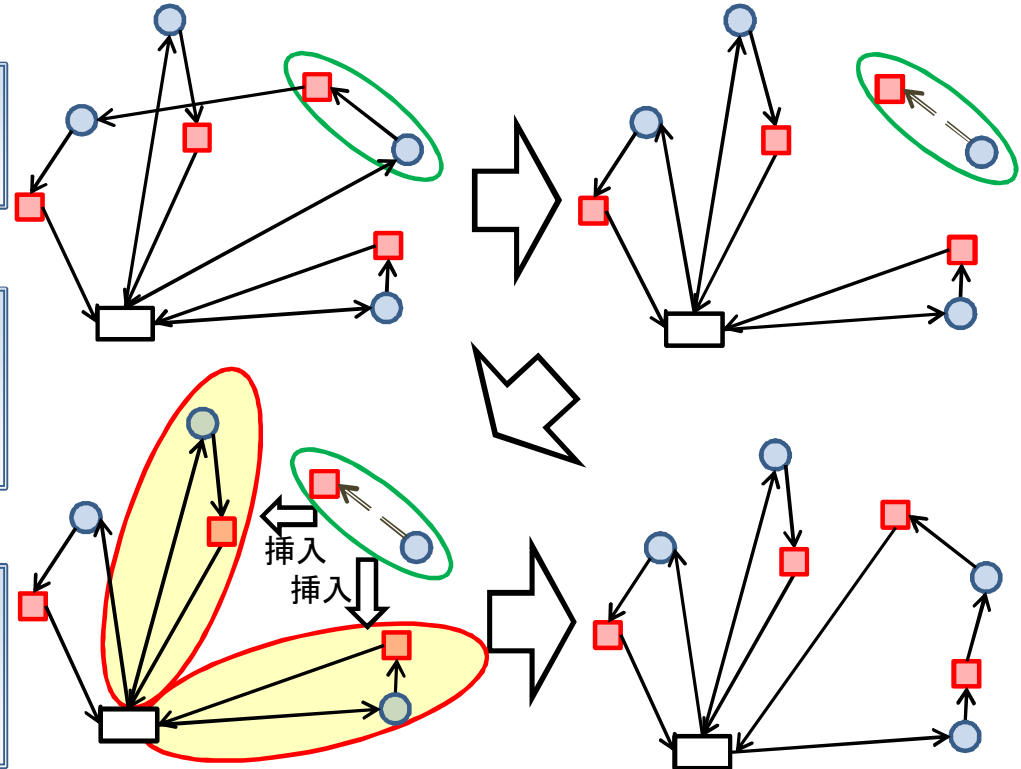


図8: 挿入法実行例

Step1~8までの操作を解の改善が見られなくなるまで繰り返す.

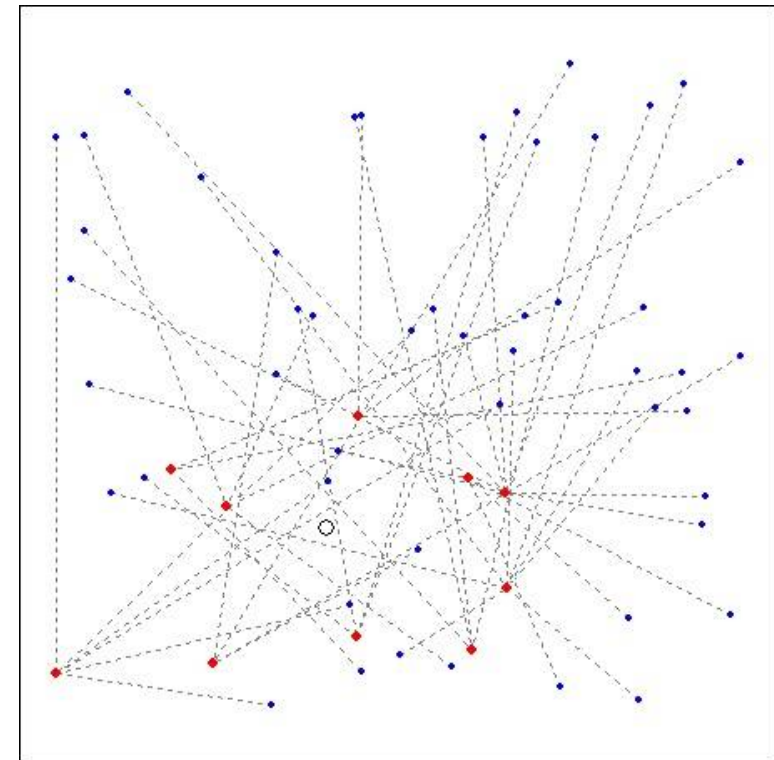
4. 数値実験

4.1 概要

- 5km × 5kmの正方形の領域に, デポ, 乗車/降車地点をランダムに発生.
- 降車地点数: 10箇所
- 乗車地点数: $n = 50, 100, 150, 200$
- 乗車希望時刻: 8:00 ~ 12:00の間で発生.
- 降車希望時刻:
乗車希望時刻に, 乗車地点から降車地点への所要時間と10~15分の時間的余裕を加えた時刻
- バスの運行速度: 時速20km
- 乗車定員数: $q = 10$

バスの台数 m を変化させた

first moveの結果: 初期解を100回変えて, 改善を行って得られた解の
平均総不満度, 平均最大不満度, 平均総移動距離



○ デポ ● 乗車地点 ● 降車地点

図9: デポ, 乗車/降車地点の発生
($n = 50$ の場合)

4.2 総不満度の比較

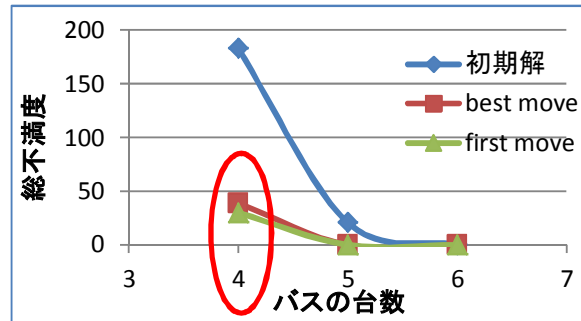


図10: $n = 100$

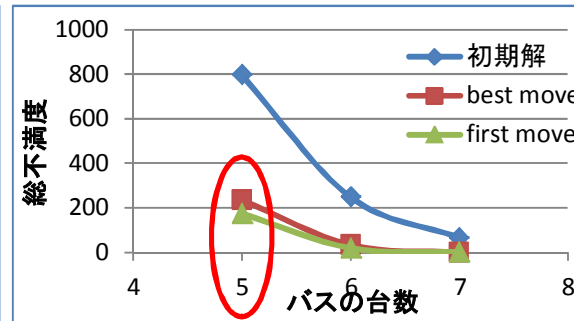


図11: $n = 150$

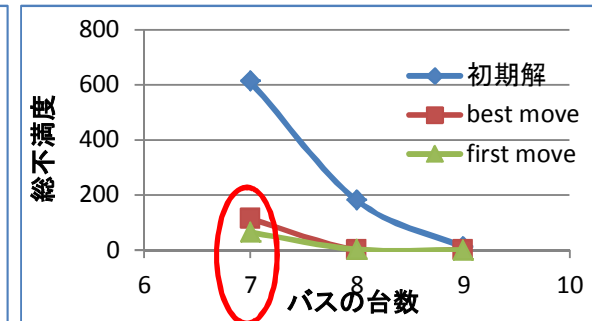


図12: $n = 200$

バスの台数を減らすと
総不満度が増加

first moveの総不満度の方が小さい

この結果以外では, first moveの
総移動距離の方が大きい

表1: first moveの総不満度の方が
小さいときの総移動距離の比較

		best move		first move	
		総不満度	総移動距離	総不満度	総移動距離
n=100	m=4	39	291757	30.1	284458
n=150	m=6	33	399587	19.21	404119
	m=5	235	371626	174.87	375714
n=200	m=7	114	505319	65.01	511171

総移動距離を短くすることで, 早く目的地に
到着し不満度を少なくすることができると考
えていたが, そのような関係性はない.

4.3 計算時間の比較

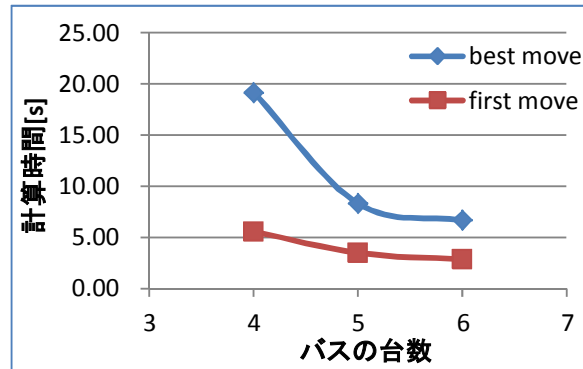


図13: $n = 100$

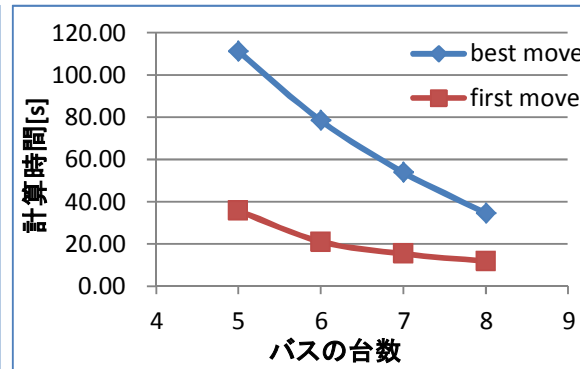


図14: $n = 150$

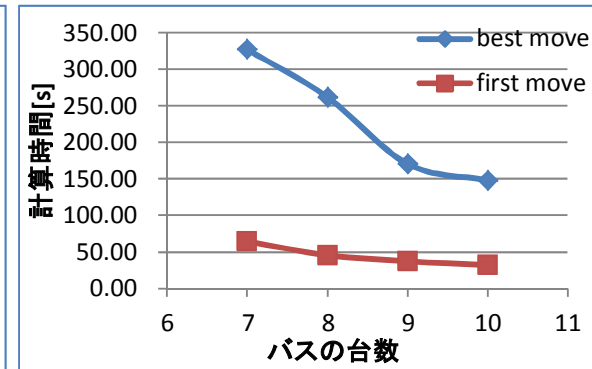
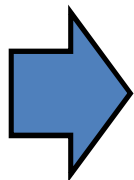


図15: $n = 200$

first moveの計算時間の方が短い



first move方が計算時間が短く、精度の良い解が得られた

4.4 バスの台数に関する考察

$(n, m) =$

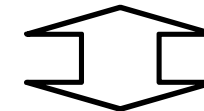
$(100, 5), (150, 6), (200, 8)$

- 平均総不満度が小さい
- 最大不満度がそれぞれ1分未満, 約5分, 2分と小さい

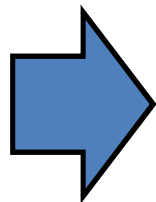


バスの台数を1台減らす

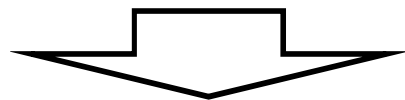
- 利用者1人あたりの平均不満度が約1分と小さい



- 不満度(遅延時間)の最大値が約21分と大きくなる.



日常生活において、約束の時間に対して約20分もの遅れは信頼関係を損ねる危険性あり.



利用者が100人, 150人, 200人の場合はそれぞれ5台, 6台, 8台が適切

表2: バスの台数と
総不満度, 最大不満度

		総不満度	最大不満度
n=100	m=5	0.12	0.1
	m=4	30.1	9.1
n=150	m=6	19.21	5.43
	m=5	174.87	21.06
n=200	m=8	2.78	1.74
	m=7	65.01	10.07

5. まとめと今後の課題

<まとめ>

- デマンドバスが利用者の降車希望時刻よりも遅れて到着することを「利用者の不満度」として考え、利用者の総不満度を最小にするデマンドバスの運行計画問題を提起し、その解法を提案した.
- バスの巡回経路長を短くすることが必ずしも不満度の小さい運行計画にならないことがわかった.

<今後の課題>

- 提案した解法は局所最適解に陥りやすいため、改善法の探索方法を工夫し、解の精度を上げる.

参考文献

- [1]: 鈴木文彦 (2012), 地方におけるオンデマンドバス交通の可能性と課題, オペレーションズ・リサーチ 経営の科学 Vol.57, no.3, pp.124-129.
- [2]: 全国デマンドバス交通システム導入機関連絡協議会,
<http://www.demand-kyougikai.jp/index.html> (2012/8/26).
- [3]: 大和裕幸, 稗方和夫 (2006), オンデマンドバスの実用化研究,
<http://www.nakl.t.u-tokyo.ac.jp/odt/dl/paper/CREST.pdf> (2012/6/22).
- [4]: 柳浦睦憲, 茨木俊秀 (2001), 組合せ最適化-メタ戦略を中心として-, 朝倉書店, 237pp.

抄録訂正

抄録に誤りがありましたので、訂正致します。

抄録p.80 4.数値実験 表1 $n=150, m=6$ の提案解法 first moveの結果

誤 総移動距離 174.87

正 総移動距離 404119

ご清聴ありがとうございました

付録

付録：局所探索法

- 現在の解に少し変形を加えて、解の改善を行う。
- 解の改善が可能な限り続ける。

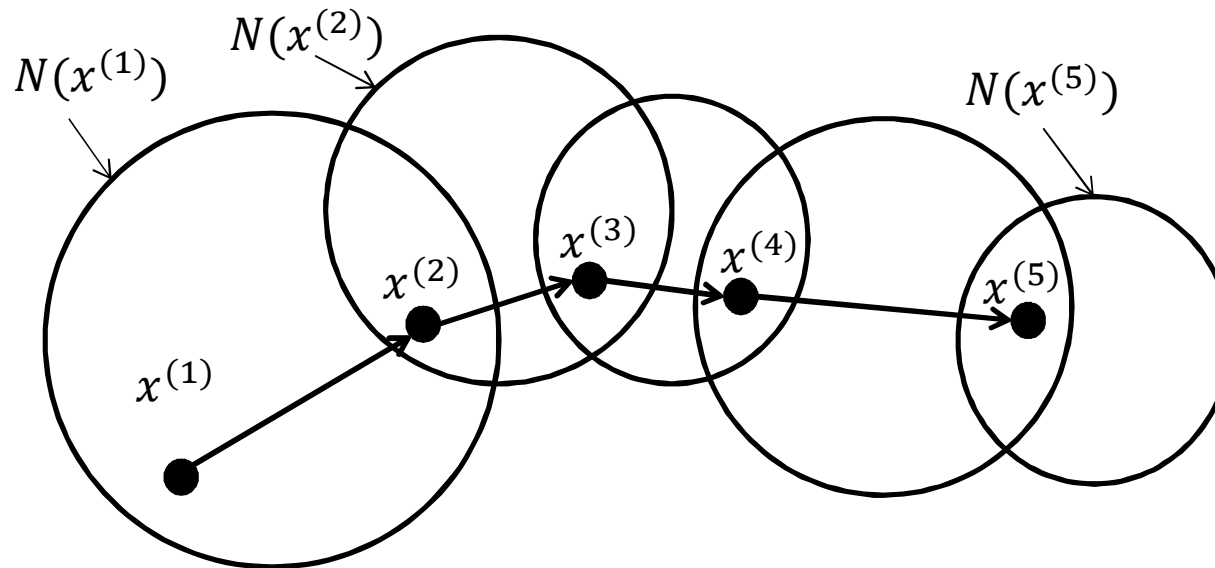


図16: 局所探索進行の様子

F : 実行可能領域 $x \in F$: 実行可能解 $x^{(k)}$: k 番目の解
 $N(x) \subset F$: x に少しの変形を加えることによって得られる解の集合

付録: best move

乗客の交換(移動)によって得られる全ての交換(移動)近傍の中で, 最も良い解に遷移する方法.

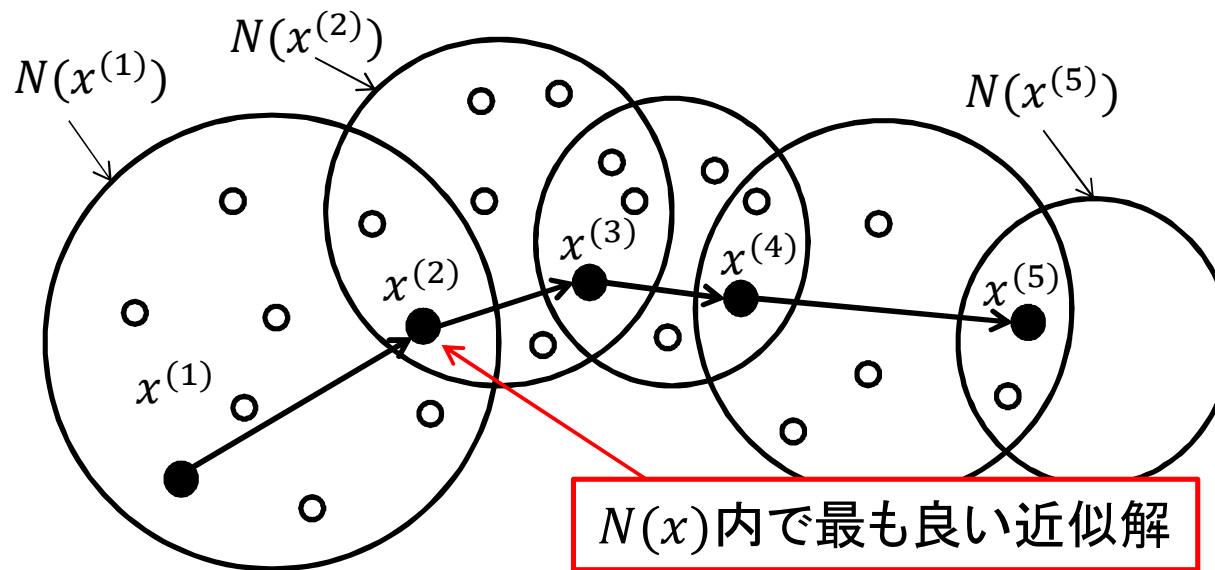


図17: best move進行の様子

F : 実行可能領域 $x \in F$: 実行可能解 $x^{(k)}$: k 番目の解
 $N(x) \subset F$: x に少しの変形を加えることによって得られる解の集合

付録: first move

乗客の交換(移動)により得られる全ての交換(移動)近傍を調べるのではなく、より良い解が見つかり次第すぐにその解に遷移する方法

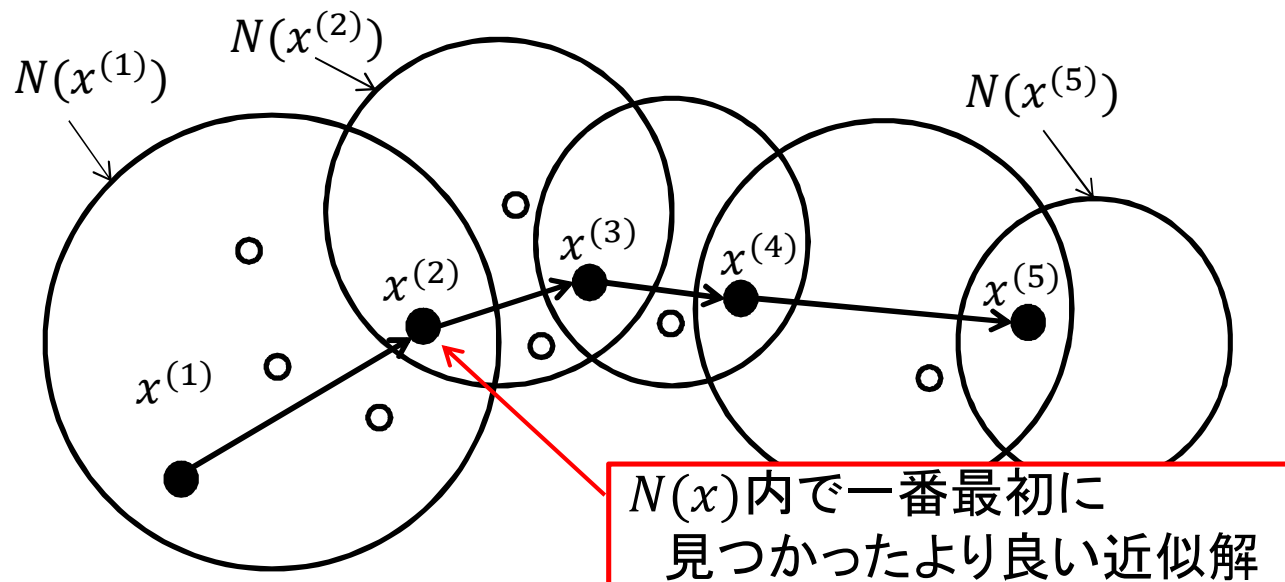


図18: first move進行の様子

F : 実行可能領域 $x \in F$: 実行可能解 $x^{(k)}$: k 番目の解
 $N(x) \subset F$: x に少しの変形を加えることによって得られる解の集合

付録: デポ, 乗車/降車地点の発生

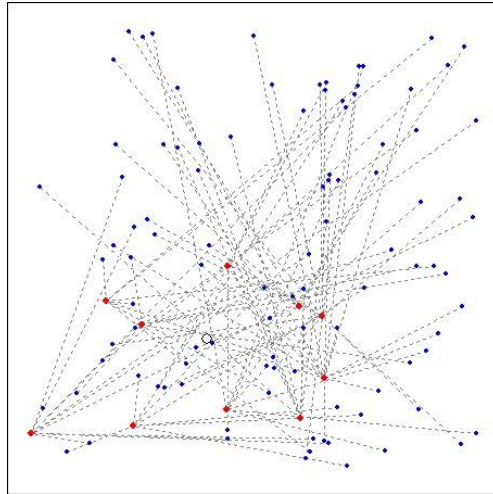


図19: $n = 100$ の場合

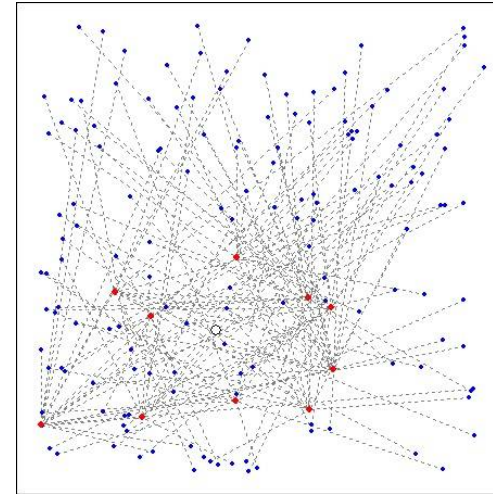


図20: $n = 150$ の場合

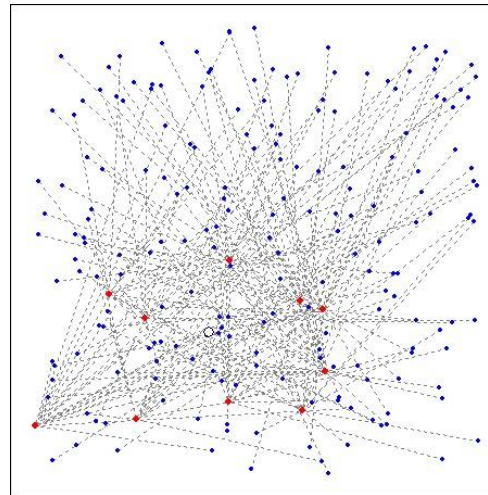


図21: $n = 200$ の場合

付録：数値実験結果(1)

表3: 実験結果(総不満度, 最大不満度, 総移動距離)

		初期解			提案解法					
					best move			first move		
		総不満度	最大不満度	総移動距離	総不満度	最大不満度	総移動距離	総不満度	最大不満度	総移動距離
n=50	m=4	0	0	202334	0	0	175776	0	0	179848
	m=3	7	5	205949	0	0	180180	0	0	184180
	m=2	399	33	172446	173	27	158681	154.14	15.62	161907
n=100	m=6	0	0	309705	0	0	272531	0	0	277343
	m=5	21	5	314264	0	0	287508	0.12	0.1	284790
	m=4	183	18	297620	39	10	291757	30.1	9.1	284458
n=150	m=8	0	0	449324	0	0	395975	0	0	400798
	m=7	66	12	486998	0	0	421214	0.72	0.62	411475
	m=6	248	28	458886	33	6	399587	19.21	5.43	404119
	m=5	797	39	396138	235	30	371626	174.87	21.06	375714
n=200	m=10	0	0	580731	0	0	493979	0	0	498026
	m=9	11	6	607969	0	0	506567	0	0	508273
	m=8	181	13	581436	1	1	526573	2.78	1.74	518769
	m=7	613	19	545284	114	18	505319	65.01	10.07	511171

付録：数値実験結果(2)

表4：実験結果(計算時間)

		best move	first move
n=50	m=4	0.43	0.27
	m=3	0.46	0.27
	m=2	1.79	0.75
n=100	m=6	6.66	2.83
	m=5	8.30	3.48
	m=4	19.15	5.57
n=150	m=8	34.64	11.75
	m=7	53.84	15.25
	m=6	78.39	20.99
	m=5	111.26	35.60
n=200	m=10	147.60	32.36
	m=9	170.77	37.11
	m=8	261.67	45.66
	m=7	327.40	64.51

付録：数値実験 結果 ($n = 50$)

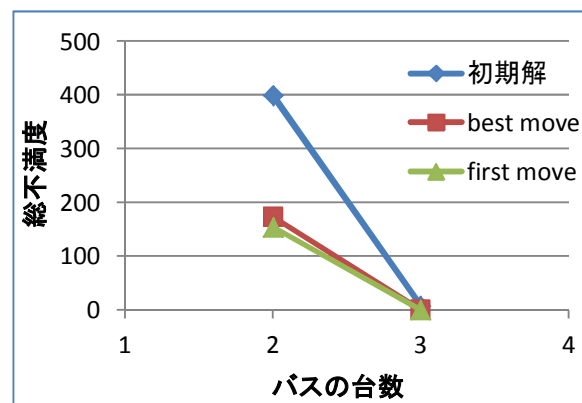


図22: 総不満度の比較

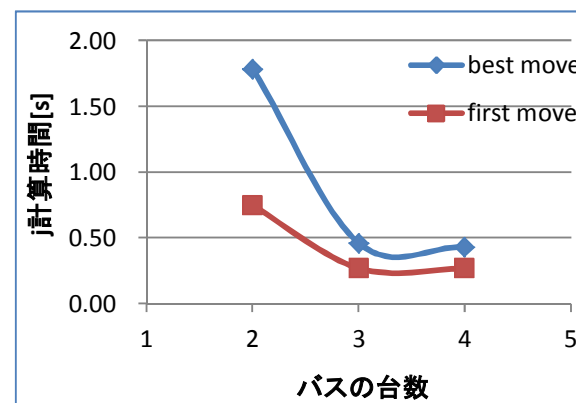


図23: 計算時間の比較