集配作業を伴う
巡回セールスマン問題に対する
解法の研究

沼田研究室

4408035

小西　昌子

目次

1.1 はじめに

◆現在，繰り返し物を運ぶという作業は至る所で行われている.

◆その中でも,「物を配送しながら回収する」という巡回作業に着目.

(例)　ラーメンの配達

配送物:ラーメン / 回収物:空の容器

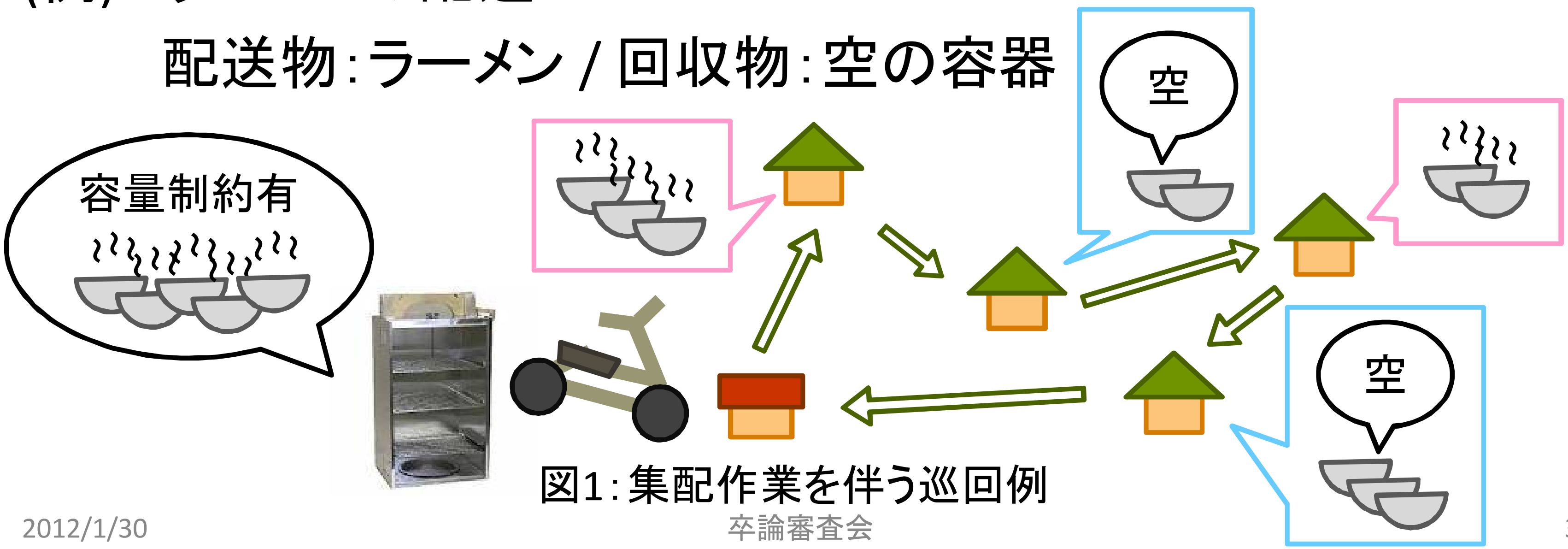


図1:集配作業を伴う巡回例

1.2 TSPPDについて

◆ 集配作業を伴う巡回セールスマン問題

　TSPPD (Travelling Salesman Problem with Pick-up and Delivery)

容量制約のある配送車が集配作業を同時に行いながら，全ての集配地点を1度ずつ訪問し，再び出発地点に戻ってくる時の最短経路を求める問題

1.3 発見的解法の必要性

TSPPDでは，集配地点数が大きくなると厳密解を求めることは困難 [1]

その為

現実的な計算時間の中で比較的良い精度の準最適解を与えてくれる発見的解法が必要

TSPPDに対する既存の発見的解法例

- Pick-up and Delivery along Optimal Tour (PDOT) [2]
- The Cheapest Feasible Insertion (CFI) [2]

1.4 研究目的

既存の発見的解法と比べ，より短い巡回路を得ることができる新しい解法を提案し，その性能評価を行う.

集配作業を同時に伴う巡回活動の更なる効率化を図る.

2.1 問題設定 (1)

◆1台の配送車 (積載容量 : q)

◆出発点 (デポ) :

◆回収点 : (回収量 : p_i)

◆配送点 : (配送量 : d_i)

◆各点間の距離 : $c(i,\ j)$

- $q = \sum_{i=1}^{n} p_i = \sum_{i=1}^{n} d_i$ であると仮定.

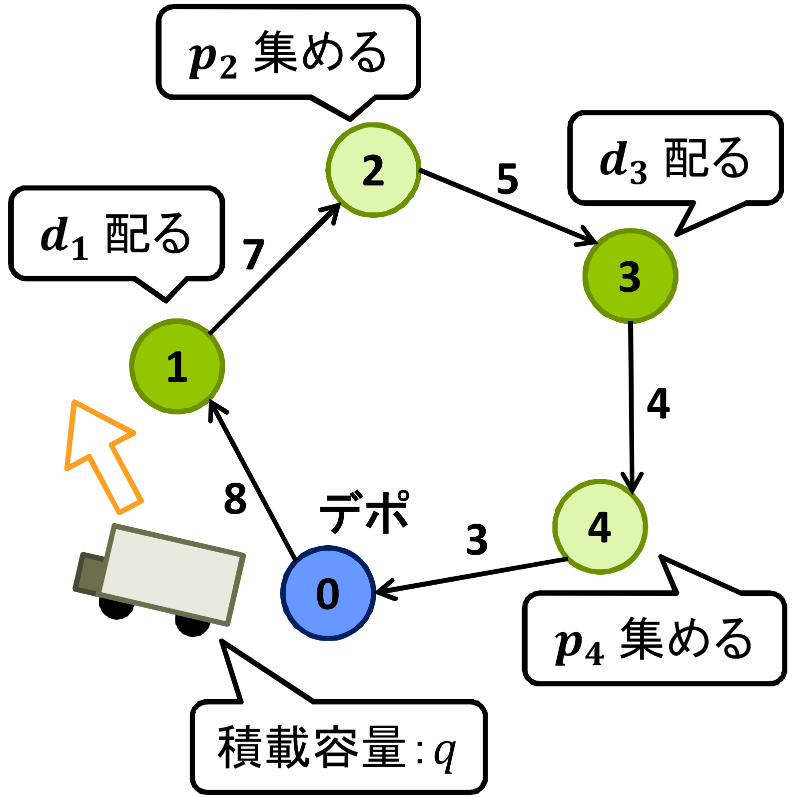


図2 : TSPPDの設定

2.2 問題設定 (2)

1. 配送車は全ての配送物を積んだ状態でデポを出発し,全ての回収物を積んだ状態でデポに戻る
2. 配送車は積載容量を超えた状態で移動することは出来ない

$$\boxed{\text{積載している回収物}} + \boxed{\text{積載している配送物}} \leq q$$

以上の設定のもとで最短な巡回路を求める

2.3 積載容量と巡回路長

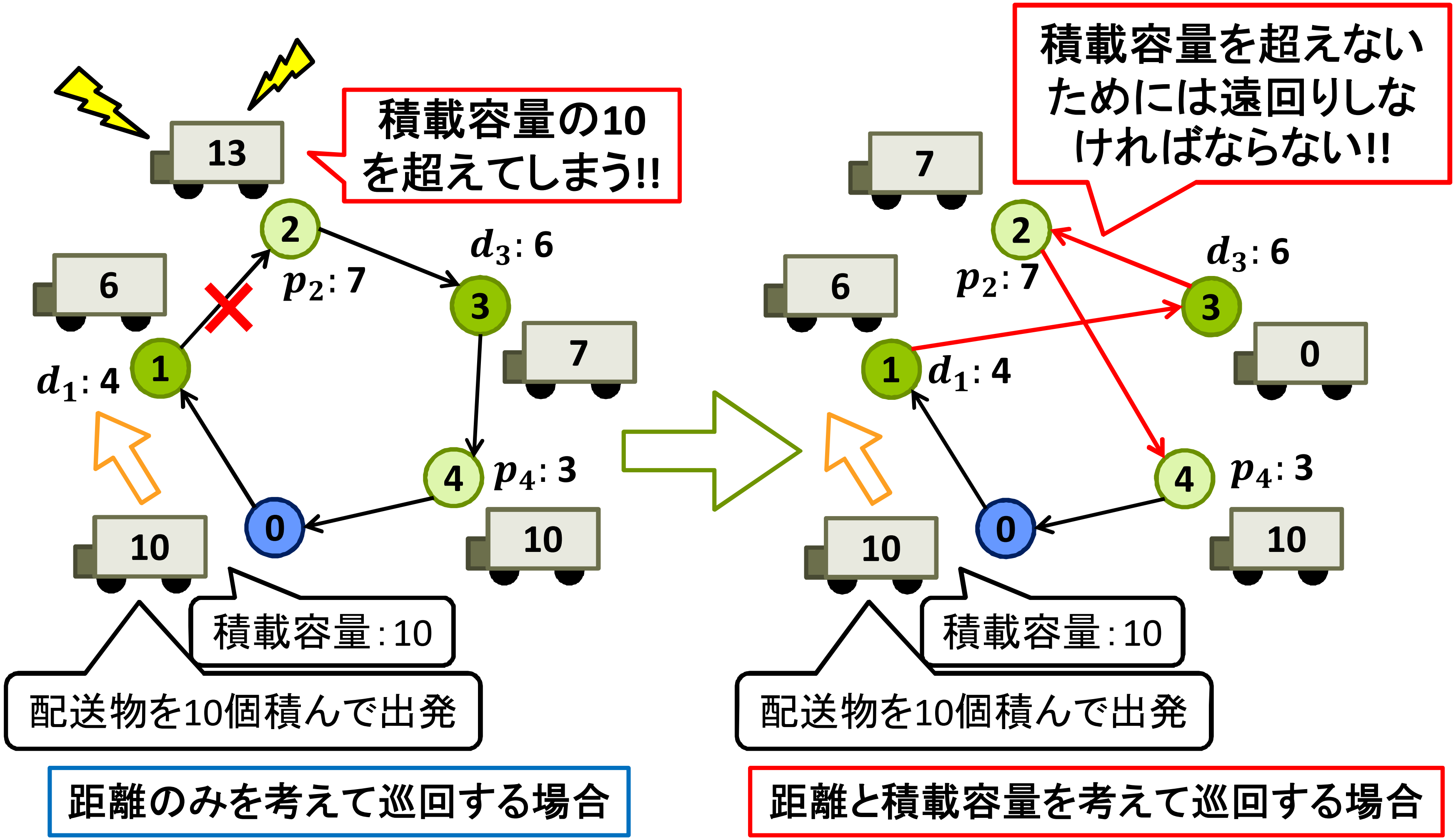


図3：積載容量と巡回路長

3.1 既存の発見的解法

◆PDOT [2] (Pick-up and Delivery along Optimal Tour)

Step1:デポを除く全ての点を通る最短巡回路を求める.

Step2: 配送車の積載量が最大である点の次にデポを挿入する (容量制約を満たす).

→ 巡回路を決定

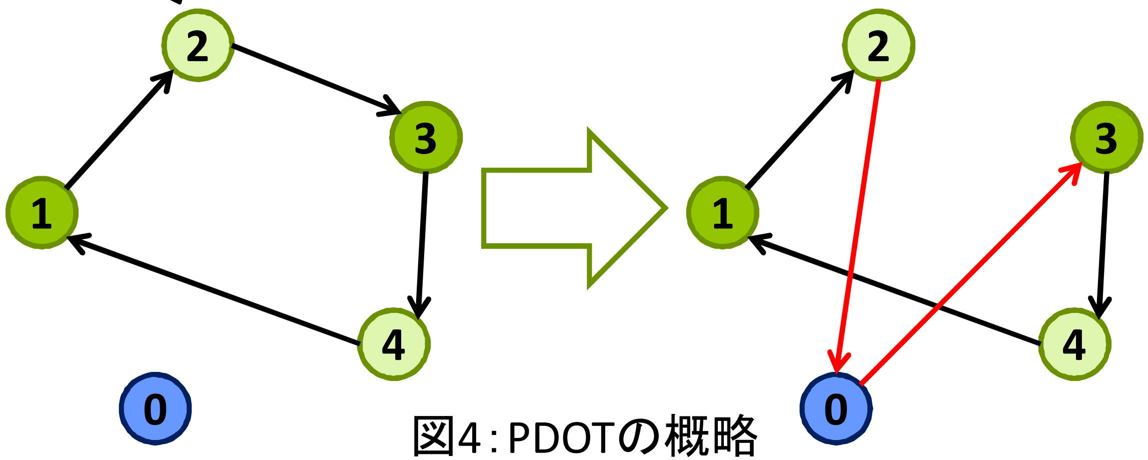


図4 : PDOTの概略

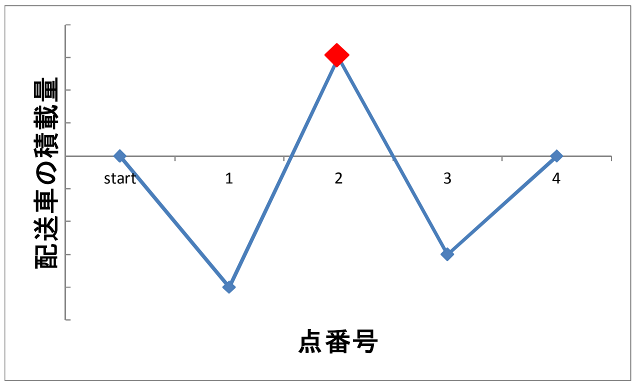


図5 : 配送車の積載量の推移

3.2 予備実験：厳密解とPDOTの比較

◆ 問題例：bays29, eil51, st70, eil76 [4]
◆ デポの位置：原点(0, 0), 全点の中心

初期巡回路が最短でも誤差率は高くなる

デポが中心にある方が誤差率は低い

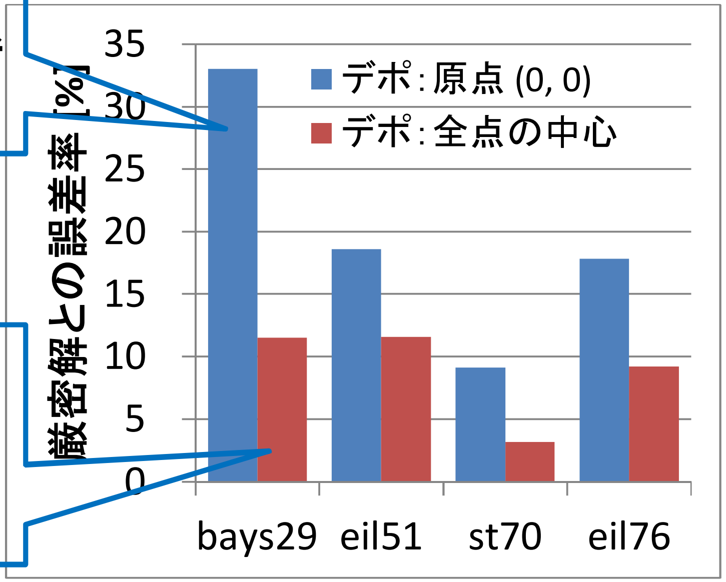


図6：PDOTにおける厳密解の誤差率 [%]

：デポ
：配送点
：回収点

積載量最大点の位置がデポから遠くなると得られる巡回路は長くなる

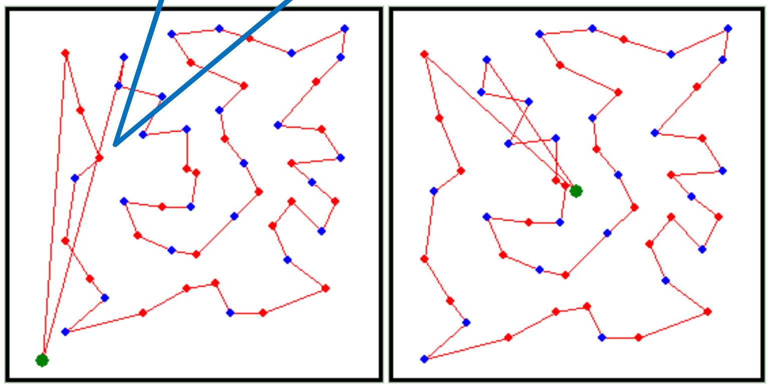


図7：eil51の実行例

4.1 PDOTの問題点と提案解法

1. 積載量最大点の位置とデポが遠くなる
→デポを含めて初期巡回路を構築

2. 初期巡回路が最短でも良い巡回路は得られない
→「制約違反量」という新たな指標を導入

3-(1) 最短巡回路を求めることは困難
3-(2) なるべくデポと積載量最大点の位置を近くしたい
→発見的に多数の準最短巡回路を求める

4.2 制約違反量

◆制約違反量

巡回路内で積載容量を超えてしまった点での超過量

例：図8の点2での制約違反量は3

◆制約違反量の総和

巡回路内において積載容量を超えている全ての点での超過量の総和

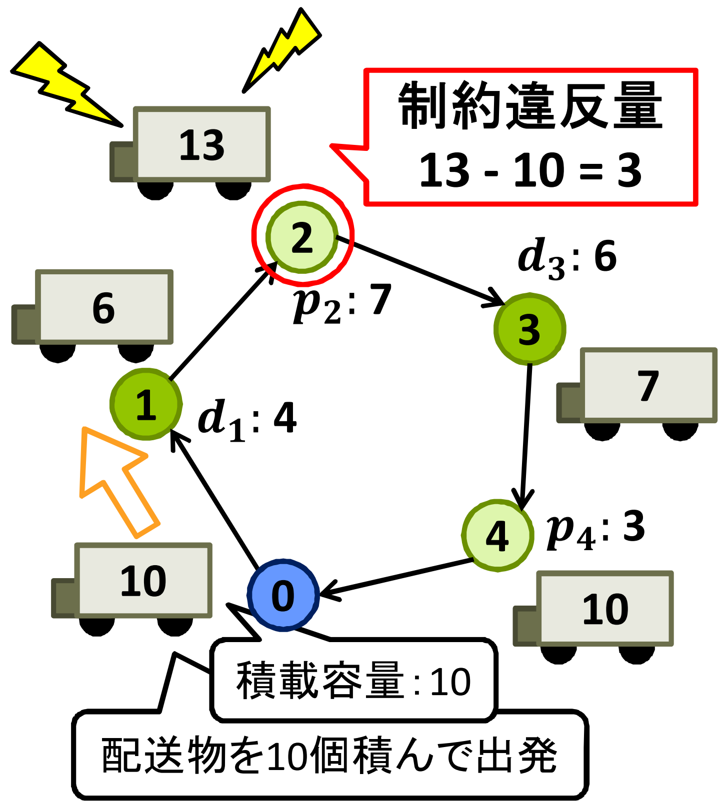


図8：制約違反量の例

4.3 提案解法1

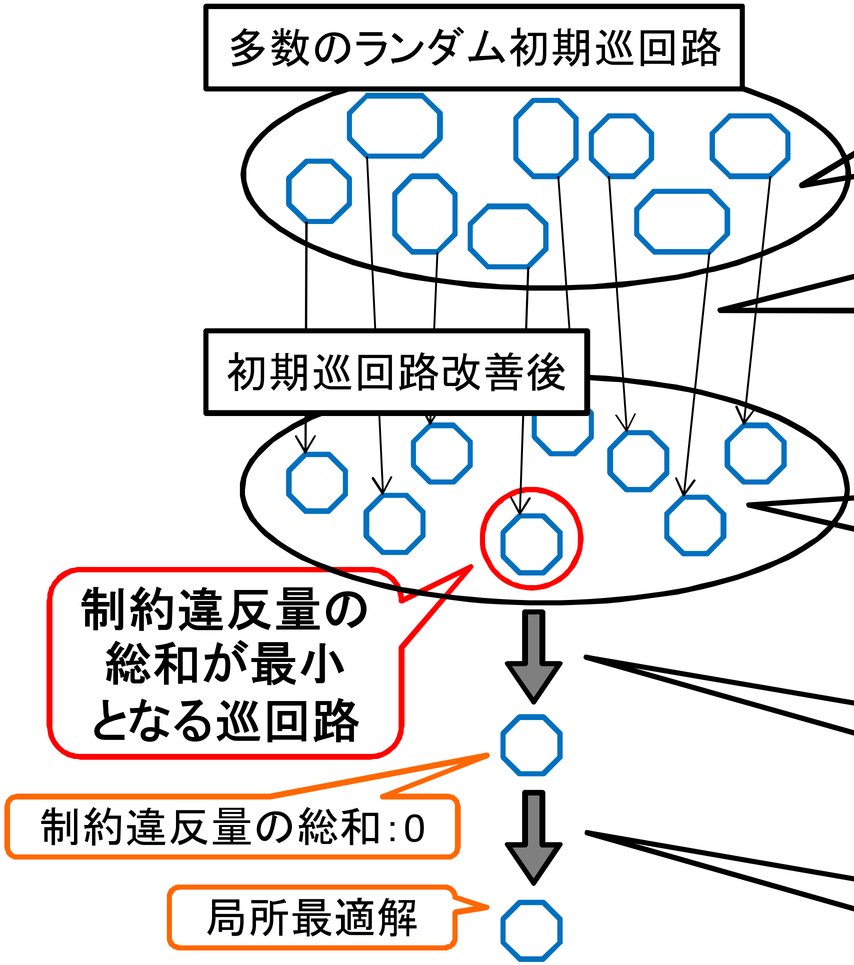


図9：提案解法1の流れ

Step1：**初期巡回路の生成**
デポを含む全ての点を通る多数のランダムな初期巡回路を生成

Step2：**初期巡回路改善**
2-opt法[※]を用いて巡回路長を改善

Step3：**制約違反量の総和が最小となる巡回路の選択**
次探索の初期巡回路を決定

Step4：**局所探索法Ⅰの実行**
制約違反量の総和を0にする

Step5：**局所探索法Ⅱの実行**
巡回路長を改善する

※2-opt法：付録参照

4.4 提案解法2

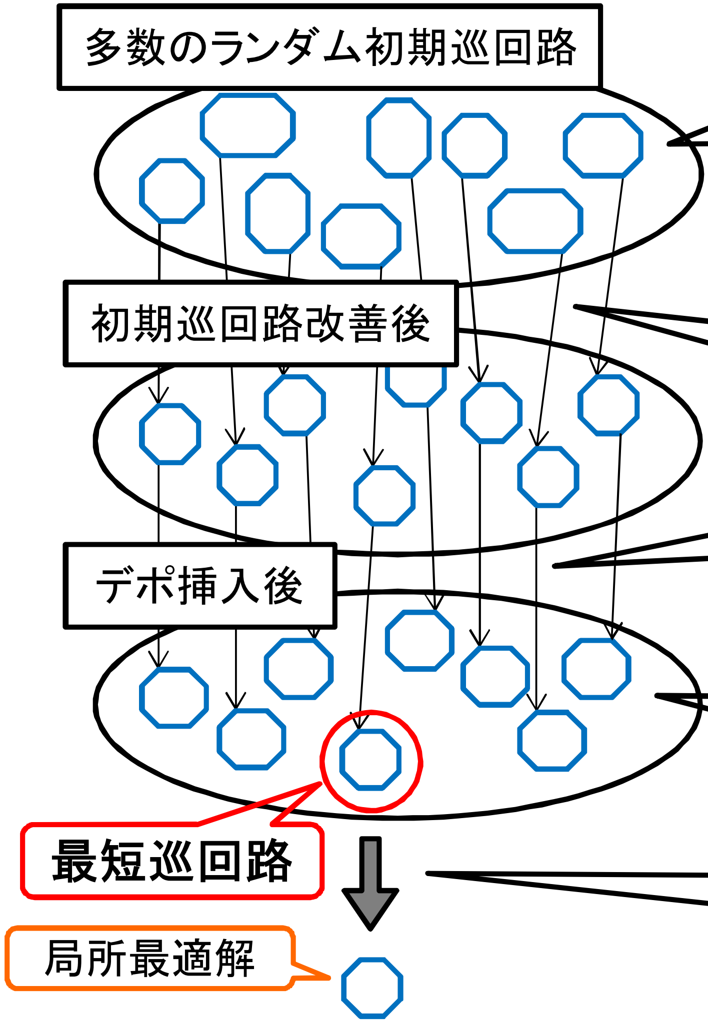


図10：提案解法2の流れ

Step1：初期巡回路の生成
デポを除く全ての点を通る多数のランダムな初期巡回路を生成

Step2：初期巡回路改善
2-opt法を用いて巡回路長を改善

Step3：デポの挿入
積載量最大点の次にデポを挿入

Step4：最短巡回路の選択
次探索の初期巡回路を決定

Step5：局所探索法Ⅱの実行
巡回路長を改善する

4.5 制約違反量を改善する手法

◆局所探索法Ⅰ：制約違反量の総和が0になるように巡回路を改善する手法

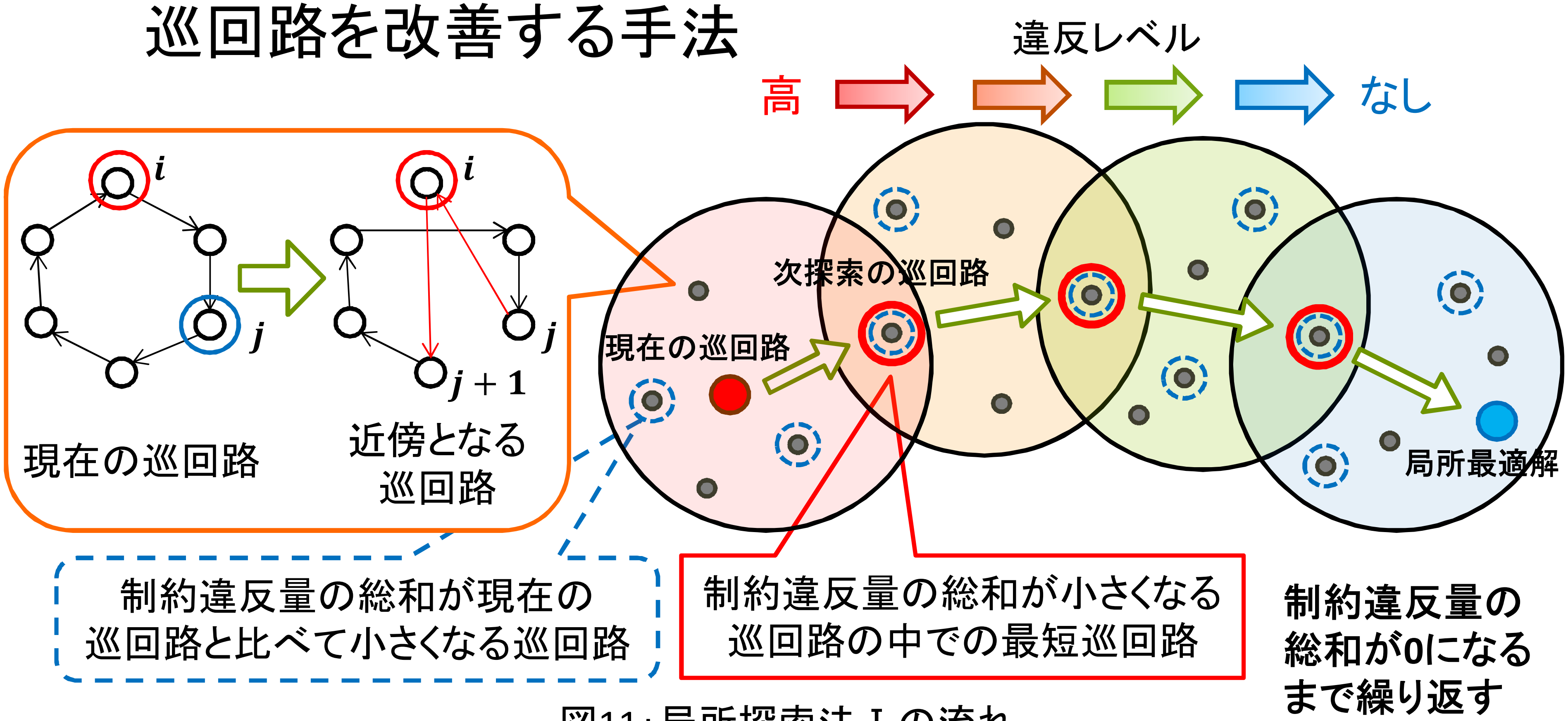


図11：局所探索法Ⅰの流れ

4.6 巡回路長を改善する手法

◆局所探索法Ⅱ：容量制約を満たしながら巡回路長の改善を行う手法

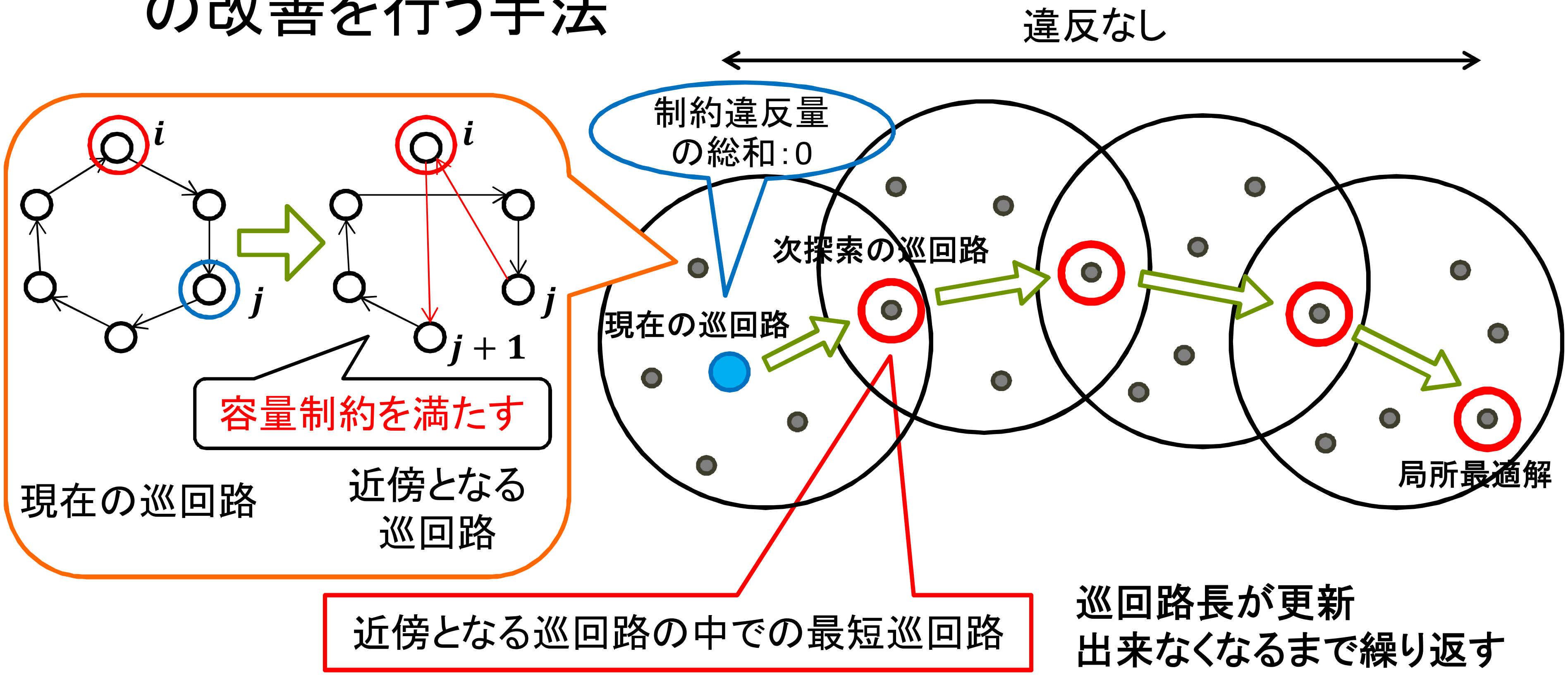


図12：局所探索法Ⅱの流れ

5.1 数値実験

◆提案解法のプログラムを実装した.

[問題例]:bays29, eil51, eil76, kroA100, kroA200, lin318, pa561, rat783 [4]

[初期巡回路の個数]:提案解法1, 2ともに50個

[提案解法の試行回数]:各問題例につき50回

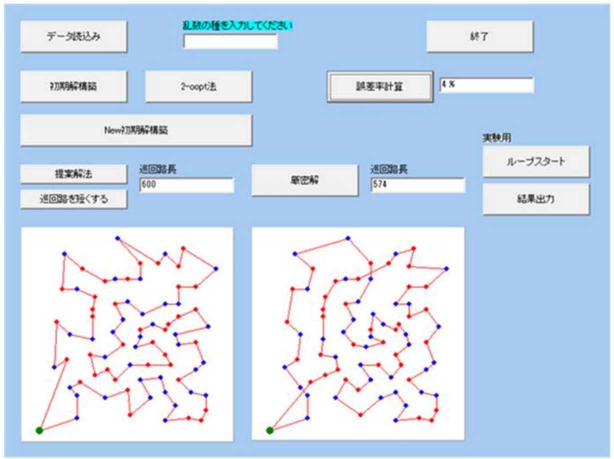


図13:プログラム実行画面 [提案解法1]

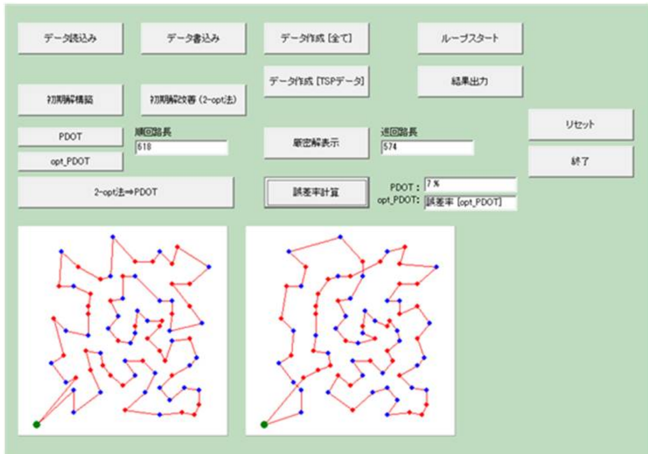


図14:プログラム実行画面 [提案解法2]

5.2 提案解法とPDOTの比較

- ◆ 提案解法の巡回路長はPDOTより短い
- ◆ 提案解法1，2は同程度の性能

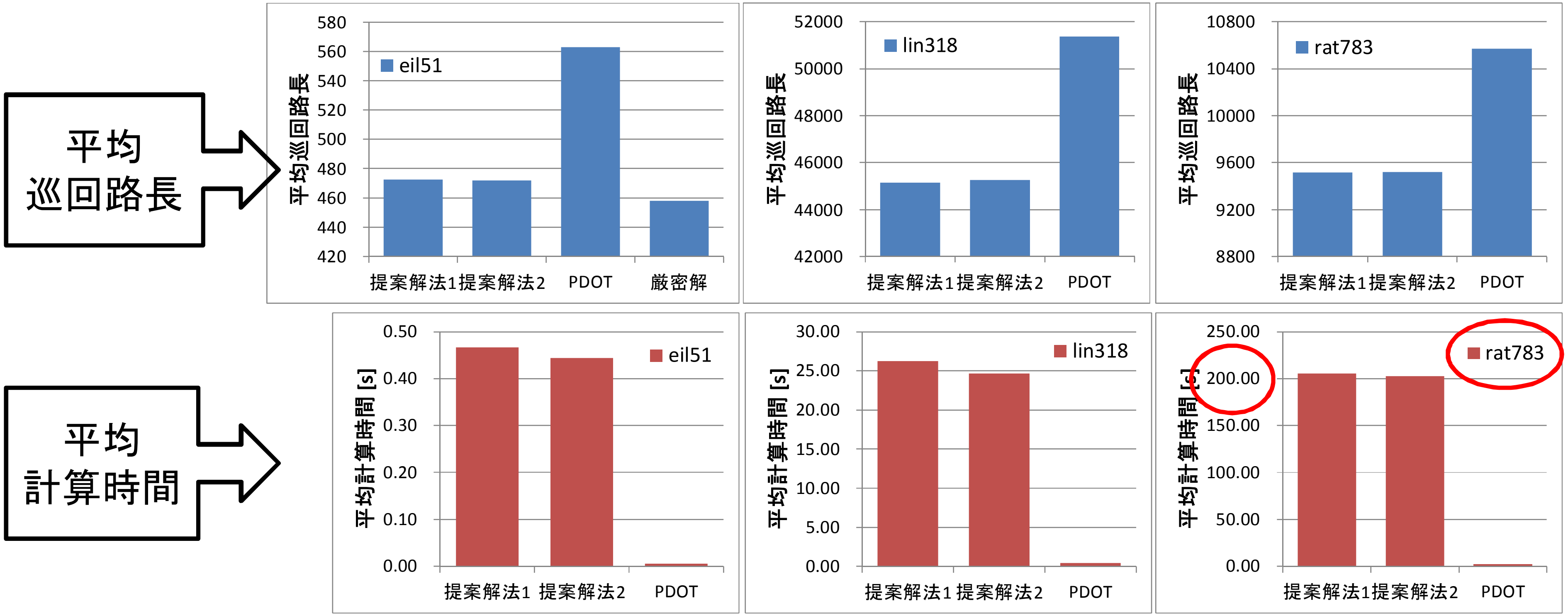


図15：eil51 (左) と lin318 (中) とrat783 (右) の実験結果

6. まとめと今後の課題

◆まとめ

1. TSPPDに対し2つの解法を提案し，その性能を検証した.
2. 提案解法は厳密解が求められない問題に対しても，現実的な時間内に既存の発見的解法よりも良い巡回路を求めることが出来た.

◆今後の課題

1. 提案解法の性能の向上
2. 複数の配送車で集配作業を行いながら巡回したり，複数種類の集配物を扱ったりする問題への取り組み

参考文献

[1] 山本芳嗣，久保幹雄 (1997)，巡回セールスマン問題への招待，初版第1刷，朝倉書店，167pp.

[2] Gur Mosheiov (1994)，The Traveling Salesman Problem with pick-up and delivery，*European Journal of Operation Research*，Vol.79，No.2，pp.299-310.

[3] gurobi，http://www.gurobi.com/ (2011.10.28).

[4] TSPLIB, http://compot.ifi.uni-heidelberg.de/software/TSPLIB95 (2011.6.27).

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.128　4.4 提案解法2

誤　Step2：各初期巡回路に対してデポを挿入する.

正　Step2：2-opt法を用いて各初期巡回路を改善し，デポを挿入する.

ご清聴ありがとう御座いました

付録

付録:記号化

- P: 回収場所の集合
- D: 配送場所の集合
- N: 全ての配送・回収点 $[D \cup P]$ $(|N| = n)$
- $c(i,\ j)$: 点 i, j 間の距離
- $x_{i\ j} = \begin{cases} 1: 点\ i\ から点\ j\ へで配送車が直接移動する \\ 0: それ以外 \end{cases}$
- $y_{i\ j}$: 点 i から点 j に移動する際に配送車が積載している回収物の総和
- $z_{i\ j}$: 点 i から点 j に移動する際に配送車が積載している配送物の総和

付録:定式化

(TSPPD)

$$\min \sum_{i=0}^{n}\sum_{j=0}^{n} c(i,\ j)x_{i\ j} \tag{1}$$

$$s.t. \sum_{i=0}^{n} x_{i\ j} = 1, \quad j = 0, \dots, n \tag{2}$$

$$\sum_{j=0}^{n} x_{i\ j} = 1, \quad i = 0, \dots, n \tag{3}$$

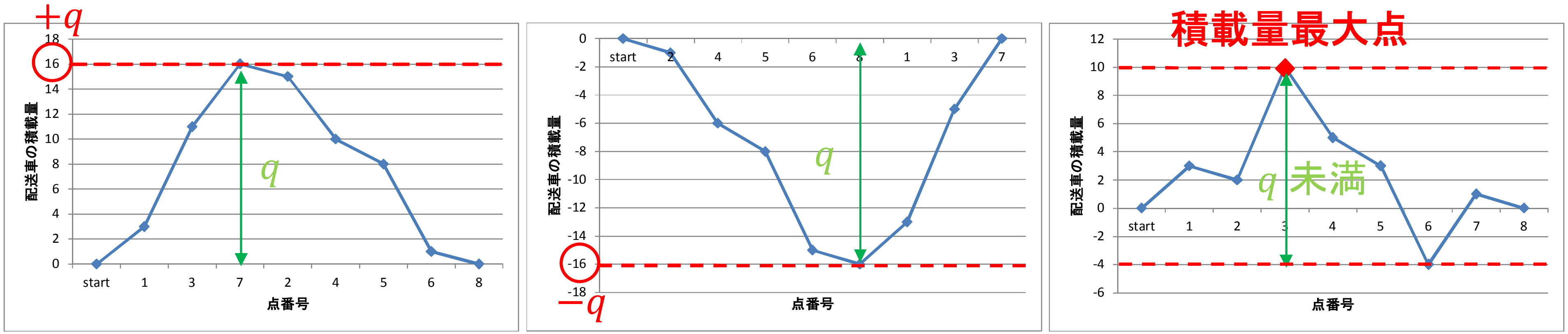
$$\sum_{j=0}^{n} y_{i\ j} - \sum_{k=0}^{n} y_{k\,i} = \begin{cases} p_i & i\ f\ i \in P \\ -q & i\ f\ i = 0 \\ 0 & i\ f\ i \in D \end{cases} \tag{4}$$

$$\sum_{j=0}^{n} z_{i\ j} - \sum_{k=0}^{n} z_{k\,i} = \begin{cases} -d_i & i\ f\ i \in D \\ q & i\ f\ i = 0 \\ 0 & i\ f\ i \in P \end{cases} \tag{5}$$

$$y_{i\ j} + z_{i\ j} \le q * x_{i\ j}, \quad i,\ j = 0, \dots, n \tag{6}$$

$$x_{i\ j} = 0,\ 1, \quad y_{i\ j} \ge 0, \quad z_{i\ j} \ge 0 \tag{7}$$

付録：PDOTの原理 (1)



(a) 全ての回収点を訪問後，全ての配送点を訪問する場合の積載量の推移

(b) 全ての配送点を訪問後，全ての回収点を訪問する場合の積載量の推移

(c) 回収作業を行いながら配送作業を行う場合の積載量の推移

図16：配送車の積載量の推移例

ある初期巡回路における最大積載量と最小積載量の差は q 以下となる

付録:PDOTの原理 (2)

積載量最大点の次にデポを挿入
→全ての集配点における積載量は 0 以下

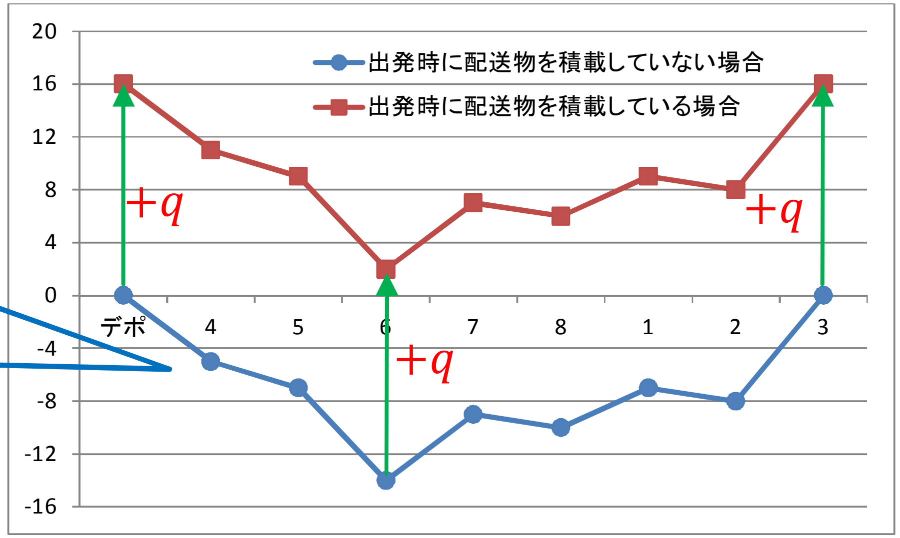


図17:デポ挿入後の積載量の推移例

最大積載量と最小積載量の差は q 以下
出発時に積載している配送物は q 個

出発時に積載している配送物を考慮
→全ての集配点における積載量は0以上 q 以下
⇒容量制約を満たす

付録：予備実験 結果

表1：PDOTの巡回路長と,厳密解との誤差率

	デポ：原点（0, 0）			デポ：全点の中心		
	巡回路長		誤差率 [%]	巡回路長		誤差率 [%]
	PDOT	厳密解		PDOT	厳密解	
bays29	13610	10230	33.04	10288	9224	11.54
eil51	543	458	18.56	483	433	11.55
st70	777	712	9.13	716	694	3.17
eil76	674	572	17.83	593	543	9.21

付録：2-opt法

◆2-opt法

1. 巡回路の中から適当な2本の枝を選択する
2. 選択した2本の枝を繋ぎかえたときに,巡回路長が短くなれば巡回路を更新する
3. 手順2を満たす枝がなくなるまで手順2を繰り返す

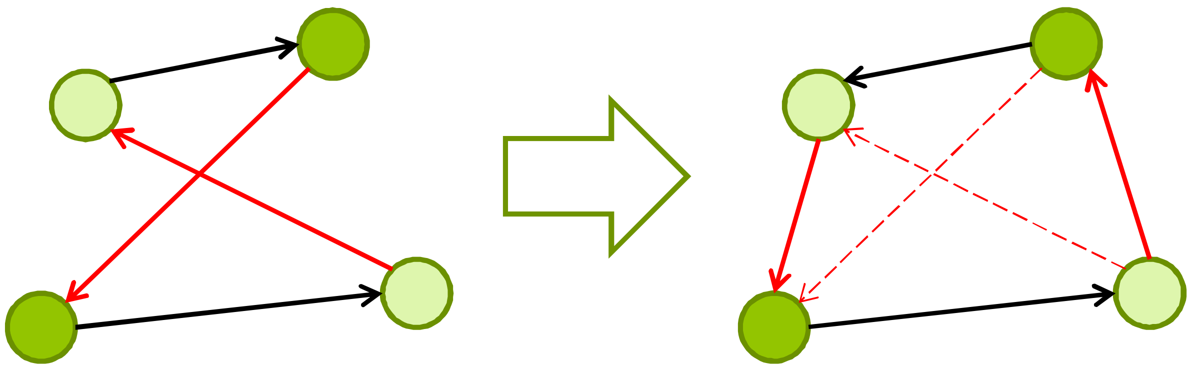


図18：2-opt法の流れ

付録：数値実験 結果

表2：提案解法とPDOTの平均巡回路長と平均計算時間

	(平均) 巡回路長				(平均) 計算時間 [s]			
	解法1	解法2	PDOT	厳密解	解法1	解法2	PDOT	厳密解
bays29	10255	10255	12944	10230	0.13	0.12	0.00	20
eil51	473	472	563	458	0.47	0.44	0.01	2345
eil76	600	606	682	572	1.07	0.98	0.02	7305
kroA100	22945	22729	28240		2.07	2.02	0.04	
kroA200	31540	31469	36826		9.67	9.62	0.16	
lin318	45143	45263	51381		26.25	24.65	0.42	
pa561	16559	16508	18040		106.58	99.09	1.23	
rat783	9513	9518	10571		205.97	202.58	2.57	

※ $n = 70$ 以上の問題例では現実的な時間内では厳密解を得ることができなかった.

付録:数値実験 結果

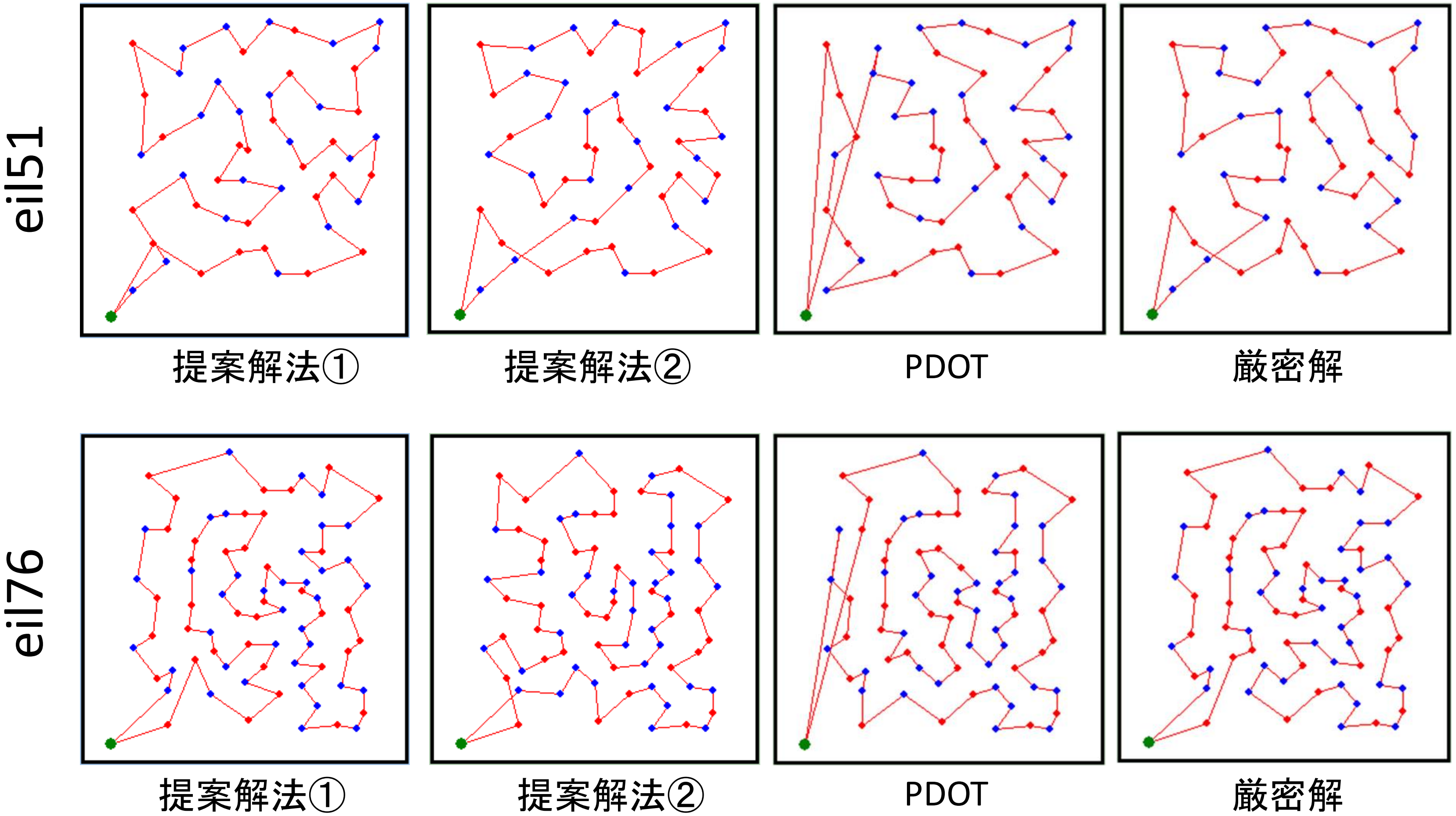


図19:数値実験結果