

収集コストとルーティングコスト を考慮した収集経路決定問題

沼田研究室

4402004 阿知波 次朗

発表構成

1. はじめに

1.1 背景

1.2 目的

2. モデル

2.1 前提条件

2.2 本研究で扱う
モデルのボロノイ図

3. 定式化

3.1 目的関数

3.2 アクセスコストと
ルーティングコスト

3.3 具体的な目的関数

4. 解法

4.1 解法

4.2 最急降下法による
局所的最適解の探索

4.3 最急降下方向の決定

5. 数値実験

5.1 アルゴリズム

5.2 実験方法

5.3 による影響

5.4 実験結果と考察

6. まとめ

7. 今後の課題

1. はじめに

1.1 背景

回収領域に**回収対象**が面的に分布している。
その回収対象を**停止点**に停まりながら**一回の巡回**
で回収して戻ってくる。

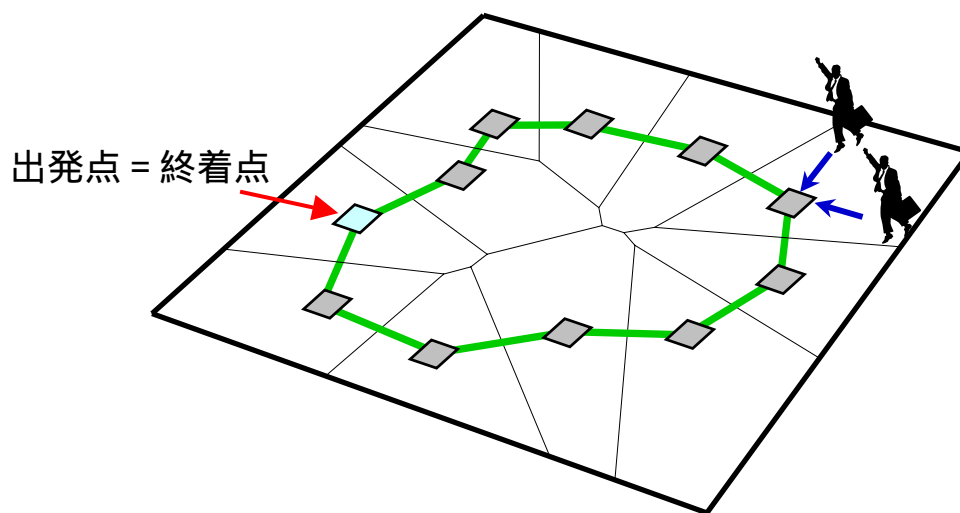


図1.1 収集経路決定問題

1. はじめに

1.1.1 収集経路決定問題

ある領域内に散らばるモノを回収して戻ってくる構造

- 移動図書館^[3]
- 石焼いも屋台
- 断水時の給水車
- 校庭の落ち葉集め
- ごみ収集車

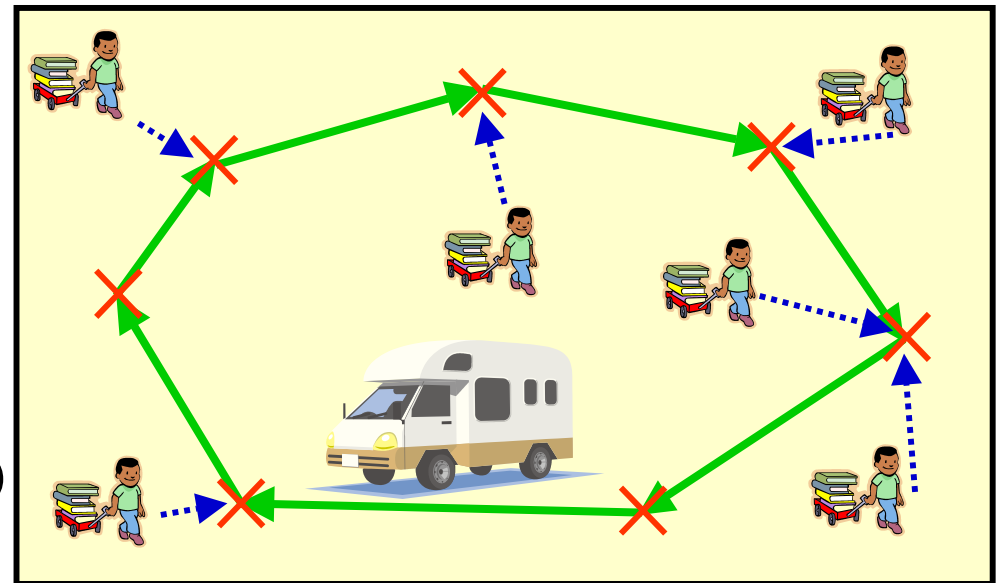


図1.2 図書館の移動経路

X: 図書館の貸出・返却地点

1.2 目的

{ **アクセスコスト**: 停止点へのアクセスにかかるコスト
ルーティングコスト: 停止点間の移動にかかるコスト

- ・ **新しいモデルとその解法の提案.**
- ・ **2つのコストの重視度合いによって回収経路の形状がどのように変化するかを調べる.**

プログラムを用いて回収経路の形状の変化を見る.

2. モデル

2.1 前提条件

2次元平面上に移動施設の停止位置を決めれば回収経路が決定する。

- ・ 経路は**巡回路** (出発点 = 終着点)
- ・ $f(x, y)$ は地点 (x, y) での**回収対象の分布**
- ・ 回収対象は最寄の停止点で回収
- ・ 巡回路は隣り合う停止点間を結ぶ**折れ線**

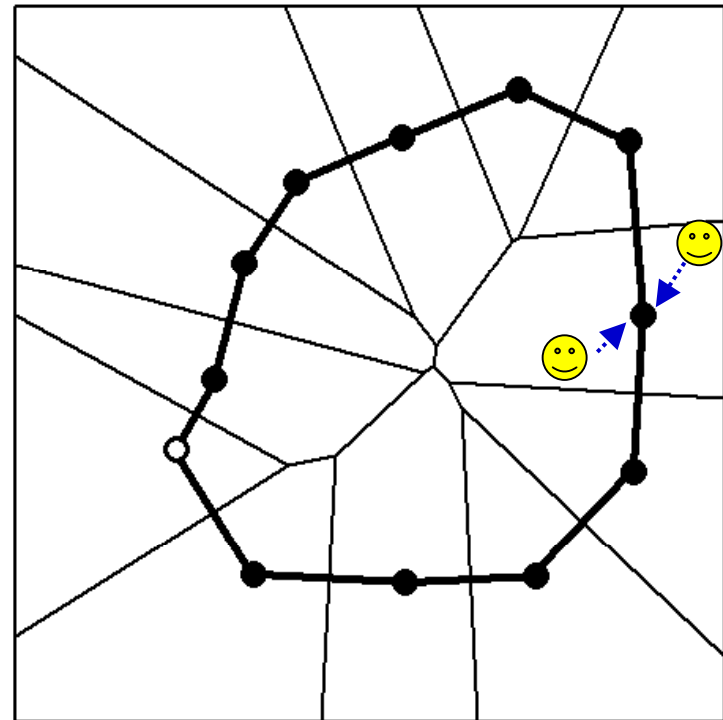


図2.1 アクセスコストとルーティングコスト。

2. モデル

2.2 停止点に対するボロノイ図

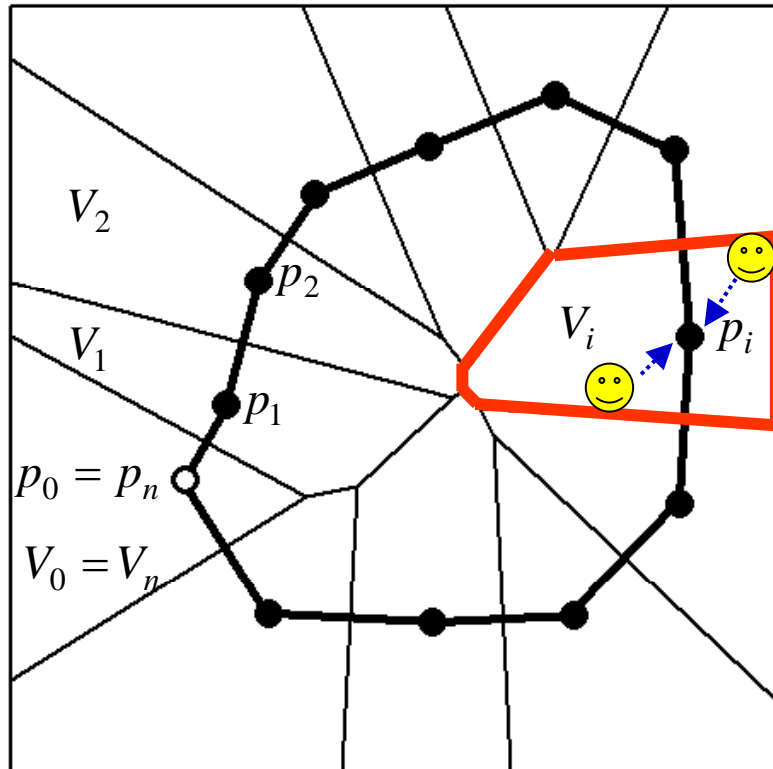


図2.2: アクセスコストとルーティングコスト.

: 停止点 p_i ($i = 1, 2, \dots, n-1$)

: 停止点 p_i ($i = 0, n$)

V_i : 停止点 p_i に対する
ボロノイ領域

● — ● : 停止点サービス
移動経路

p_i は与えられている

V_i に存在する人は p_i が
最も近い

3. 定式化

3.1 目的関数の導入

$$\min \quad \text{トータルコスト: } C_T = (1 - \alpha) \text{アクセスコスト} + \alpha(\text{ルーティングコスト}) \quad (3.1)$$

$\alpha (0 \leq \alpha \leq 1)$: 正の実数 (重みづけ)

アクセスコスト C_A : 利用者の停止点までの
平均移動距離

ルーティングコスト C_R : 停止点間の移動距離

3. 定式化

3.2 アクセスコストと ルーティングコスト

$$C_A : \frac{1}{N} \sum_{i=1}^n \iint_{V_i} \left\{ \sqrt{(x_i - x)^2 + (y_i - y)^2} \right\} f(x, y) dx dy \quad (3.2)$$

$$C_R : \sum_{i=1}^n \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \quad (3.3)$$

(x_i, y_i) : 点 p_i の $x - y$ 座標

V_i : 点 p_i に対するボロノイ領域

N : 回収領域内の総回収対象数

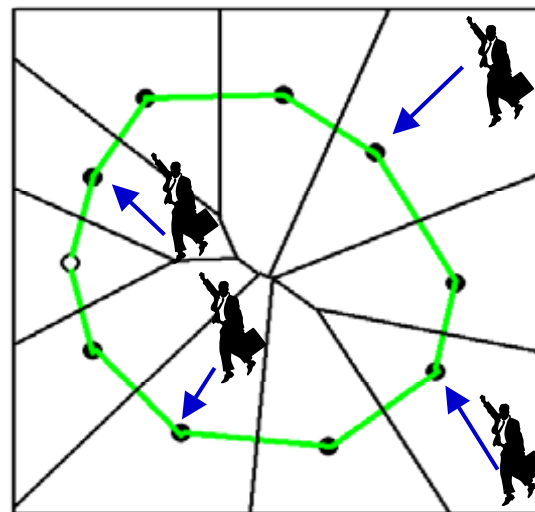
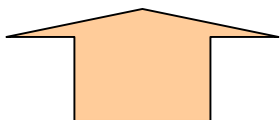


図3.1 アクセスコストとルーティングコスト。

3. 定式化

3.3 目的関数

$$C_T(\mathbf{z}) = (1 - \alpha) \frac{1}{N} \sum_{i=1}^n \iint_{V_i} \sqrt{(x_i - x)^2 + (y_i - y)^2} f(x, y) dx dy \quad C_A$$
$$+ \alpha \sum_{i=1}^n \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \quad C_R \quad (3.4)$$



上式を最小化する

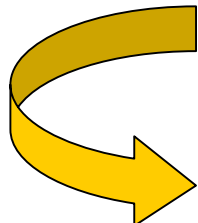
ただし $\mathbf{z} = \underbrace{(x_1, y_1, \dots, x_n, y_n)^T}_{2n \text{ 個}}$

4. 解法

4.1 解法

目的関数を最小にする z が求まれば
停止点の位置, すなわち, 巡回経路が決まる.

C_A に着目: 非凸な非線形計画問題^[2,3]

 大域的最適解を求めるのは困難.
局所的最適解を求める.

目的関数の最小化には最急降下法を用いる.

4. 解法

4.2 最急降下法による

局所的最適解の探索

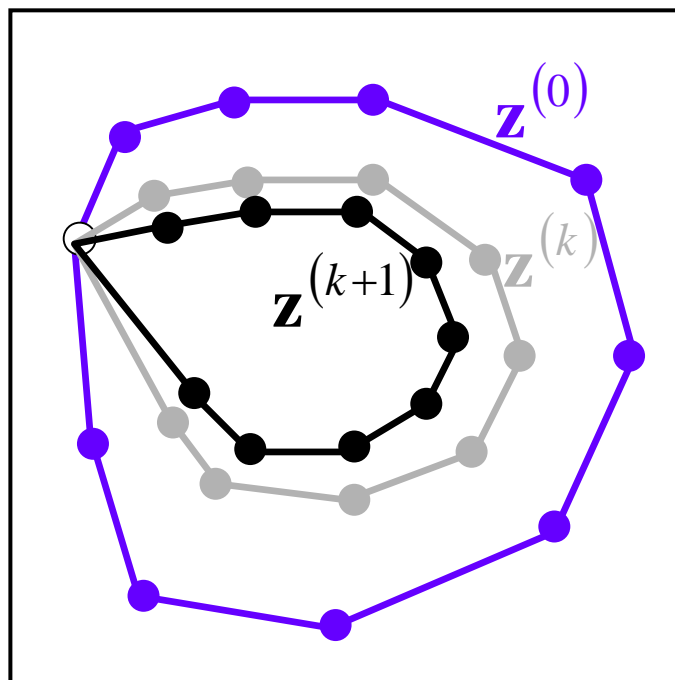


図4.1 更新前後の回収経路

$\mathbf{z}^{(k)}$, k 回目の停止点

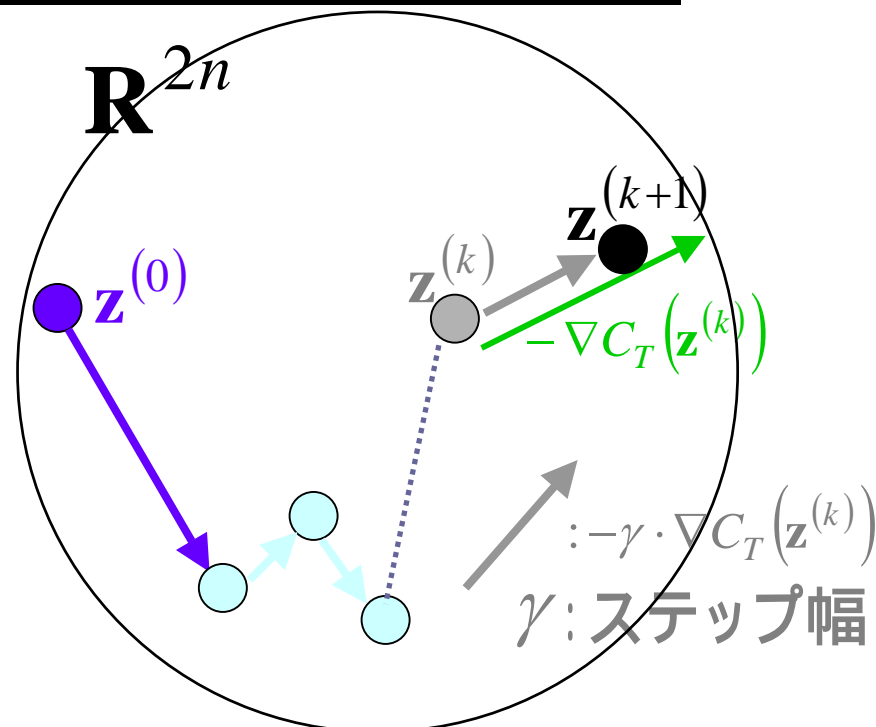


図4.2 最急降下法での解の決定方法
($2n$ 次元空間)

4. 解法

4.3 最急降下方向の決定

最急降下方向 $-\nabla C_T(\mathbf{z}) = -\left(\frac{\partial}{\partial x_1} C_T, \frac{\partial}{\partial y_1} C_T, \dots, \frac{\partial}{\partial x_n} C_T, \frac{\partial}{\partial y_n} C_T \right)^T$ は

以下のようになる。

2n個

x_i 成分

$$\begin{aligned} \frac{\partial}{\partial x_i} C_T(\mathbf{z}) = & \iint v_i \frac{(x_i - x) f(x, y)}{\sqrt{(x_i - x)^2 + (y_i - y)^2}} dx dy \\ & + \alpha \frac{(x_i - x_{i-1})}{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}} \\ & + \alpha \frac{(x_{i+1} - x_i)}{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}} \end{aligned} \quad (4.1)$$

5. 数値実験

5.1 プログラムの概要

停止点数, の入力

初期解ファイルの読み込み

初期解の回収経路・ボロノイ図を出力

初期解を逐次更新

最適解の決定

最適解の回収経路・ボロノイ図を出力

初期解・最適解のデータをファイル出力

```
停止点数はいくつですか？
16
現在のルーティングコストのWEIGHTを入力し
てください
0.9
```

図5.1: 停止点数, の入力



- ・ステップ幅
用意した数値の中から C_T を最小にするものを選択
- ・停止規則

プログラム作成

Microsoft Visual C++.net と LEDA (C++のライブラリ集) を使用

5. 数値実験

5.2 実験方法

入力: 停止点の数, α の値

出力: 目的関数値と初期解・最適解の回収経路

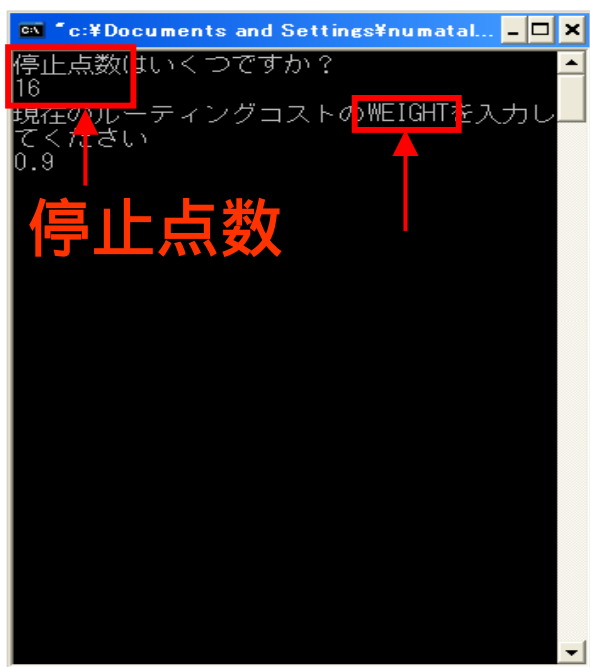


図5.2 入力画面.

2006/1/30

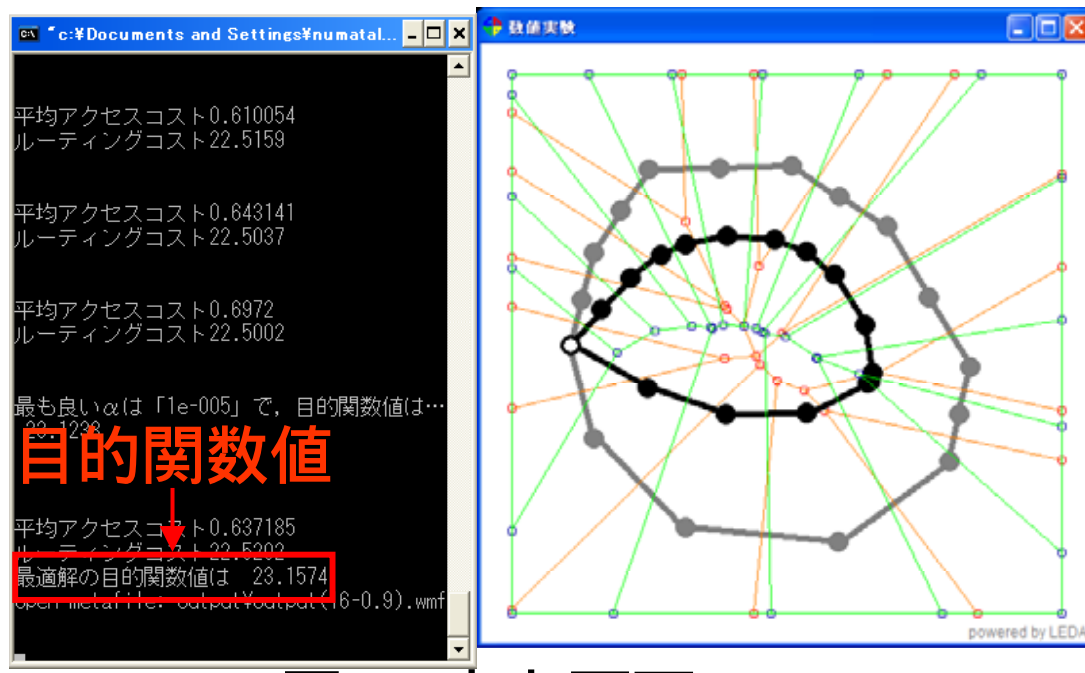


図5.3 出力画面.

卒業研究

5. 数値実験

5.3 α による影響

固定値: 停止点数、初期形状

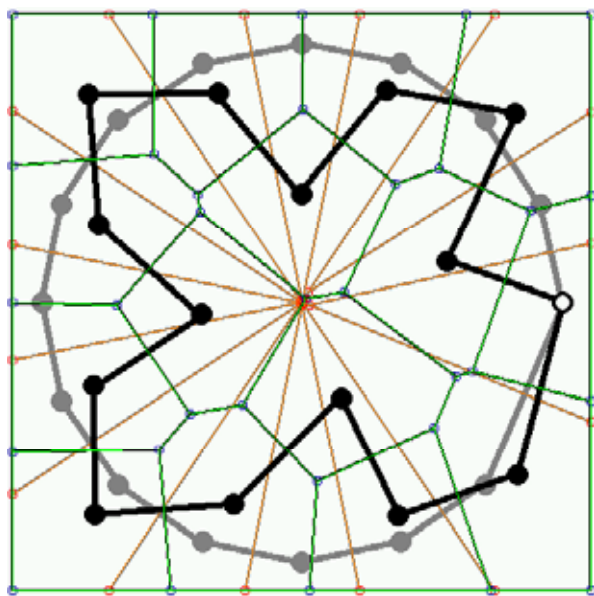


図5.4 初期解と最適解の回収経路.
($\alpha = 0.005$)

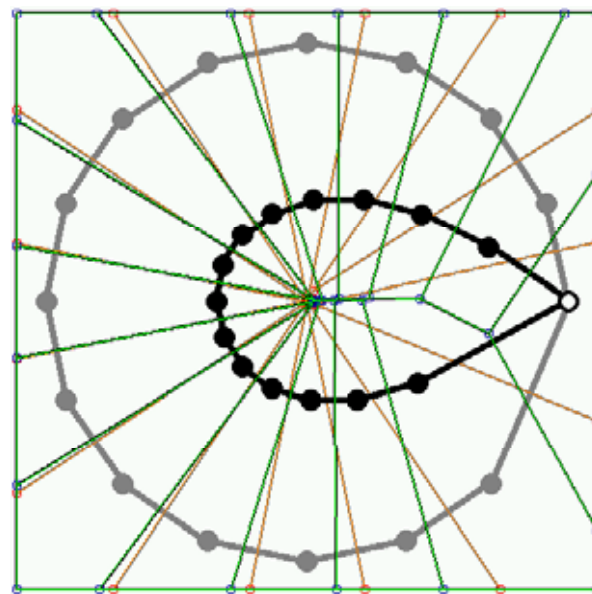


図5.5 初期解と最適解の回収経路.
($\alpha = 0.595$)

5. 数値実験

5.4 実験結果と考察

変数： α

定数：停止点数，初期形状

アクセスコスト

α : 重視度合い

ルーティングコスト

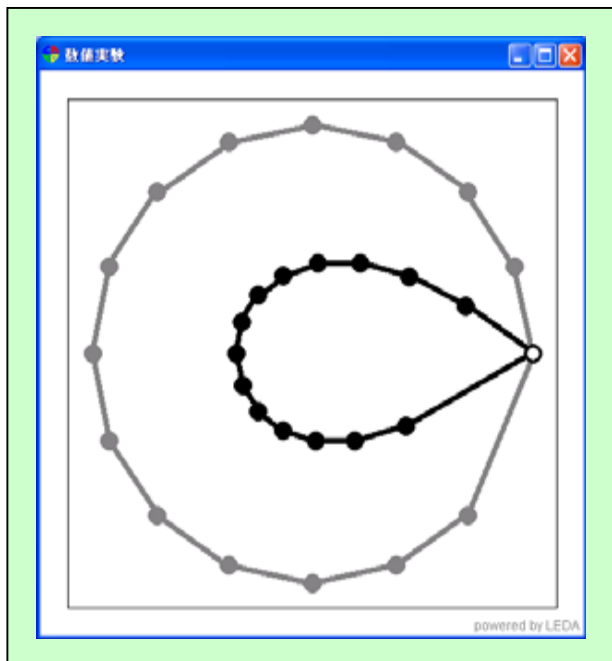


図5.6 初期解と最適解の回収経路。
出発点：右中

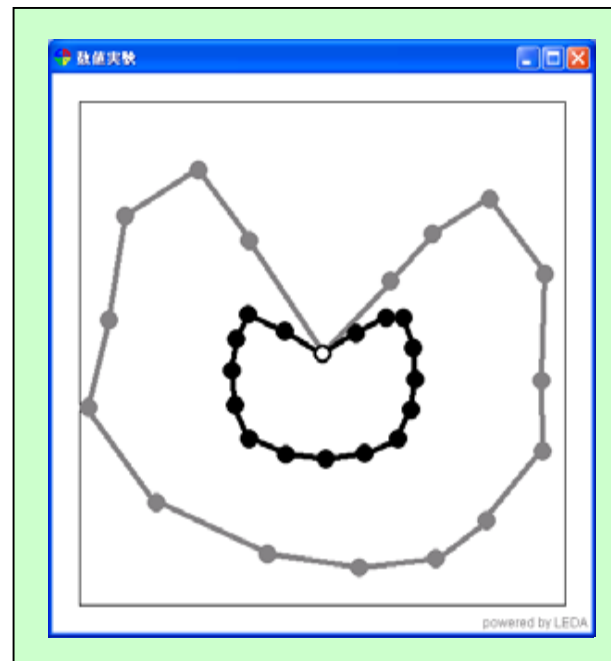
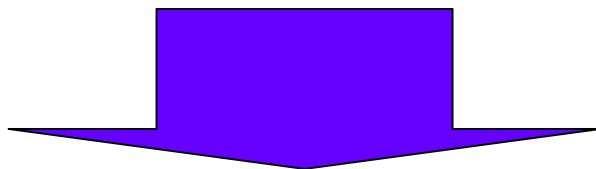


図5.7 初期解と最適解の回収経路。
出発点：中心

6. まとめ

- ・ **アクセスコスト**と**ルーティングコスト**を同時に考慮したモデル, 並びに解法の提案を行った.
- ・ の変化に伴い, 最適解の回収経路がどのように変化するか調べた.



提案モデルの解法が有用であることがわかった.

7. 今後の課題

- アクセスコストを異なる関数で与える。
 - 人の苦勞, アクセス量, モノの回収量に対応したアクセスコストなど
- アクセスコスト・ルーティングコスト以外のコスト関数を与える。
 - 停止点数に対応したコスト

参考文献

- [1]伊理 正夫(監), 腰塚 武志(編)(1986):『計算幾何学
と地理情報処理』, 第2版, 共立出版.
- [2]A. Okabe, B. Boots, K. Sugihara and S.-N. Chiu (2000):
*Spatial Tessellation Concepts and Applications of
Voronoi Diagrams*, 2nd Edition, John Wiley and Sons,
Chichester.
- [3]岡部 篤行, 鈴木 敦夫(1992):『最適配置の数理』, 朝
倉書店.
- [4]浅野 哲夫, 小保方 幸次(2002):『LEDAで始める
C/C++プログラミング』, サイエンス社.

抄録訂正

p110

$$C_T(x_1, y_1, \dots; x_n, y_n) = C_A(x_1, y_1, \dots; x_n, y_n) + \alpha \cdot C_R(x_1, y_1, \dots; x_n, y_n)$$

$$C_T(x_1, y_1, \dots; x_n, y_n) = (1-\alpha) \cdot C_A(x_1, y_1, \dots; x_n, y_n) + \alpha \cdot C_R(x_1, y_1, \dots; x_n, y_n)$$

$(0 \leq \alpha \leq 1)$