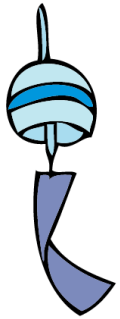
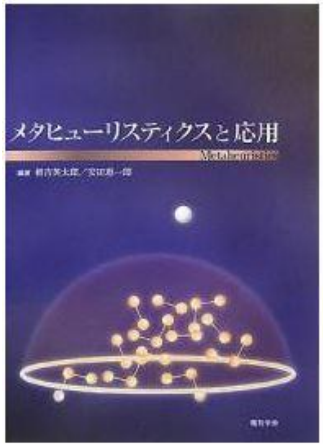


メタヒューリスティクスと応用

～第6章 スケジューリング問題への応用～

タブーサーチを用いた配送計画システム
配送計画への応用(誘導局所探索法)



メタヒューリスティクスゼミ　～夏の祭典～
2013/07/10(水)　　M1 今泉孝章

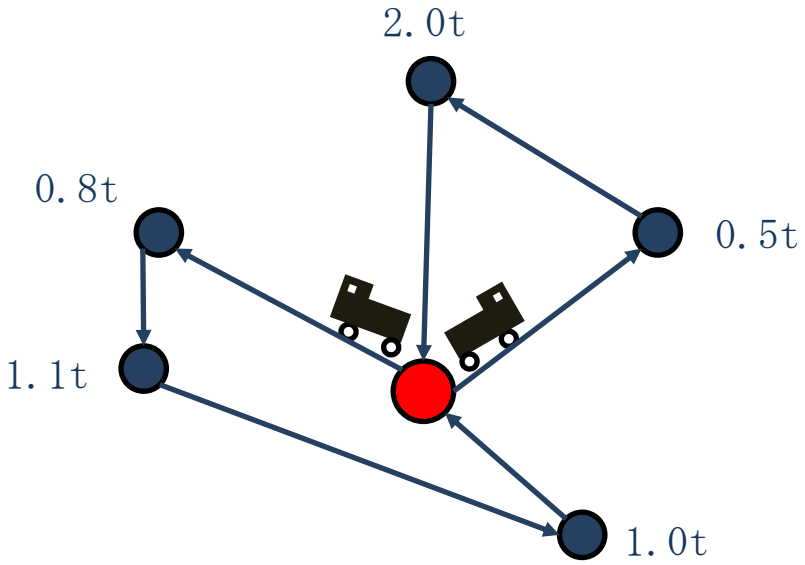
タブーサーチを用いた配送計画システムと配送計画への応用(誘導局所探索法)

- **配送計画問題（Vehicle Routing Problem）**

  デポ、配送先、車両、各配送先の需要が所与のとき最もコストが小さくなるように、配送先割り付け、配送順を決定する問題


×2

1台で3t運べるとする



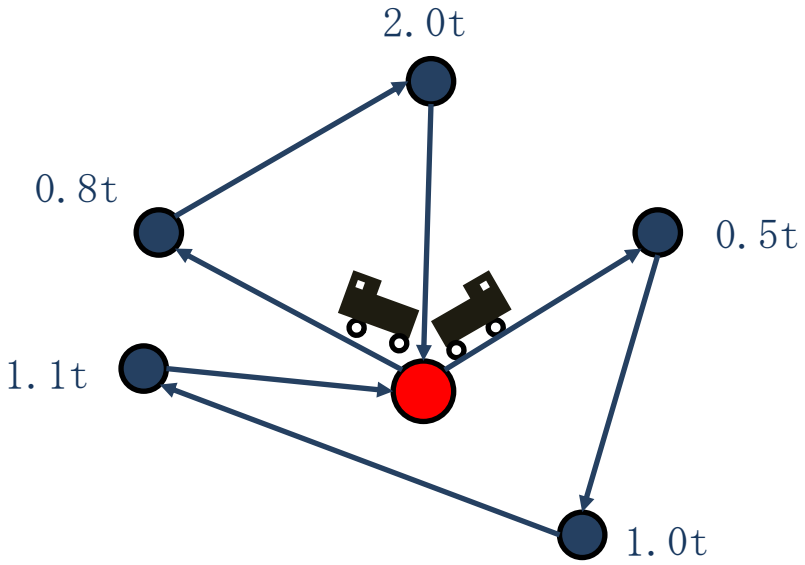
タブーサーチを用いた配送計画システムと配送計画への応用(誘導局所探索法)

●配送計画問題（VRP： Vehicle Routing Problem）

デポ、配送先、車両、各配送先の需要が所与のとき最もコストが小さくなるように、配送先割り付け、配送順を決定する問題


×2

1台で3t運べるとする



その他制約

- 時間制約
- 通行制限

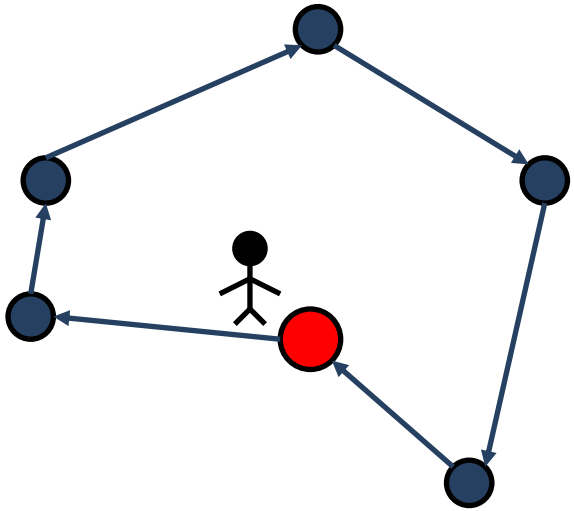
タブーサーチを用いた配送計画システムと配送計画への応用(誘導局所探索法)

●巡回セールスマン問題（TSP： Tranveling Salesman Problem）

訪問する都市が与えられたとき、それらをそれぞれ一回ずつ訪問して出発地に戻る最短ルートを決定する問題

VRPにおいて目的関数が総移動距離、車両が1台配送時刻、配送物などの制約がなくなったもの

↓



改善法における初期解の設定が難しい

タブーサーチを用いた配送計画システムと配送計画への応用(誘導局所探索法)

局所探索法アルゴリズム

```
1  x := 適当な初期解
2  while improve(x) ≠ φ do
3      x := improve(x)
4  return x
```

→初期解の作成

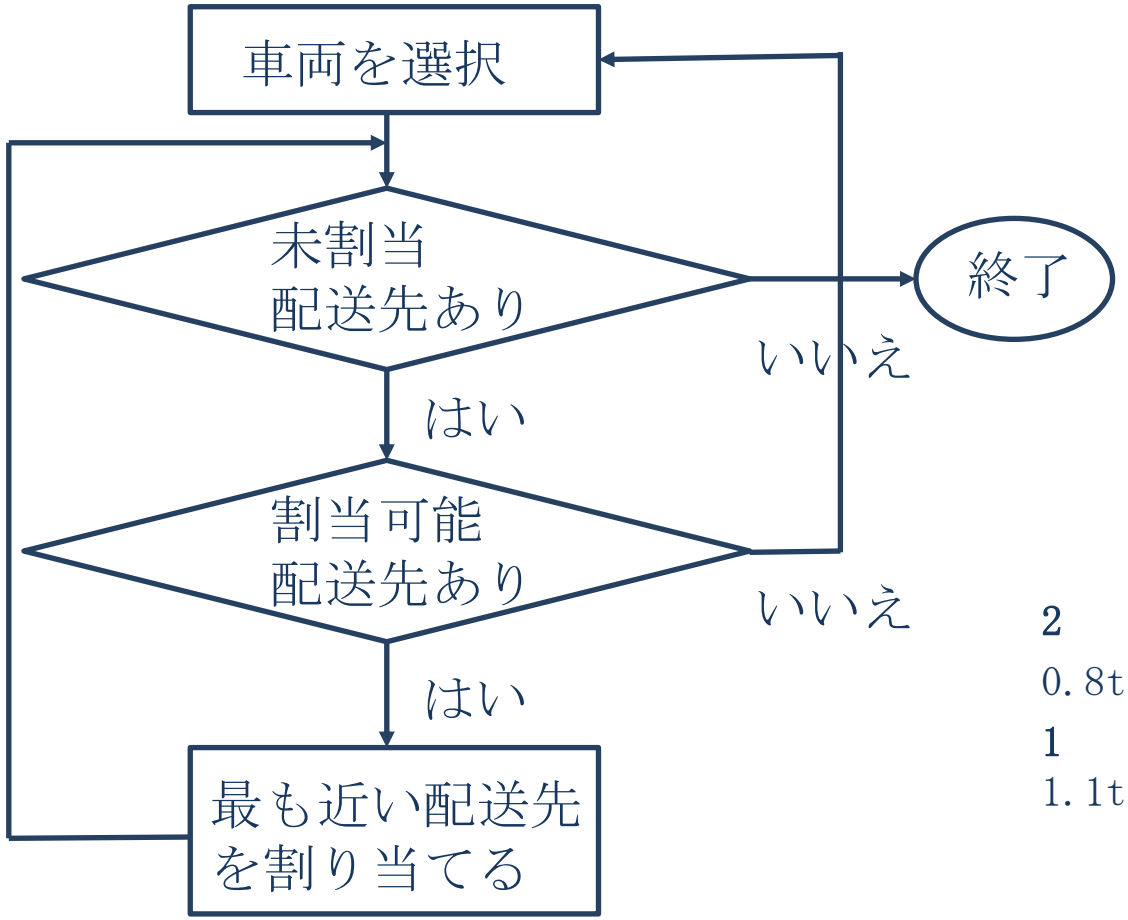
→近傍の構成

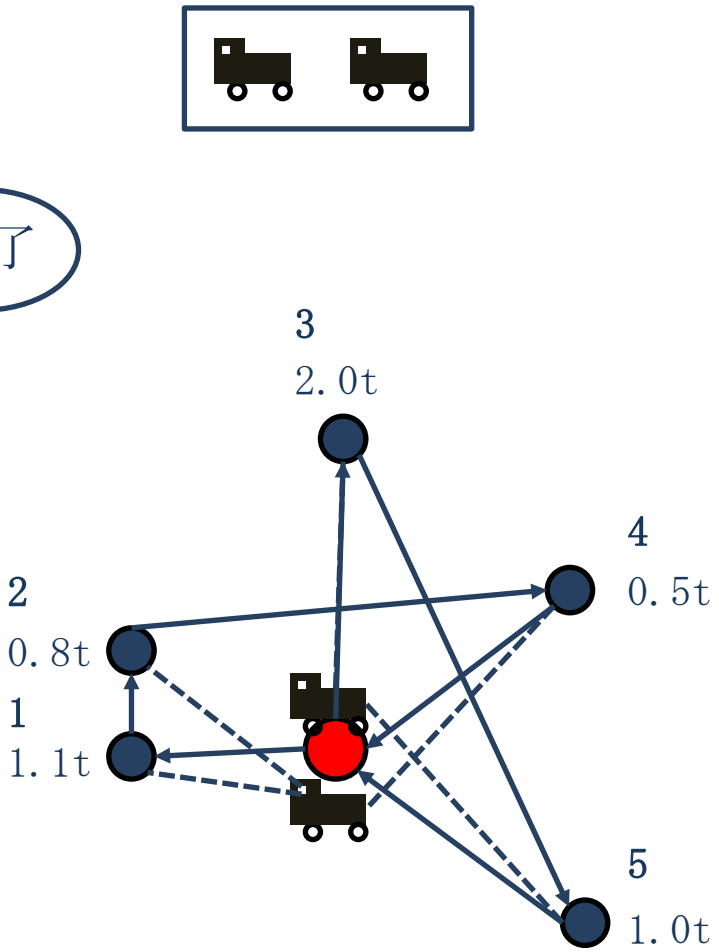
→解の評価

それぞれの要素が配送計画問題でどのように扱われているか説明する

タブーサーチを用いた配送計画システムと配送計画への応用(誘導局所探索法)

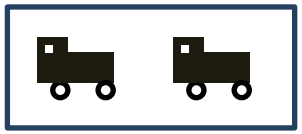
初期解の作成～その1～





タブーサーチを用いた配送計画システムと配送計画への応用(誘導局所探索法)

初期解の作成～その2 セービング法～



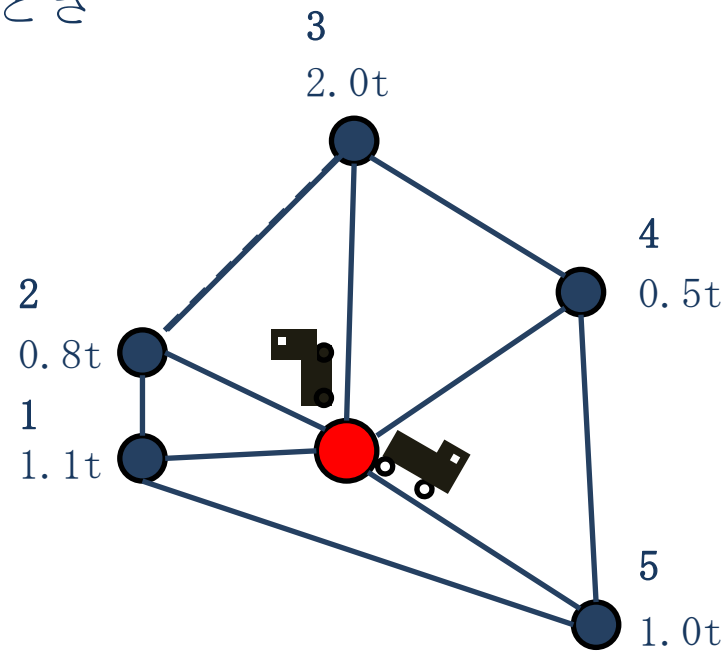
1:配送先と同じ台数の車両があるとしすべての配送先との往復ルートをつくる

↓

2:需要を超過しない配送先同士をつないだときに最も距離が小さいものからつなぐ

↓

3:車両台数になるまで継続する



タブーサーチを用いた配送計画システムと配送計画への応用(誘導局所探索法)

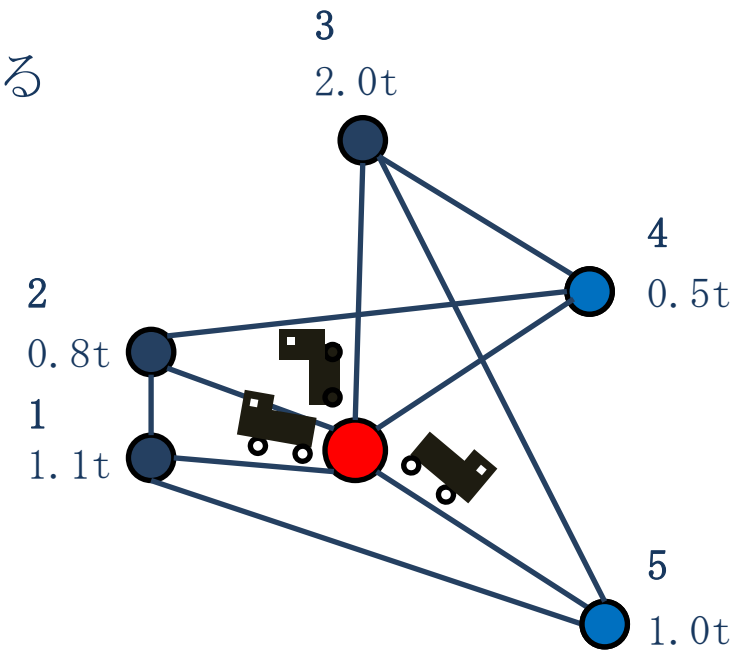
近傍の構成

初期解の一部を変更することで解の改善を行う→近傍に移動

1. 車両割り当て変更

(a)配送先の交換

二つの車両間で配送先を交換する



タブーサーチを用いた配送計画システムと配送計画への応用(誘導局所探索法)

近傍の構成

初期解の一部を変更することで解の改善を行う→近傍に移動

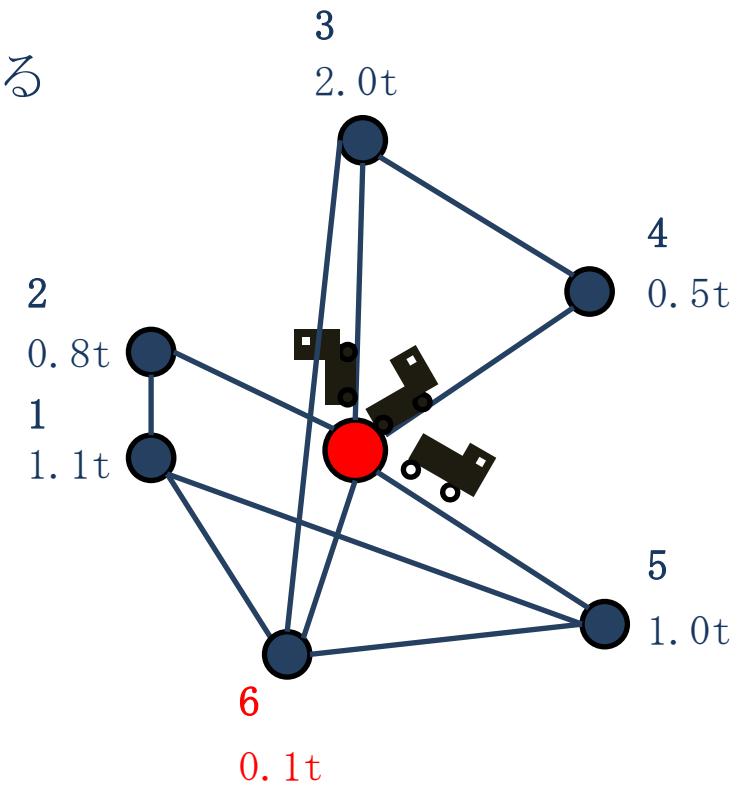
1. 車両割り当て変更

(a)配送先の交換

二つの車両間で配送先を交換する

(b)配送先の移動(削除・追加)

車両の配送先を一つ削除し別の車両に追加する



タブーサーチを用いた配送計画システムと配送計画への応用(誘導局所探索法)

近傍の構成

初期解の一部を変更することで解の改善を行う→近傍に移動

1. 車両割り当て変更

   (a)配送先の交換

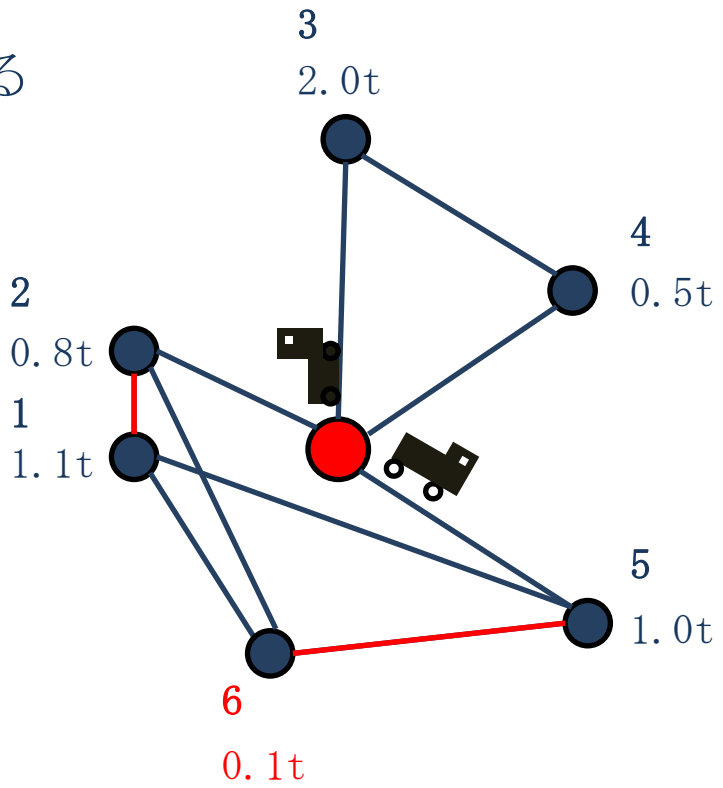
   二つの車両間で配送先を交換する

   (b)配送先の移動(削除・追加)

   車両の配送先を一つ削除し別の車両に追加する

2. 配送順序の変更

   2-opt法などを用いる



1、2のいづれかもしくは両方を用いて近傍を構成する

タブーサーチを用いた配送計画システムと配送計画への応用(誘導局所探索法)

近傍解の評価(改善か改悪か)

評価指標

- 使用車両台数
- 車両の総走行距離
- 業務時間(最後の車両が配送センタに到着する時間)
- 配送時間が違反されているかのペナルティ値
- 複数の評価指標の総和

厳密に問題を定義し誘導局所探索法によってVRPの解法を示していく

タブーサーチを用いた配送計画システムと配送計画への応用(誘導局所探索法)

定数・変数の定義

$K = \{0,1,\cdots,m\}$: 車両の集合

$N = \{0,1,\cdots,n\}$: 配送先、デポの集合(デポはN=0)

$d_{ij} \quad (i, j \in N \ and \ i \neq j)$: 点iと点jとの距離→配送先、デポ間の距離

u_k^d : 車両kの距離単価

$c_{ij}^k \ (k \in K \ and \ i \neq j)$: 車両kが点ij間を移動する際のコスト
一般に $c_{ij}^k = d_{ij} u_k^d$

$x_{ij}^k = \begin{cases} 1 \\ 0 \end{cases}$ 車両kが点iから点jへ移動する場合1,それ以外0

タブーサーチを用いた配送計画システムと配送計画への応用(誘導局所探索法)

目的関数

$$\sum_{i,j\in N,i\neq j,k\in K} c_{ij}x_{ij}^{k} = \sum_{i,j\in N,i\neq j,k\in K} d_{ij}u_{k}^{d}x_{ij}^{k}$$

→ルートとして実際に利用したものについてのコストの総和

制約条件

$$\sum_{i\in N,i\neq j,k\in K} x_{ij}^{k} = 1 \quad for \quad {}^{\forall} j \in N$$

$$\sum_{j\in N,i\neq j,k\in K} x_{ij}^{k} = 1 \quad for \quad {}^{\forall} i \in N$$

→どの配送先も1台の車両によって配送される

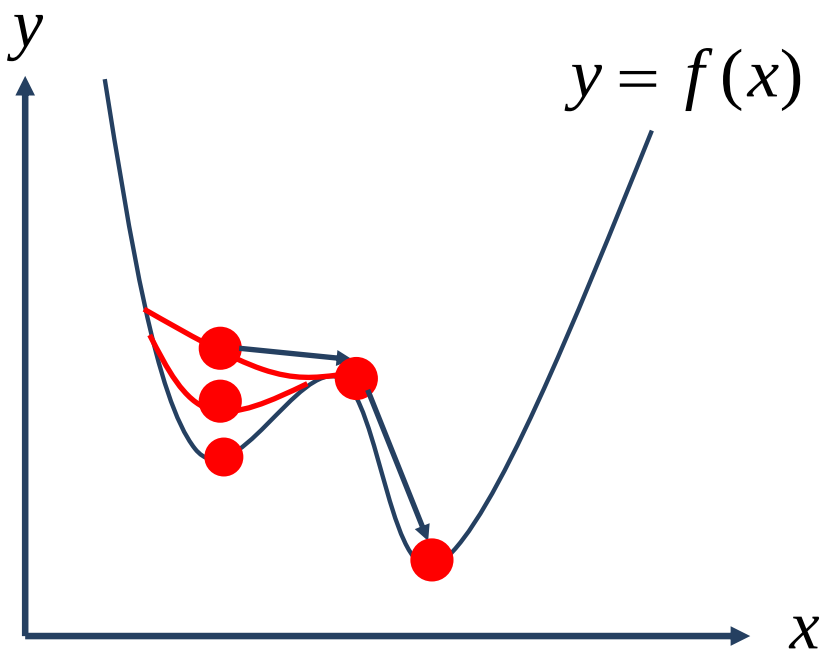
$$\sum_{i\in N,k\in K} x_{i0}^{k} = m$$ デポへの到着

$$\sum_{j\in N,k\in K} x_{0j}^{k} = m$$ デポから出発

→デポへは車両がm回出入りする

タブーサーチを用いた配送計画システムと配送計画への応用(誘導局所探索法)

誘導局所探索法を用いたVRP



ペナルティ更新規則

$$u' = \arg\max_{u \in U} \frac{c(u)}{1 + LTM(u)}$$

$$LTM(u') := LTM(u') + 1$$

もとの目的関数 $f(x)$ が「要素」によって分解されると仮定

↓

要素の集合を U とし $u \in U$ のある要素 u に関するコストを $c(u)$ とする

↓

$$f(x) = \sum_{u \in U} c(u) x_u$$

コストの大きい要素にペナルティ $LTM(u)$ を課し目的関数を拡張

↓

$$\bar{f}(x, \lambda) = \sum_{u \in U} \{c(u) x_u + \lambda \cdot LTM(u)\}$$

タブーサーチを用いた配送計画システムと配送計画への応用(誘導局所探索法)

誘導局所探索法を用いたVRP

VRPにおける要素、要素に関するコストとは?

機械スケジューリング問題の場合
要素: ジョブの処理順
　　とすでに行われたジョブ
コスト: 遅れ

ジョブ1

処理順番	1	2	3	4
行われたジョブ	なし	2|3|4	23|34|42	234
遅れ	0	0|0|0	1|3|2	5

ジョブ2

処理順番	1	2	3	4
行われたジョブ	なし	1|3|4	13|34|41	134
遅れ	0	0|0|0	0|0|0	1

ジョブ3

処理順番	1	2	3	4
行われたジョブ	なし	1|2|4	12|24|41	124
遅れ	0	0|0|1	0|3|2	4

ジョブ4

処理順番	1	2	3	4
行われたジョブ	なし	1|2|3	12|23|31	123
遅れ	0	1|2|3	3|5|4	6

タブーサーチを用いた配送計画システムと配送計画への応用(誘導局所探索法)

VRPにおける要素、要素に関するコストとは?

目的関数 $\sum_{i,j\in N, i\neq j, k\in K} c_{ij}x_{ij}^{k} = \sum_{i,j\in N, i\neq j, k\in K} d_{ij}u_{k}^{d}x_{ij}^{k}$ →関連するインデックスは i、j、k

従って要素集合 U は i 、j 、k の組み合わせの集合となりその要素 u は $u=(i,j,k)$ で表される

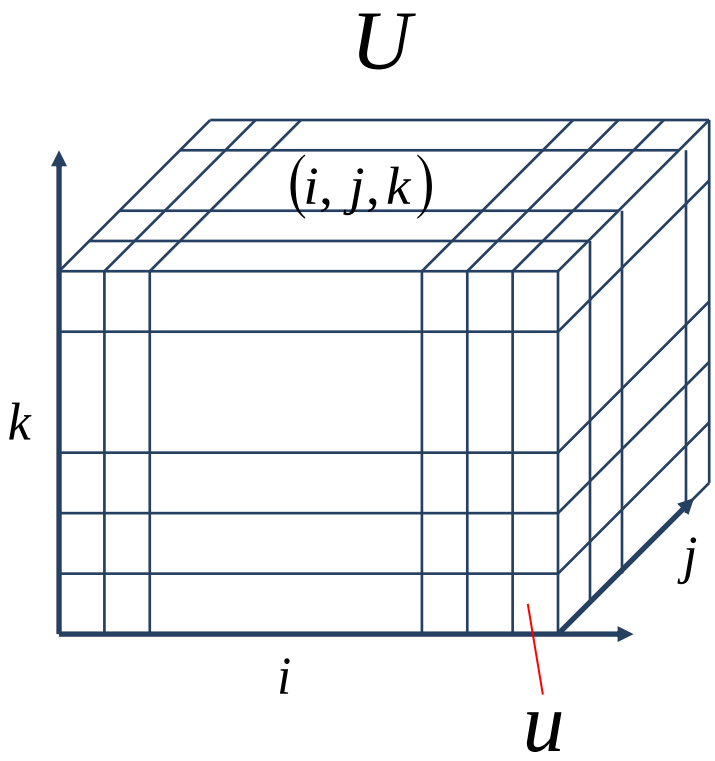
→車両 k で点 ij 間を移動するということが要素になる

コストは車両 k で点 ij 間を移動したときのコストになる→ $d_{ij}u_{k}^{d}$

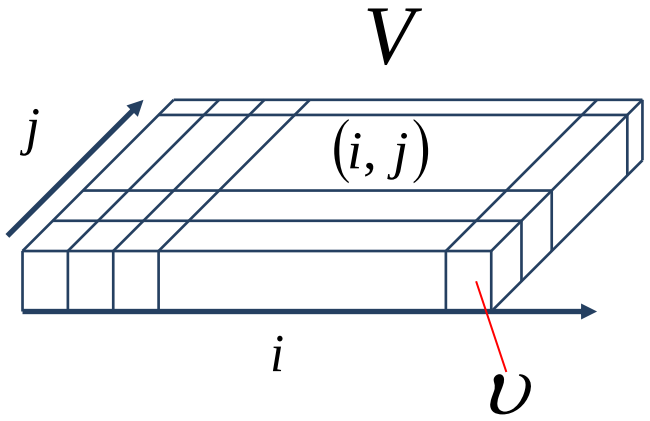
$$f(x)=\sum_{u\in U} c(u)x_{u} = \sum_{i,j\in N, i\neq j, k\in K} d_{ij}u_{k}^{d}x_{ij}$$

実際には要素をさらに分割する

タブーサーチを用いた配送計画システムと配送計画への応用(誘導局所探索法)



要素を分解して
新しい要素をつくる→



車両 k を同一視して都市間の
移動 (i, j) のみを要素とする

新しい要素集合 を V 、その要素を υ 、要素 υ に関するコストを $c(\upsilon)$ 、
解が要素 υ を持つとき1、持たないとき0の決定変数を x_υ とする

$$f(x)=\sum_{u\in U} c(u)x_u = \sum_{i,j\in N, i\neq j, k\in K} d_{ij}u_k^d x_{ij} = \sum_{\upsilon\in V} c(\upsilon)x_\upsilon$$

タブーサーチを用いた配送計画システムと配送計画への応用(誘導局所探索法)

$$LTM(u) \rightarrow LTM'(\upsilon)$$ ペナルティを課す要素が変化

$$\bar{f}(x,\lambda) = \sum_{u \in U}\{c(u)x_u + \lambda \cdot LTM(u)\} \rightarrow \sum_{\upsilon \in V}\{c(\upsilon)x_\upsilon + \lambda \cdot LTM'(\upsilon)\}$$

拡張目的関数

ペナルティ更新規則

$$\upsilon' = \arg\max_{\upsilon \in V} \frac{c(\upsilon)}{1 + LTM'(\upsilon)}$$

$$LTM(\upsilon') := LTM(\upsilon') + 1$$

パラメータ更新則

$$\lambda = \alpha \times \left\{ \frac{\sum_{\upsilon \in V^+} c(\upsilon)}{|V^+|} \right\}$$

V^+ : 要素の集合 V のうち $c(\upsilon) > 0$ の部分集合

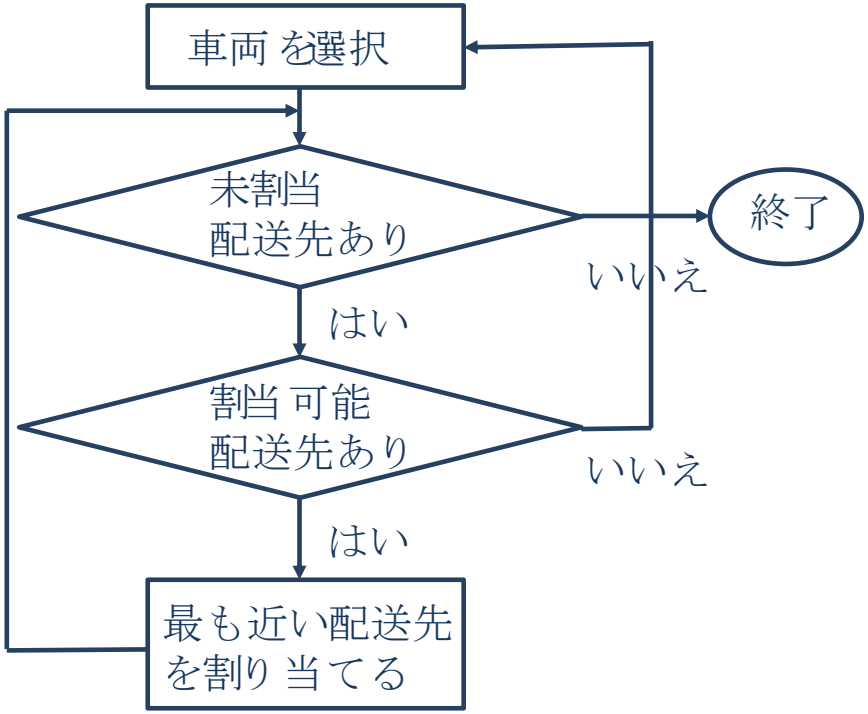
α : 0～1の値をとる平準化パラメータ

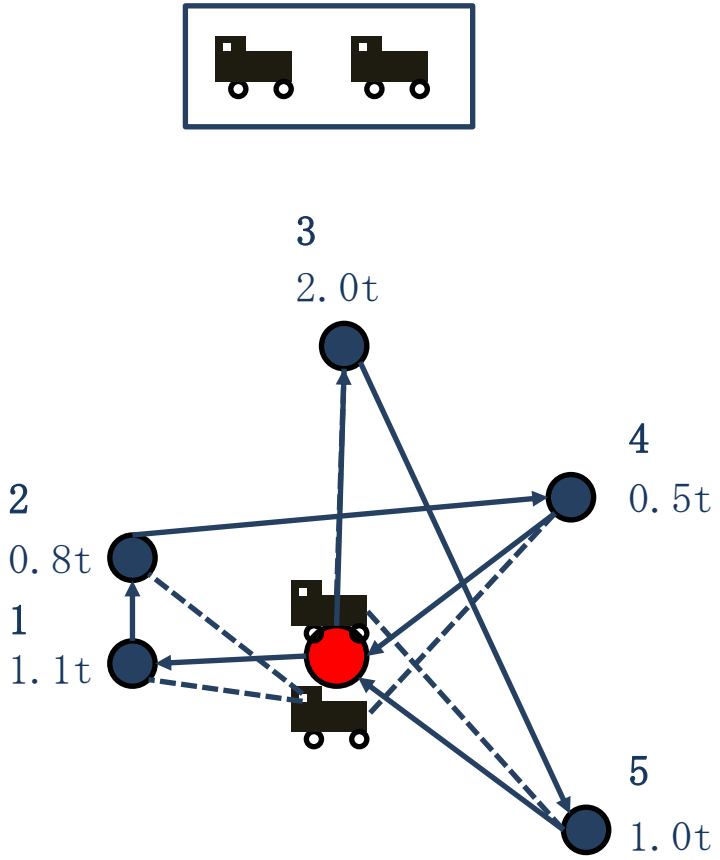
得られた解の要素の中で配送先間の距離が大きいものから優先的にペナルティが課される

タブーサーチを用いた配送計画システムと配送計画への応用(誘導局所探索法)

アルゴリズム

1 x :=適当な初期解





タブーサーチを用いた配送計画システムと配送計画への応用(誘導局所探索法)

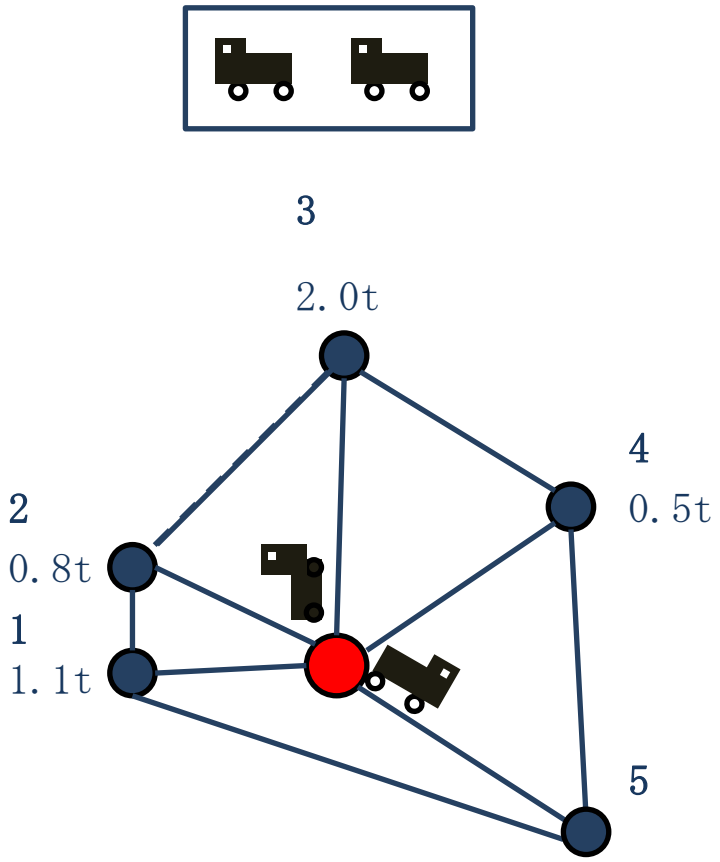
アルゴリズム

1 x :=適当な初期解

1:配送先と同じ台数の車両があるとしすべての配送先との往復ルートをつくる

2:需要を超過しない配送先同士をつないだときに最も距離が小さいものからつなぐ

3:車両台数になるまで継続する



タブーサーチを用いた配送計画システムと配送計画への応用(誘導局所探索法)

アルゴリズム

1 $x :=$ 適当な初期解

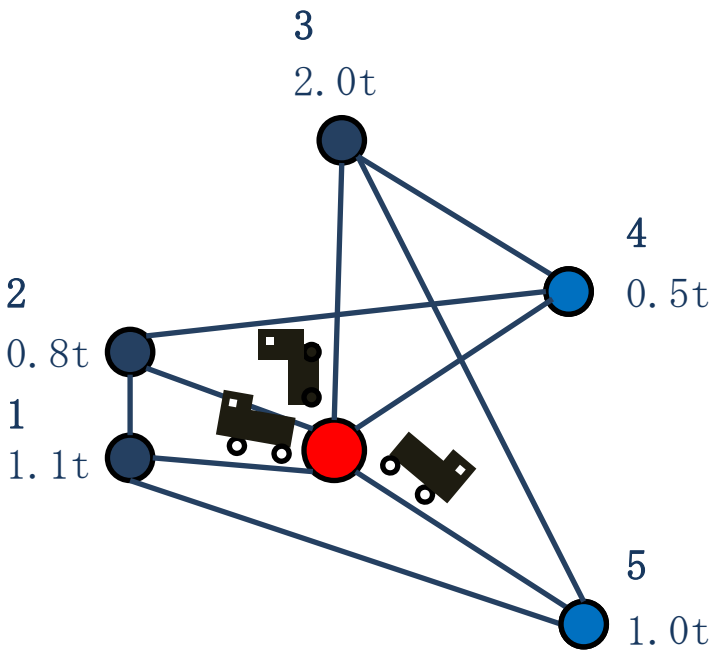
2 最良解 $x^* := x$

3 長期メモリ $LTM'(v)$ $\forall v \in V$ を0に初期化

4 *while* 終了基準が満たされていない *do*

5 *while* $improve(x, \lambda) \neq \phi$ *do*

1. 車両割り当て変更
   (a)配送先の交換
   二つの車両間で配送先を交換する



タブーサーチを用いた配送計画システムと配送計画への応用(誘導局所探索法)

アルゴリズム

1 $x :=$ 適当な初期解

2 最良解 $x^* := x$

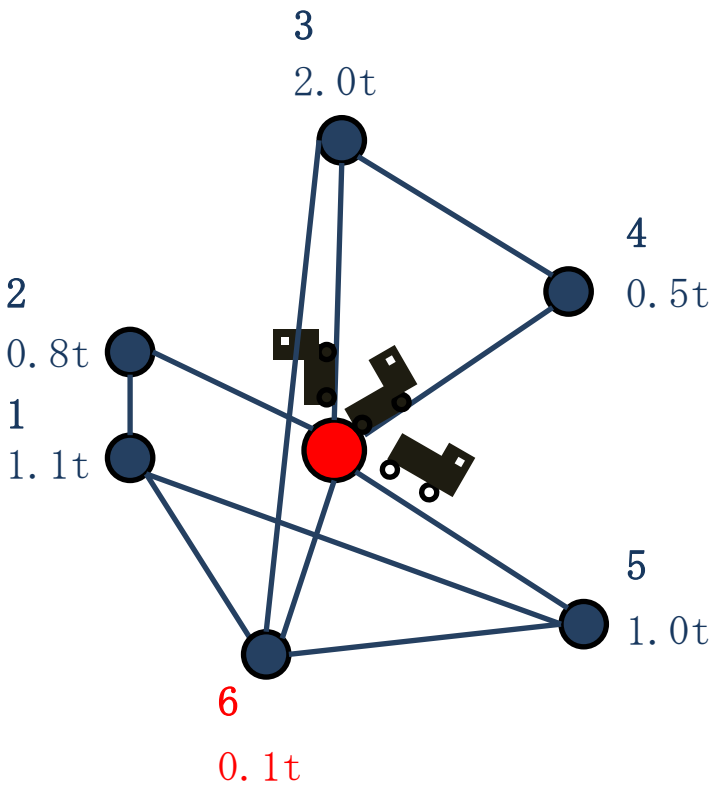
3 長期メモリ $LTM'(\upsilon)\ \forall \upsilon \in V$ を0に初期化

4 *while* 終了基準が満たされていない *do*

5 *while* $improve(x, \lambda) \neq \phi$ *do*

1. 車両割り当て変更
   (a)配送先の交換
   二つの車両間で配送先を交換する

   (b)配送先の移動(削除・追加)
   車両の配送先を一つ削除し
   別の車両に追加する



タブーサーチを用いた配送計画システムと配送計画への応用(誘導局所探索法)

アルゴリズム

1 $x :=$ 適当な初期解

2 最良解 $x^* := x$

3 長期メモリ $LTM'(v)\ \forall v \in V$ を0に初期化

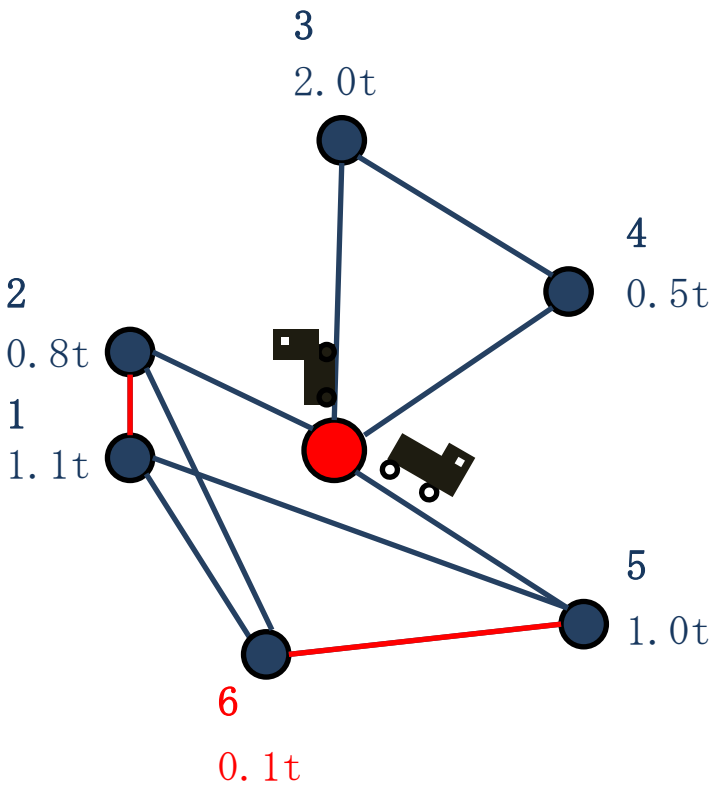
4 *while* 終了基準が満たされていない *do*

5 *while* $improve(x, \lambda) \neq \phi$ *do*

1. 車両割り当て変更
   (a)配送先の交換
   二つの車両間で配送先を交換する

   (b)配送先の移動(削除・追加)
   車両の配送先を一つ削除し
   別の車両に追加する

2. 配送順序の変更
   2-opt法などを用いる



タブーサーチを用いた配送計画システムと配送計画への応用(誘導局所探索法)

アルゴリズム

1 $x :=$ 適当な初期解

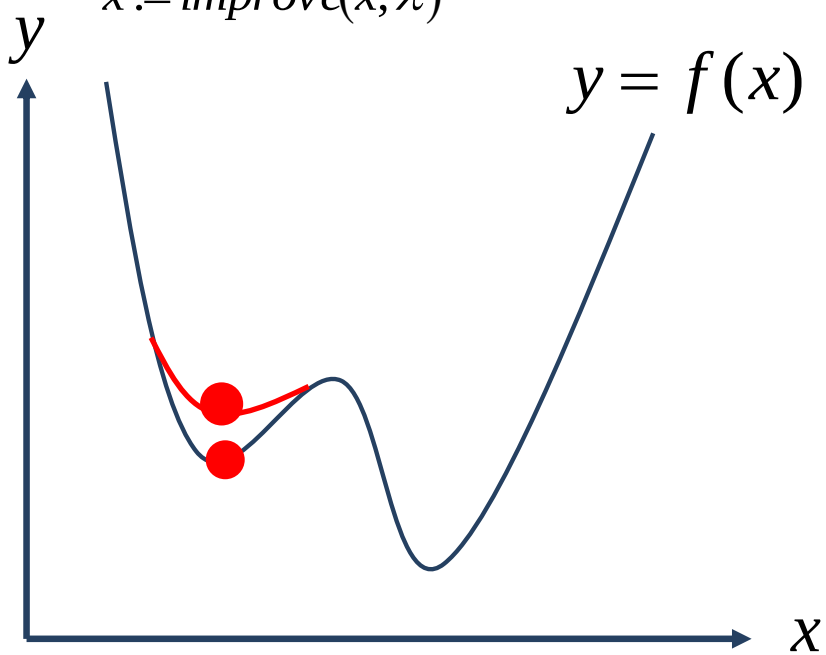
2 最良解 $x^* := x$

3 長期メモリ $LTM'(\upsilon)\ \forall \upsilon \in V$ を0に初期化

4 *while* 終了基準が満たされていない *do*

5 *while* $improve(x,\lambda) \neq \phi$ *do*

6 $x := improve(x,\lambda)$



目的関数

$$\sum_{\upsilon \in V} \{c(\upsilon)x_\upsilon + \lambda \cdot LTM'(\upsilon)\}$$

$$\lambda = \alpha \times \left\{ \frac{\sum_{\upsilon \in V^+} c(\upsilon)}{|V^+|} \right\}$$

近傍解を目的関数で評価

タブーサーチを用いた配送計画システムと配送計画への応用(誘導局所探索法)

アルゴリズム

1 $x :=$ 適当な初期解

2 最良解 $x^* := x$

3 長期メモリ $LTM'(v)$ $\forall v \in V$ を0に初期化

4 *while* 終了基準が満たされていない *do*

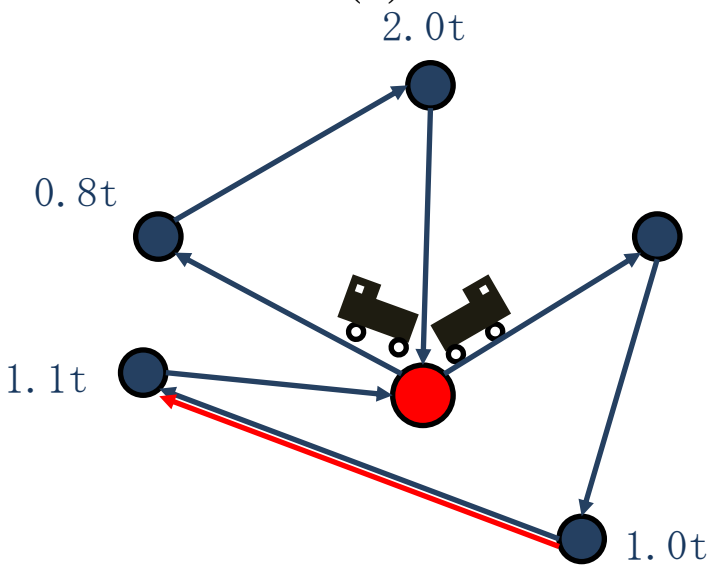
5 　*while* $improve(x, \lambda) \neq \phi$ *do*

6 　　$x := improve(x, \lambda)$

7 　得られた局所最適解 x を用いて長期メモリ $LTM'(v)$ を更新

$$v' = \arg\max_{v \in V} \frac{c(v)}{1 + LTM'(v)}$$

$$LTM(v') := LTM'(v') + 1$$



タブーサーチを用いた配送計画システムと配送計画への応用(誘導局所探索法)

アルゴリズム

1 $x :=$ 適当な初期解

2 最良解 $x^* := x$

3 長期メモリ $LTM'(\upsilon)$ $\forall \upsilon \in V$ を0に初期化

4 *while* 終了基準が満たされていない *do*

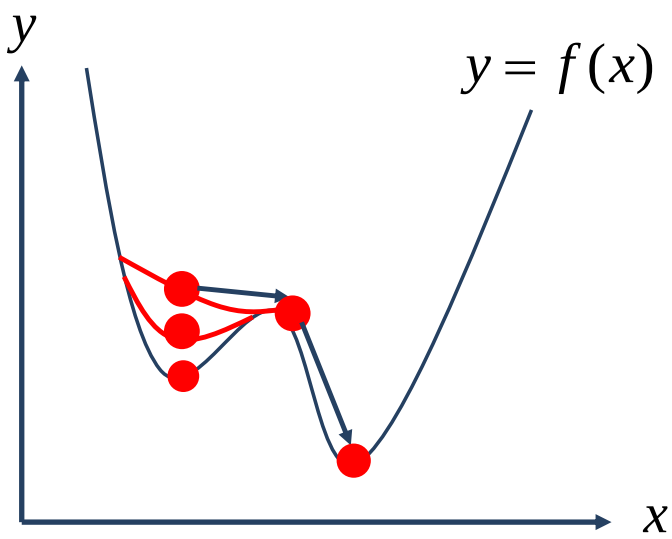
5 　　*while* $improve(x, \lambda) \neq \phi$ *do*

6 　　$x := improve(x, \lambda)$

7 　　得られた局所最適解 x を用いて長期メモリ $LTM'(\upsilon)$ を更新

8 　　ペナルティ用パラメータ λ を更新

$$\lambda = \alpha \times \left\{ \frac{\sum_{\upsilon \in V^+} c(\upsilon)}{|V^+|} \right\}$$



タブーサーチを用いた配送計画システムと配送計画への応用(誘導局所探索法)

アルゴリズム

1 $x :=$ 適当な初期解

2 最良解 $x^* := x$

3 長期メモリ $LTM'(v)$ $\forall v \in V$ を0に初期化

4 *while* 終了基準が満たされていない *do*

5 　　*while* $improve(x, \lambda) \neq \phi$ *do*

6 　　$x := improve(x, \lambda)$

7 　　得られた局所最適解 x を用いて長期メモリ $LTM'(v)$ を更新

8 　　ペナルティ用パラメータ λ を更新

9 　　*if* $f(x) < f(x^*)$ *then* $x^* := x$

10 *return* x^*