自動倉庫における入出庫スケジューリング問題の解法の提案

沼田研究室
4402057　野村　要

発表構成

1．はじめに
　1．1　本研究の背景
　1．2　自動倉庫
　1．3　自動倉庫における問題点
　1．4　問題に対する先行研究
　1．5　研究の目的
2．問題設定
3．定式化
　3．1　記号の定義
　3．2　目的関数
　3．3　制約条件
　3．4　定式化のまとめ
4．文献[1]の解法
5．提案する解法
6．数値実験
　6.1実験概要
　6.2実験結果と考察
7．まとめ
8．今後の課題

1．1　本研究の背景

生産、流通活動において、大規模な物流センターが普及してきている

物流センターが抱える問題

多頻度納品

リードタイム増加

作業員増加

多くの物流センターでは自動倉庫を導入して解消を狙っている

1．2　自動倉庫

高さ，水平方向に物品を格納するセルが並んでいる巨大倉庫

スタッカークレーン(以後SCと略記)という昇降装置が自動に物品を運搬

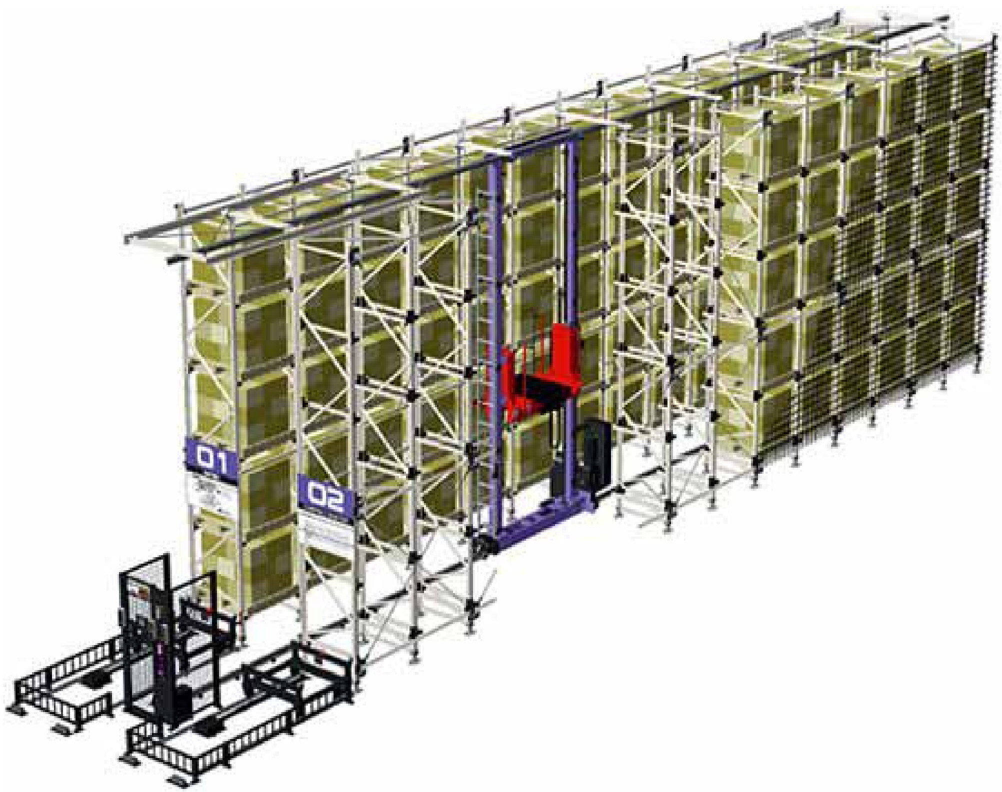


図1.**立体自動倉庫**

出所：http://www.daifuku.co.jp/products/storage/cs/feature.html

メリット

- **数多くの物品を、コンピューター制御で入出できる**
- **人員削減、入出作業のミスが軽減、作業が早くなる**

1．3　自動倉庫おける問題点

◆倉庫内に入出庫すべき物品は**多数**

◆SCが一度に運べる物品数は**たかだか数個**

(ex.SCの容量3個、入出庫すべき物品120個)

倉庫での作業時間が大きいと
リードタイムが増加してしまう

能率的に入出庫作業を行う必要がある

1．3　自動倉庫における問題点

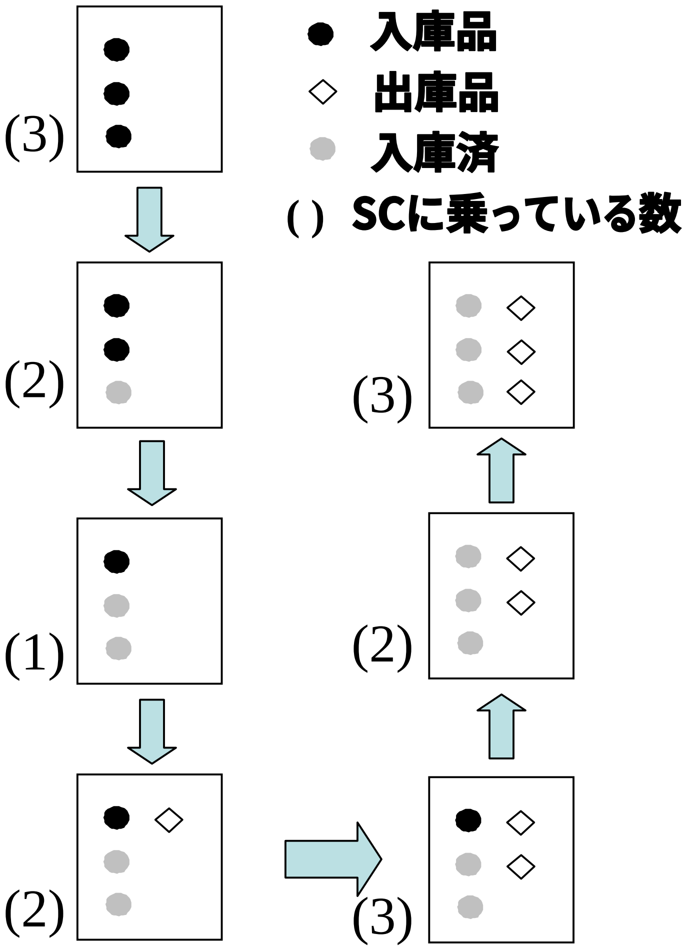


図2. SCに乗せられる物品数

(SC容量=3の場合の例)

能率的な入出庫作業

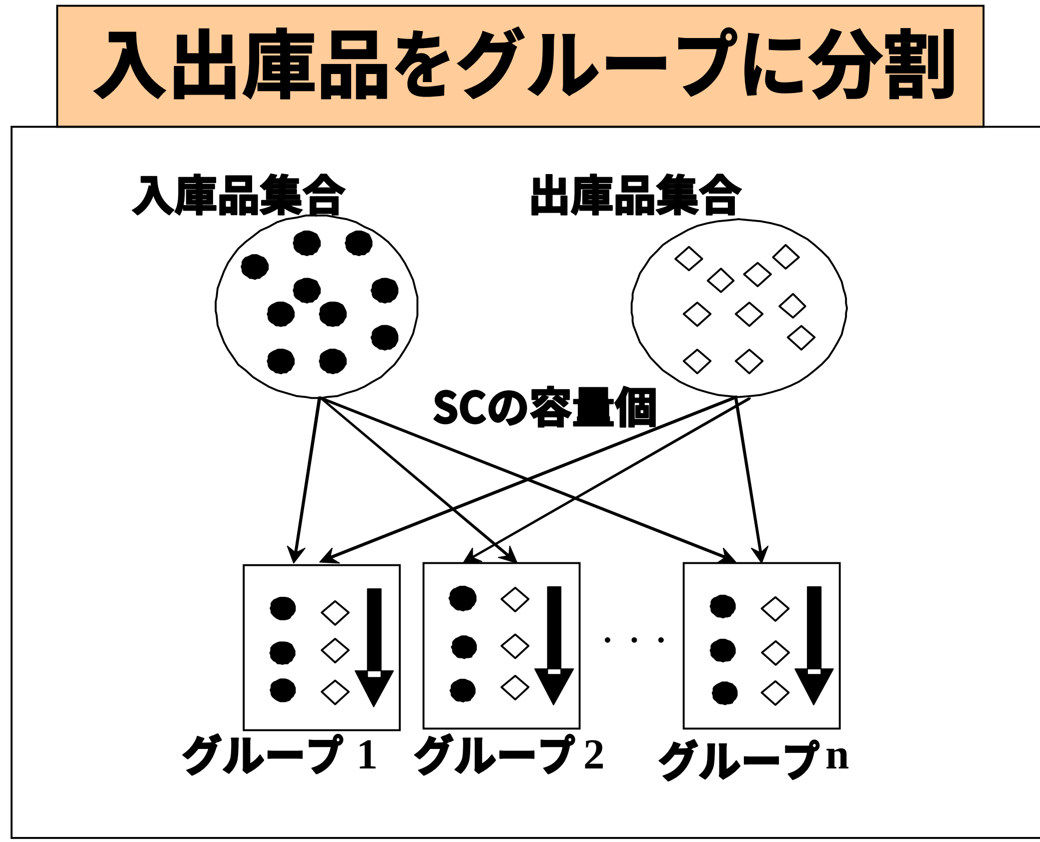


図3. グループ分割

(ＳＣ容量=3の場合)

能率的な入出庫作業

グループ内で巡回路を決定する

どういう順序で巡回すると入出が早く済むか

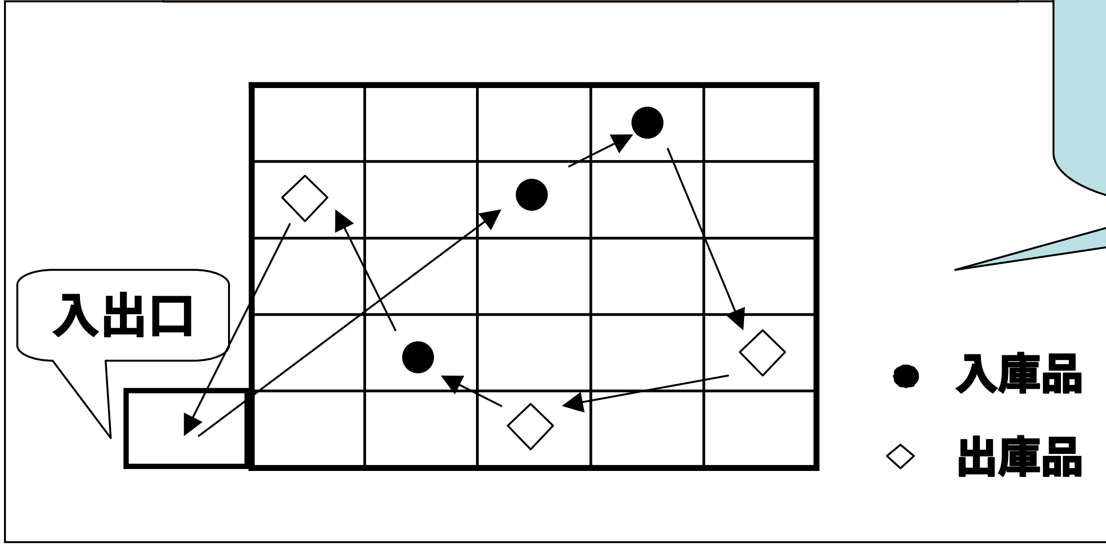


図4．巡回路決定(ＳＣ容量=3)

能率的な入出庫作業とは

入出庫品をグループに分割

＋

グループ内で巡回を決定する

＝　入出庫スケジューリング問題

1．4　先行研究とその問題点

この問題題を取り上げた先行研究として文献[1]が挙げられる

↓

なるべく短い時間で仕事を終える入出庫スケジュールを作成する近似アルゴリズムを提案している．

文献[1]

・数理計画としては定式化しておらず

解空間の大きさについて述べているだけである

⇒ **数理計画としての定式化を行いたい．**

・入庫品の順序だけを変化させて入出庫スケジュール作成

⇒ **解法に改善の余地がある．**

1．5　本研究の目的

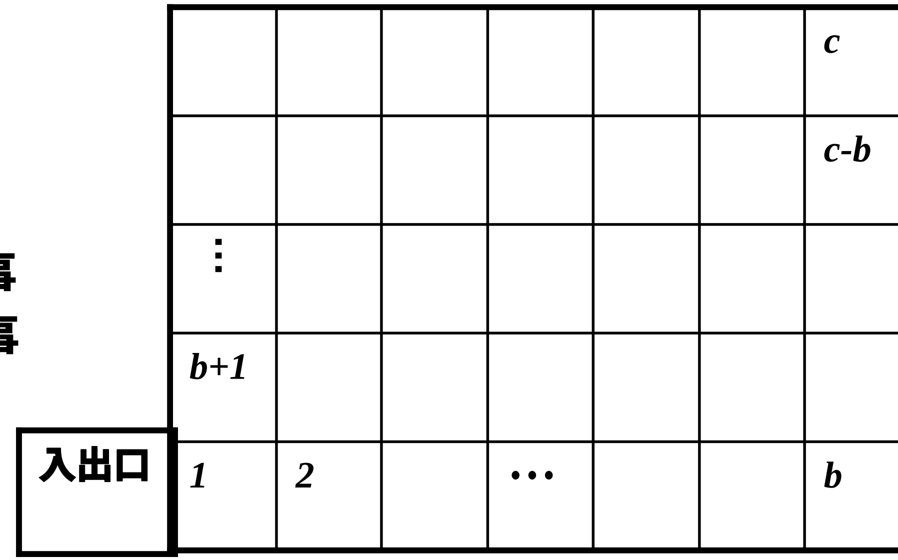
(1) 数理計画としての問題の定式化

(2) 出庫も考慮した新しい解法の提案

→文献[1]の解法との比較

2.　問題設定

- 高さ*a*個、水平方向に*b*個の全部で*c*個(=*a* ×*b*)のセル
- 入庫･･･ＳＣから物品を倉庫内に降ろす事
  出庫･･･倉庫内から物品をＳＣに載せる事
- 仕事･･･入庫, 出庫作業の事でセル番号で与えられる
- 巡回･･･SCが入出口から入り、いくつかの仕事先を訪問して入出口に戻る動作
- 入出品は物品ごとに、格納されるセルが指定されている



入力項目＜入出庫仕事集合＞

●入庫仕事	●出庫仕事
2	*26*
13	*56*
…	…
1	*17*

図5. 倉庫のセル番号と入力項目の例

SCの移動とそれにかかる時間について

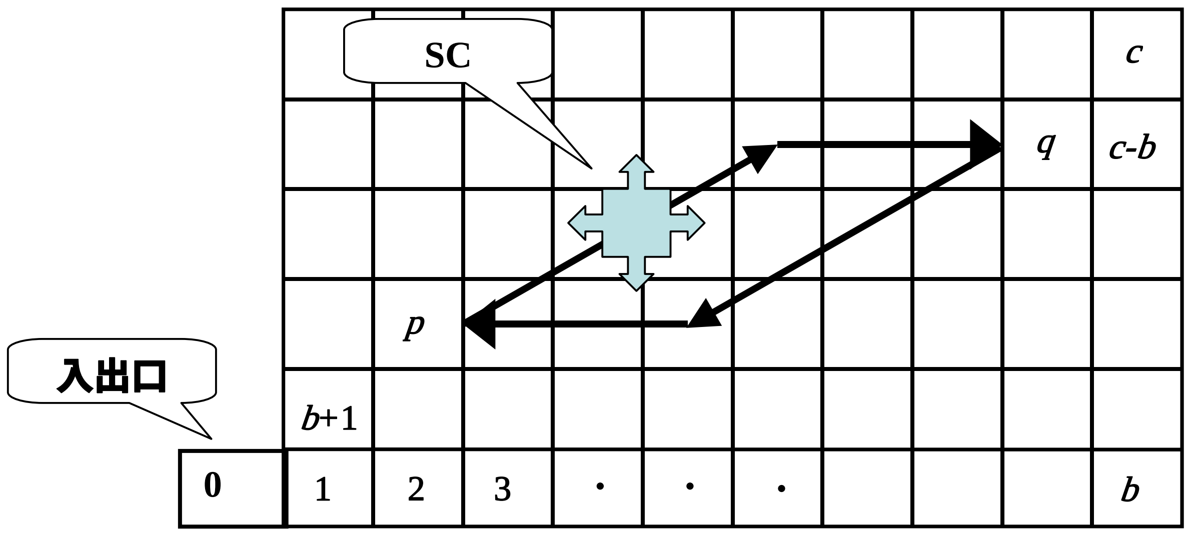


図6. pからqへのＳＣの移動

水平方向と高さ方向に独立に並列して移動できる

SCの移動に掛かる時間は

水平方向移動，高さ方向移動のどちらか大きくなる方となる

3．1　記号の定義

M : 入庫仕事数（＝出庫仕事数）

m : SCの容量

n : 総巡回数(グループ数)　$\left(= M/m\right)$

0 : 入出口のセル番号

i : 仕事番号　$\left(\overset{\text{入庫}}{1,\ldots,M},\quad \overset{\text{出庫}}{M+1,\ldots,2M}\right)$

j : 1巡回のSCの仕事番号　$(1,\ldots,2m)$

k : 巡回番号　$(1,\ldots,n)$

t_{ih} : 仕事iから仕事hへの移動時間

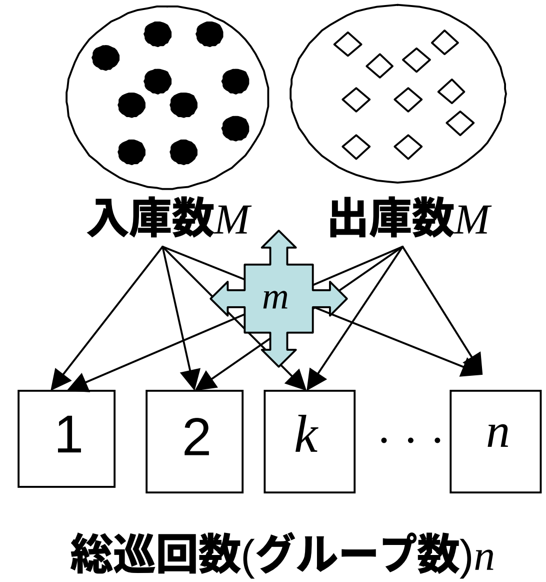


図7. グループ分割における記号の定義

$$x_{ijk}\begin{cases}1: \text{第}i\text{番目の仕事を、第}k\text{回目の巡回で}j\text{番目に処理する.}\\ 0: \text{それ以外}\end{cases}$$

3．2 目的関数

●全ての仕事を処理するのに要する移動時間の最小化

n回繰り返す

入出口から、1つ目の仕事への移動時間

＋

1つ目～2m個目の仕事への各間の移動時間

＋

2m個目の仕事から入出口への移動時間

1度の巡回にかかる時間

$$\sum_{i=1}^{2M} t_{0i} \cdot x_{i1k} + \sum_{j=1}^{2m-1} \sum_{i=1}^{2M} \sum_{h=1}^{2M} t_{ih} \cdot x_{ijk} \cdot x_{h(j+1)k} + \sum_{i=1}^{2M} t_{i0} \cdot x_{i2mk} \quad (k = 1, \ldots, n)$$

３．３　制約条件

全ての仕事は必ず処理される

$$\sum_{j}^{2m}\sum_{k}^{n} x_{ijk} = 1 \quad (i = 1,\ldots,M, M+1,\ldots,2M) \qquad (1)$$

一度の巡回の, 第j番目に処理される仕事は一つ

$$\sum_{i}^{2M} x_{ijk} = 1 \quad \begin{pmatrix} j = 1,\ldots,2m \\ k = 1,\ldots,n \end{pmatrix} \qquad (2)$$

一度の巡回で処理される仕事は$2m$個

$$\sum_{j=1}^{2m}\sum_{i=1}^{2M} x_{ijk} = 2m \quad (k = 1,..,n) \qquad (3)$$

第j番目までの入庫数は出庫した数以上であること

$$\sum_{l=1}^{j}\sum_{i=1}^{M} x_{ilk} \geq \sum_{l=1}^{j}\sum_{i=M+1}^{2M} x_{ilk} \quad \begin{pmatrix} k = 1,\ldots,n \\ j = 1,\ldots,2m \end{pmatrix} \qquad (4)$$

決定変数

$$x_{ijk} \in \{0,1\} \qquad (5)$$

3．4 定式化のまとめ

問題 SCsc

$$\min \quad TRT = \sum_{k=1}^{n}\left(\sum_{i=1}^{2M} t_{0i}\cdot x_{i1k} + \sum_{j=1}^{2m-1}\sum_{i=1}^{2M}\sum_{h=1}^{2M} t_{ih}\cdot x_{ijk}\cdot x_{h(j+1)k} + \sum_{i=1}^{2M} t_{i0}\cdot x_{i2mk}\right) \quad (3.1)$$

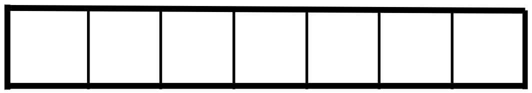
◎制約条件

3．3の(1)〜(5)

4．　文献[1]の解法

M個からなる入庫仕事のリスト だけを解として表現

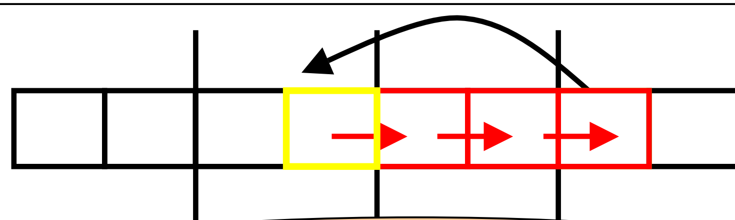
入庫仕事のリスト



入庫仕事の順序に対応した総巡回時間TRTを持っている.

入庫仕事の頭から順に, 仕事のセル番号の直近となるような出庫仕事をグルーピングしてスケジュールを作成している.

入庫仕事の順序を変更し、TRTを更新



最も小さいTRTとなった入庫仕事のリストの入出庫スケジュールを出力

文献[1]の解法の問題点

入庫仕事だけでスケジュールを作成

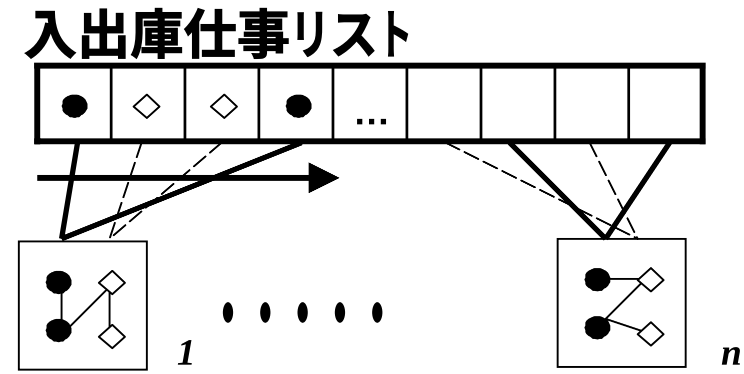
出庫仕事を考慮していない

→ **解の精度において不十分**

5．　提案する解法

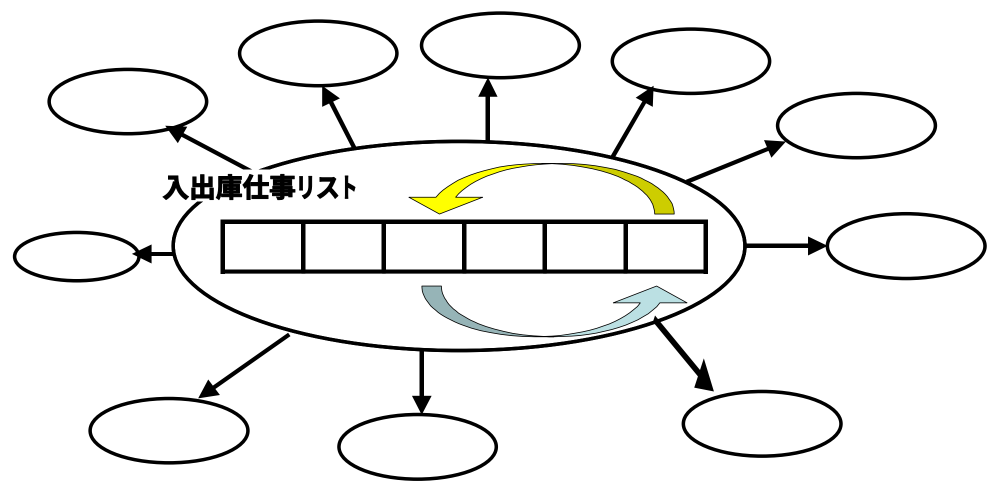
◎$2M$個からなる入出庫仕事リストを解として表現する．

入庫仕事からm個ずつ，出庫仕事からm個ずつ取り出していき，n個のグループを作る．



各グループ内での最短となる巡回路の厳密解を求める．

入庫仕事，出庫仕事両方からスケジュールを作成している．



リストの順序が入れ替わる度に，異なるTRTが得られる．

解法の流れ

1. 任意の二つの仕事の入替えを全通り行う.

2. 各入替え毎に*TRT*を求める.

3. 最も*TRT*の小さくなった順序へ移動する.

4. 1へ戻る.

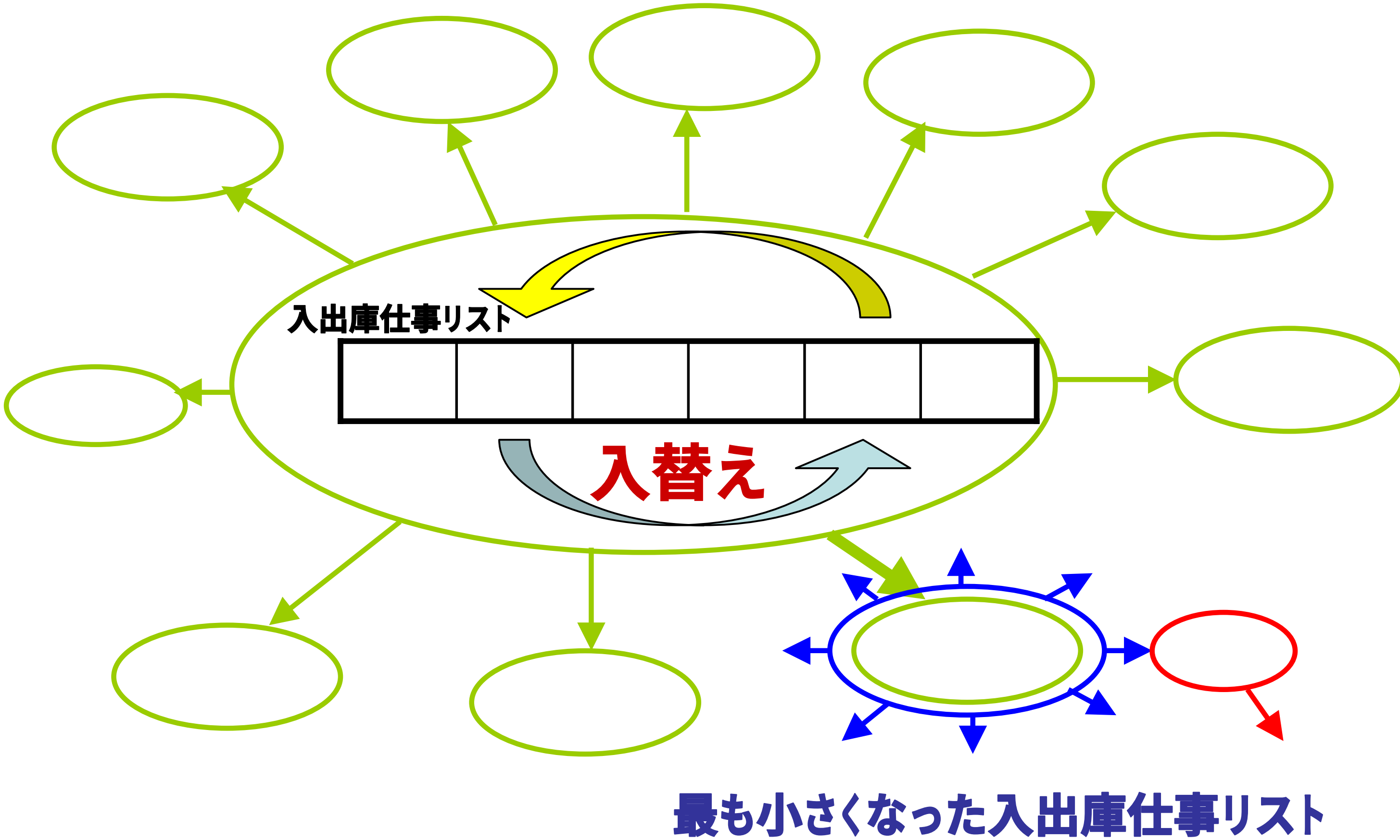


図8. 提案する解法

解法の流れ

*TRT*の値が小さくならない入出庫仕事リストに辿り着いたら

↓

総入出庫時間を短くするスケジュールを作る事が出来た．

↓

探索を終了する．

6．数値実験

6．1実験概要

文献[1]の実験と同じ条件で実験

- 高さ方向に20段、水平方向に50段の1000個のセルからなる倉庫
- SCの容量は3
- 入庫、出庫仕事数はそれぞれ30

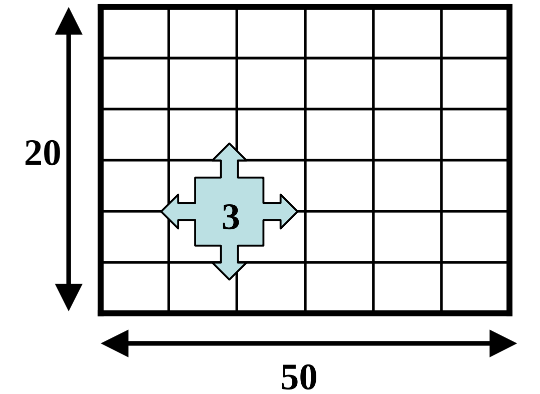


図9．実験に使う倉庫

●この条件で100個の例題をランダムに生成

与えられたリストの順番で行った場合，文献[1]，提案した解法で行った場合それぞれを実験．

プログラムはBorland社のDelphi6で作成

6．数値実験

6．2計算結果

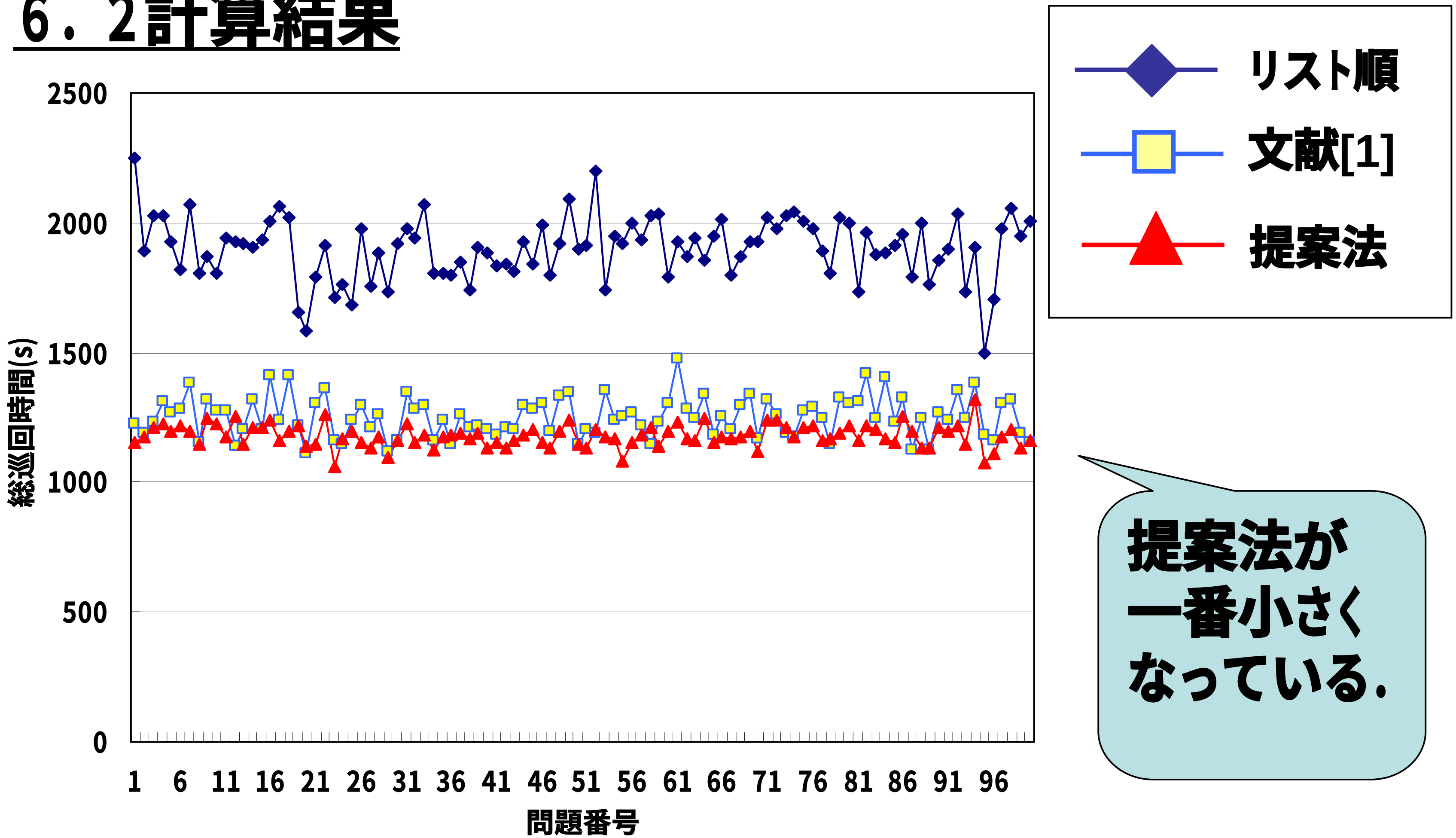


図10．100例題による各解法の総巡回時間

提案法が一番小さくなっている．

6．数値実験

6.2計算結果と考察

表1　各解法の平均総巡回時間の改善率

解法 / 実験結果	リスト順	文献[1]の解法	提案した解法
平均総巡回時間(s)	1901	1252	1179
改善率(%)		34	38
平均計算時間 (s)	0.01	50	44

4%（約80秒）の改善
→1回の入出庫スケジュール毎に80秒毎の短縮
スケジュール作成が続くことで時間短縮に繋がる.

7．まとめ

・入出庫スケジューリング問題を改めて定式化した．

・出庫品も考慮した解法を提案し，文献[1]での解法よりよい解を得ることが出来た．

8. 今後の課題

・近傍の探索に無駄がある．

・局所的な最適解で停止している．

局所的な最適解から脱出する方法を組み込むことでよりよい解を発見できる可能性がある．

参考文献

[1]胡貴彦、木瀬洋、徐悦東：
「立体自動倉庫における入出庫スケジューリングの最適化」
システム制御情報学会論文誌 vol.18,No4(2005)

[2]柳浦睦憲、茨木俊秀：
「組合せ最適化 ーメタ戦略を中心としてー」
朝倉書店(2001).

[3]山本芳嗣, 久保幹雄：「巡回セールスマン問題への招待」
朝倉書店, 1997.

訂正

p.119の探索アルゴリズムのStep3

(誤)　t←t+1

(正)　削除

p.120の主要参考文献

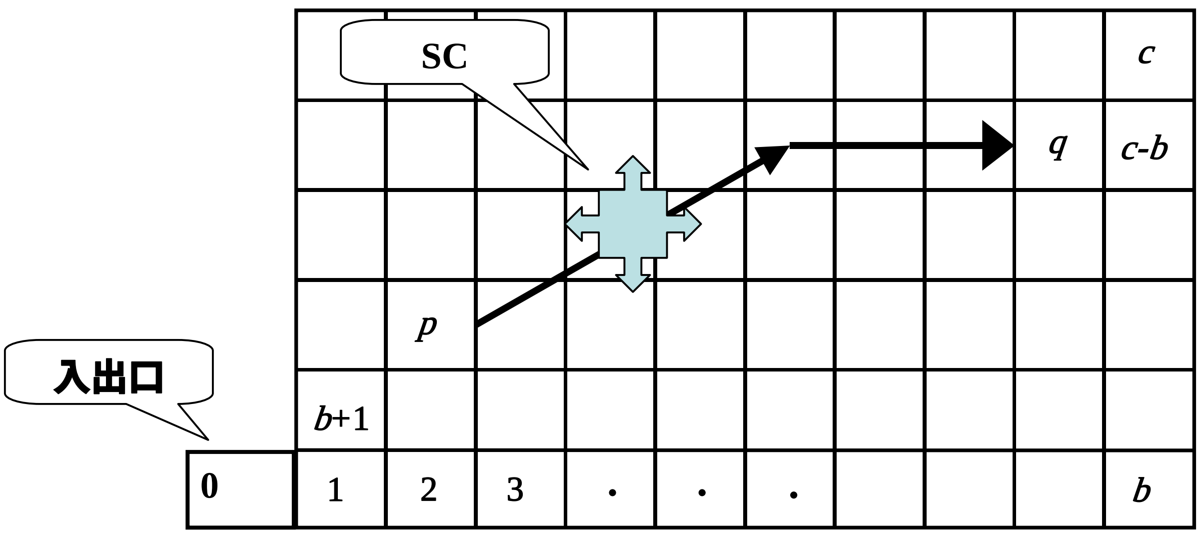
(誤)　Guiyan Hu，Hiroshi Kise，Yuedong Xu：“Optimization for Input/Output Scheduling for Automated Warehouses”，システム制御情報学論文誌Vol.18，No.4，pp.156-163，2005.

(正)　胡貴彦、木瀬洋、徐悦東：「立体自動倉庫における入出庫スケジューリングの最適化」システム制御情報学会論文誌 vol.18，No4(2005)

Appendix

SCの移動とそれにかかる時間について

p,q間での移動にかかる時間と距離の例



図A-1. pからqへのＳＣの移動

p,q**間の移動距離**

$$d(p,q) = \max\{d_X(p,q), d_Y(p,q)\}$$

p,q**間の移動時間**

$$t(p,q) = \max\{t_X(p,q), t_Y(p,q)\}$$

(チェビシェフ測度)

記号の定義

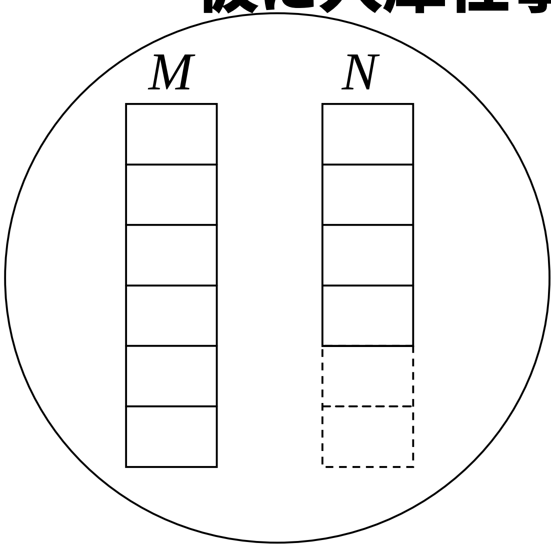
一般性を失うことなく

M=*N*=(*n* ‘*m*)とする．

$$n = \left(\lceil M/m \rceil, \lceil N/m \rceil\right)$$

ダミー仕事について

仮に入庫仕事Mと出庫仕事Nの数が異なるとした場合



足りない仕事をダミー仕事とし，

その訪問先を入出口とする

問題SCscの解空間(1)

問題SCscは仕事の組合せを決定する問題

↓

全ての組合せを調べれば最小化できる！

問題SCscの解空間(2)

◎M個ずつの入出庫品をそれぞれm個ずつのグループに分ける場合の異なるスケジュール数を $g(M,m)$ とすると

$$g(M,m)=\left\{\frac{M!}{(m!)^n n!}\right\}^2 n! \quad (4.1)$$

◎2m個のセルの巡回中どの段階でも入庫済数が出庫済数以上である時の実行可能なルーティング数を$r(m)$とすると

$$r(m)=\frac{{}_{2m}C_m}{m+1}(m!)^2=\frac{(2m)!}{m+1} \quad (4.2)$$

従って、実行可能なスケジュール数 $s(M,m)$ は

$$\boxed{s(M,m)=g(M,m)\times r(m)=\frac{(2m)!(M!)^2}{(m+1)n!(m!)^{2n}} \quad (4.3)}$$

問題SCsc の解空間(3)

4.3式に表した数のスケジュールが存在する

$$\frac{(2m)!(M!)^2}{(m+1)n!(m!)^{2n}} \quad (4.3)$$

Mの数が増加するにつれて，指数関数的に増加する．

全列挙によって最適解を求めることは困難である！

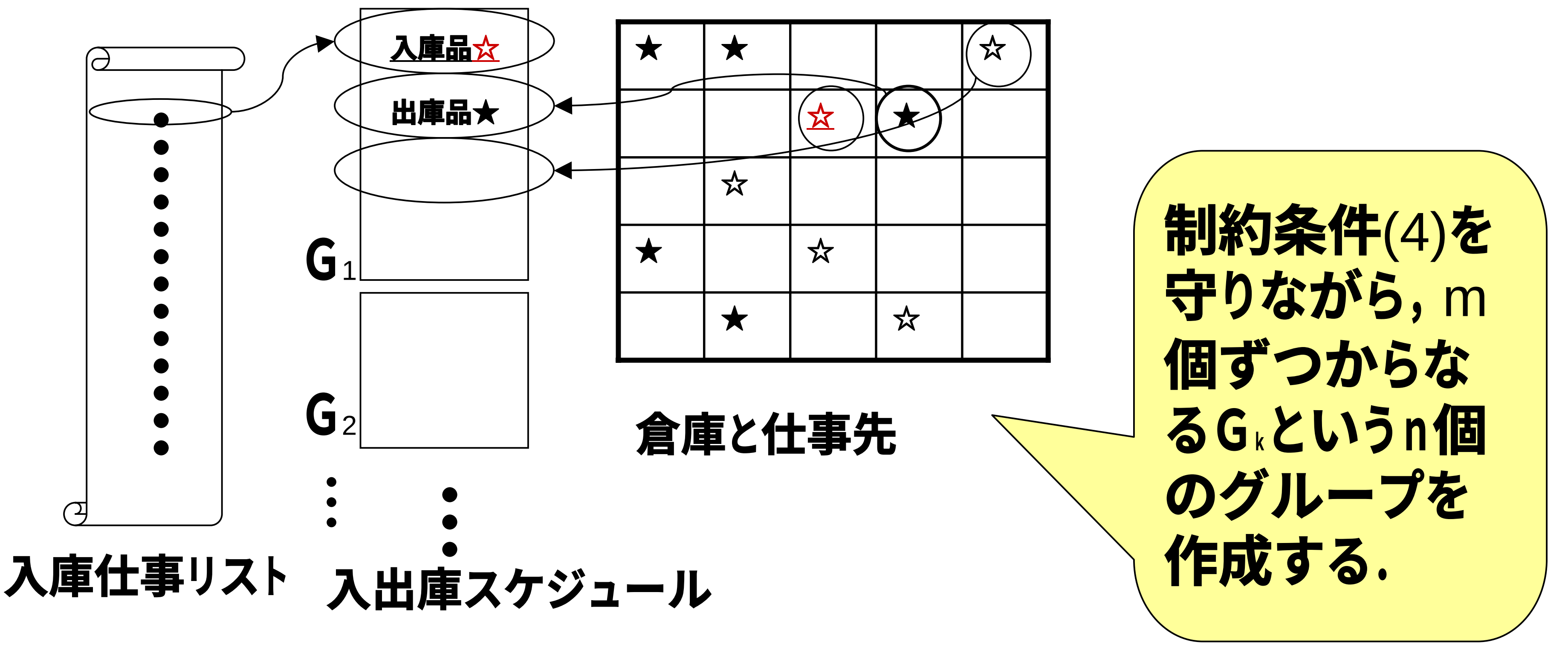
今回の実験例だと異なるスケジュール数は 9.55×10^{44}

近似解法が必要

解法

文献[1]の解法

○入庫仕事リストの最初の仕事をスケジュールに取り入れ，そのセルの**直近**のセルにリストの物品があれば取り入れていく.

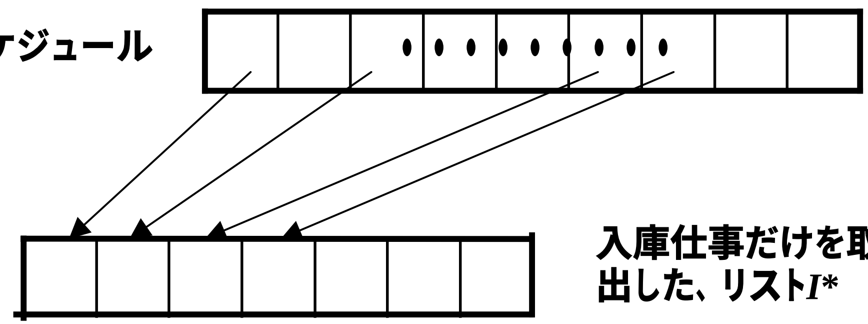


図A-4.文献[1]の解法

先行研究の解法

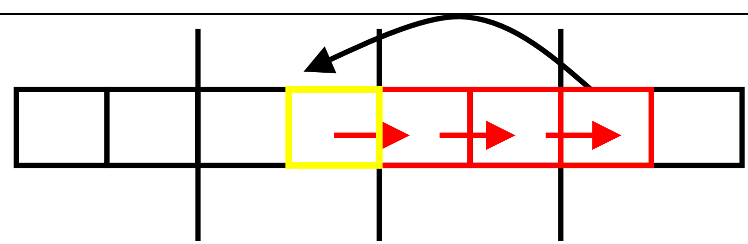
こうして作られた入出庫スケジュールから，
入庫仕事*I*だけを先頭から順に取り出す．

今作ったスケジュール



入庫仕事だけを取り出した、リスト*I**

この取り出したリスト*Iについて，入庫品の少なくとも1つが初期スケジュールでグループの先頭だったものは先頭でなくなるように，先頭でないものが先頭になるように挿入を行う．**



一つにつき210通りの近傍がある

文献[1]の解法手順

①初期解作成

Step1. 入庫リスト$I=\{i_1,\dots,i_M\}$, 出庫リスト$O=\{o_1,\dots,o_M\}$,
仕事グループ$G_k=\{G_1,\dots,G_n\}(k=1,..,n)$,
U_I,U_OをそれぞれI,Oから選び出した回数とする.

Step2. 入庫リストIの先頭要素をpとし, $G_k\leftarrow\{p\}$とする.
pの直近の要素を, $U_O<U_I<m$なら$I\cup O$から, 違うなら
Iの中から選び, それをeとし, $G_k\leftarrow G_{k+1}\cup\{e\}$とする.

Step3. これを$U_O=U_I=m$になるまで繰り返すと
1つのグループが出来上がるので
それをさらにnグループ分繰り返す.

文献[1]の解法続き

出来上がった初期リストに対して、近傍解の探索を行う

初期リストから入庫リストIだけを取り出し，I内の要素の移動によるリストの変化で，近傍解を作成する．

◎挿入法による要素の移動

その際，移動前に先頭要素だったものは少なくとも一つは先頭要素にならないよう，先頭要素でないものは先頭要素になるように生成する．

出庫リストは考慮されていない！

タブーサーチ（ＴＳ）について

◎局所探索法の一種

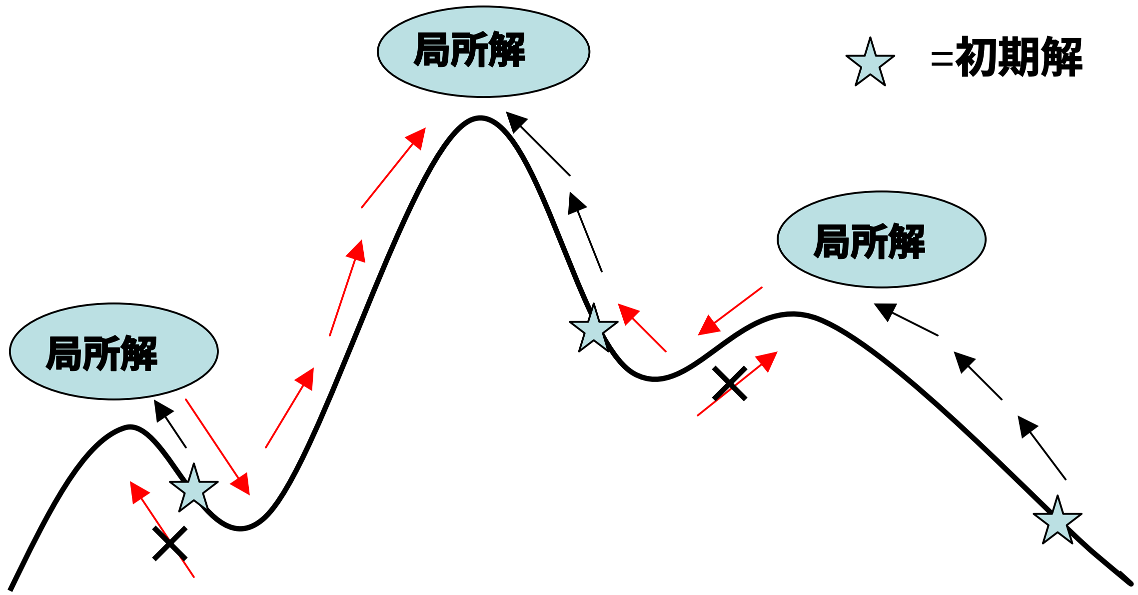


図6.ＴＳの動き

局所解でトラップされず，改悪を許すことによって最適解に巡りあう可能性を高くした方法．

局所最適解から下り勾配の小さい方に移動するがこの時同じ局所最適解に戻るのを禁止するため，一度巡った順序の選択は禁止される．