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

野村　要 (沼田　一道　助教授)

１．はじめに

１．１．背景

我々が日常で使用するスーパーや，百貨店などに並ぶ物品は，生産場所から物流センターを経由して各店舗へと輸送されているものが多い．そのため物流センターには大量の物品が頻繁に運ばれてくる．そして，運ばれてきた物品は一時的に保管される必要があるが，最近の物流センターでは自動倉庫を採用してこれに対処している．自動倉庫とは，格納庫の集合でできた巨大な倉庫である．この自動倉庫は高さ方向，水平方向に格納庫が棚状に並んでいる．大量の物品の入出作業を人手で行うことは難しいので，これを全自動で行うスタッカークレーン(Stacker Crane，以後 SC)を一つだけ備えている．SC は無人で，与えられたリストにそって物品の入出を行う．その際，入出すべき物品が多数あるのに対し，SC が一度の巡回に持ち運べる物品はたかだか数個に限られるため，SC は多数回の巡回を繰り返さなければならない．そのため，一度に扱う物品をうまくグルーピングするなどして，能率的に作業を行う必要がある．この入出庫作業計画を最適化問題として定式化し，所要時間の小さい作業計画を求めることにより，リードタイムを短縮するなどのコスト削減に繋げることができる．

１．２．目的

本研究では，自動倉庫における，SC の入出庫のルーティングと入出庫品の組合せの同時最適化を行う問題を扱う．この同時最適化問題に対する近似解法は文献[1]で既に提案されているが，文献[1]では問題の定式化が不十分であり，さらに解法に改善の余地があるように思われる．本研究では，この問題を定式化し，新たな近似解法を提案する．

２．問題設定

２．１．自動倉庫の概要

本研究で取り上げる自動倉庫は, 立体棚の高さ方向に a 個, 水平方向に b 個と一つの入出口からなる．また，各格納庫（セル）には認識番号が与えられ，それぞれに対応した物品が格納されていく．入出すべき物品は，その認識番号を並べたリストで与えられる．入庫とはセルに物品を入れることで，出庫とはセルから物品を取り出すことであり，SC からセルに物品を降ろす作業，セルから SC に物品を載せる作業を仕事と呼ぶ．巡回とは，SC が入出口から入り，いくつかのセルを訪問し，入出口に戻るという動作を指す．SC の同時に保持できる品数を SC の容量と呼び，これを m と記す．一度の巡回で操作できる物品の数は入庫品と出庫品，それぞれ m 個ずつであり，M 個の入庫品，N 個の出庫品を入出庫するため $n = \max(\lceil M/m \rceil, \lceil N/m \rceil)$ 回の巡回を要する. 本研究では以後一般性を失うことなく, $M = N(=n \cdot m)$とする．ここで，M 個の入庫仕事と M 個の出庫仕事をそれぞれ大きさ m の n 個のグループに分割し，この入庫グループと出庫グループを 1 対 1 でペアリングしたものを入出庫グループと呼ぶ. また, 各入出庫グループ中でのセルの巡回順序をルーティングと呼ぶ.

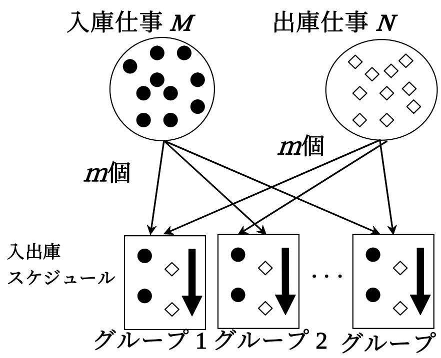

図 2.1　入出庫グループ(m=2)

２．２．入出庫スケジューリング問題

入出庫スケジューリング問題とは，入庫仕事と出庫仕事，SC の容量が与えられた上で，全仕事の総巡回時間が最小となる n 個の入出庫グループの分割と，それぞれに対する実行可能なルーティングを決定する問題である．ただし，SC は同時に m 個しか保持出来ない為，ルーティングの各段階において，入庫済み品数は出庫済み品数以上でなくてはならない．

３．定式化

2.2 節で述べた入出庫スケジューリング問題を数理計画問題として定式化し，問題SCscと呼ぶことにする．SCの最大積載量をm，入庫数，出庫数をそれぞれM，総巡回回数をn，入出口番号を 0，仕事番号を$I = (1, \ldots, M, M+1, \ldots, 2M)$とする．1~$M$は入庫仕事であり，$M$+1~2$M$は出庫仕事とする．1 巡回中の仕事の順番を$j = (1, \ldots, 2m)$，巡回番号を$k = (1, \ldots, n)$，仕事$i$から$h$への移動時間を$t_{ih}$とする．また，$x_{ijk}$は，第$i$番目の仕事を，第$k$回目の巡回で，第$j$番目に行うとき 1，それ以外は 0 をとる決定変数である．
問題 SCsc の定式化は以下のようになる．

(SCsc)

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

$$\text{s.t.} \quad \sum_{j=1}^{2m}\sum_{k=1}^{n} x_{ijk} = 1 \quad (i = 1, \ldots, M, M+1, \ldots, 2M) \tag{3.2}$$

$$\sum_{i=1}^{2M} x_{ijk} = 1 \quad (j = 1, \ldots, 2m, \quad k = 1, \ldots, n) \tag{3.3}$$

$$\sum_{j=1}^{2m}\sum_{i=1}^{2M} x_{ijk} = 2m \quad (k = 1, \ldots, n) \tag{3.4}$$

$$\sum_{l=1}^{j}\sum_{i=1}^{M} x_{ilk} \geq \sum_{l=1}^{j}\sum_{i=1}^{2M} x_{ilk} \quad (j = 1, \ldots, 2m, \quad k = 1, \ldots, n) \tag{3.5}$$

$$x_{ijk} \in \{0,1\} \quad (i = 1, \ldots, M, M+1, \ldots, 2M, \quad j = 1, \ldots, 2m, \quad k = 1, \ldots, n) \tag{3.6}$$

(3.1)式は目的関数で，総巡回時間TRTを表している．(3.2)式は，すべての仕事は必ずどこかで処理されなければならない制約条件を表している．(3.3)式は，一度の巡回の第j番目に処理できる仕事は一つだけであるという制約条件を表している．(3.4)式は，一度の巡回で処理される仕事は 2m個であるという制約条件を表している．(3.5)式は，第j番目までの入庫数は出庫数以上であるという制約条件を表している．(3.6)式はx_{ijk}は 0 か 1 であるという制約条件を表している．

４．解法

入出庫仕事の組合せは，全部で$(2M)! \times n \times (2m)!$通りであり，実行可能ルーティング数を考慮しても，入出庫スケジュールの組合せ数は$(2m)!(M!)^2 / (m+1)n!(m!)^{2n}$ 通りとなる．組合せを全列挙することにより，問題 SCsc の最適解を得ることができるが，当然ながら，M の値が大きくなると全列挙は困難である．このような問題に対処する方法として，最適性の保証はないが良質の解をできるだけ早く求めようとする近似解法がある．

４．１．文献[1]の解法

文献[1]は，入庫仕事の順序を，解に対応させる方法を採用している．入庫仕事の順序はそれに対応した解の総巡回時間 *TRT* を与える．ある入庫仕事順序の近傍は，仕事の位置を変更して出来る入庫仕事の順序全体の集合である．近傍内の各順序に対応した解の *TRT* を計算し，最も小さい *TRT* の順序に移動する．この近傍探索を繰り返し，*TRT* の最も小さくなった解の入出庫スケジュールを得る解法である．しかし，この解法は，入庫仕事しか考慮していないので，精度において不十分であると考えられる．

４．２．本研究で提案する解法

本研究では入出庫スケジュールそのものを解とし，この順序を入れ換えたものを近傍解とする．
使用する解 x は以下の手順で求める．

TRT の評価手順

Step1：k=0,　I を入庫リスト，O を入庫リストとする

Step2：$k = n\left(=\lceil M / m \rceil\right)$なら終了する．そうでなければ $k \leftarrow k+1$ として，入庫リスト，出庫リストからそれぞれ m 個ずつ抽出していき，入庫グループ，出庫グループをペアリングして一つの入出庫グループを作成する．

Step3：各グループ内でルーティングを全数探索し，最小巡回時間を与えるようなルーティングを決定する．そして，Step2 へ戻る．

生成された解をもとに以下のアルゴリズムを適用する．解法の手順は以下の通りである．

探索アルゴリズムの手順

Step1：近傍解集合内に暫定解より小さくなる解がなければ探索を終了し，Step4 へ．

Step2：解の組合せリスト内の物品の移動による近傍探索を行う．組合せリスト内のどれか二つの位置を入れ替え，元と異なる組合せを生成する．

Step3：移動した近傍解をx'とし，暫定解$W = \min\{x, x'\}$とする．$t \leftarrow t+1$．Step2 へ戻る．

Step4：準最適解 W が得られるときの組合せのリストを入出庫スケジュールとして出力する．

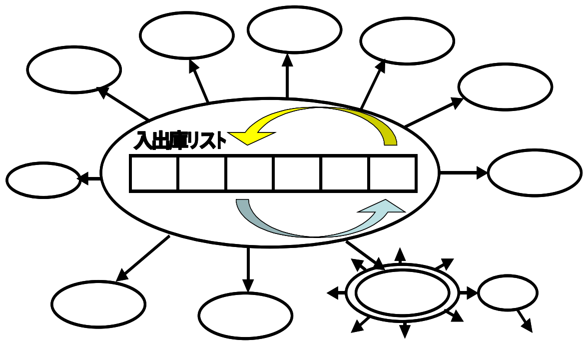

図 4.1 近傍解集合

５．数値実験

５．１．概要

4.2 節で述べたとおり，入出庫グループのリストを局所探索し，各グループのルーティングを厳密に求めることで，総巡回時間を最小にする．データはセル数が高さ方向に 20 段，水平方向に 50 段の 1000 個の倉庫，SC容量は 3，入庫仕事数 30，出庫仕事数 30，各仕事の訪問先はランダムに与え，問題を 100 個生成した．仕事間の移動時間t_{ih}は，各セルの大きさを高さ，幅共に 1.5m，SCの最高速度は，高さ方向に 0.5[m/s]，水平方向に 1.67[m/s]，加速度は 0.3[m/s^2]で一定として計算した．文献[1]の解法と，本研究で提案した解法を用いて，入出庫スケジューリング問題を解き，計算結果の比較を行った．プログラムはBorland社のDelphi6 で作成した．

５．２．実験結果および考察

実験結果を図 5.1 に示す．リスト順とは，与えられた入出庫リストを並び替えせずにそのまま順に入出作業を行うことを指している．文献[1]の解法に比べて，近傍探索の範囲を広げた結果 100 個の問題中 85 問について提案した解法の方が総巡回時間を小さくすることができた．平均して，リスト順に入出庫作業を行うより約 40%(約 800 秒)の短縮となり，更に文献[1]の解法より約 4%(約 80 秒)ほど短縮できている．これより，文献[1]の解法よりよい解を与える解法を得ることが出来たと言える．実際に自動倉庫を操作する際には，入出庫リスト操作は断続的に与えられるので，毎操作ごとにこの約１分の短縮が続くことで，大幅な時間短縮に繋がることが考えられる．

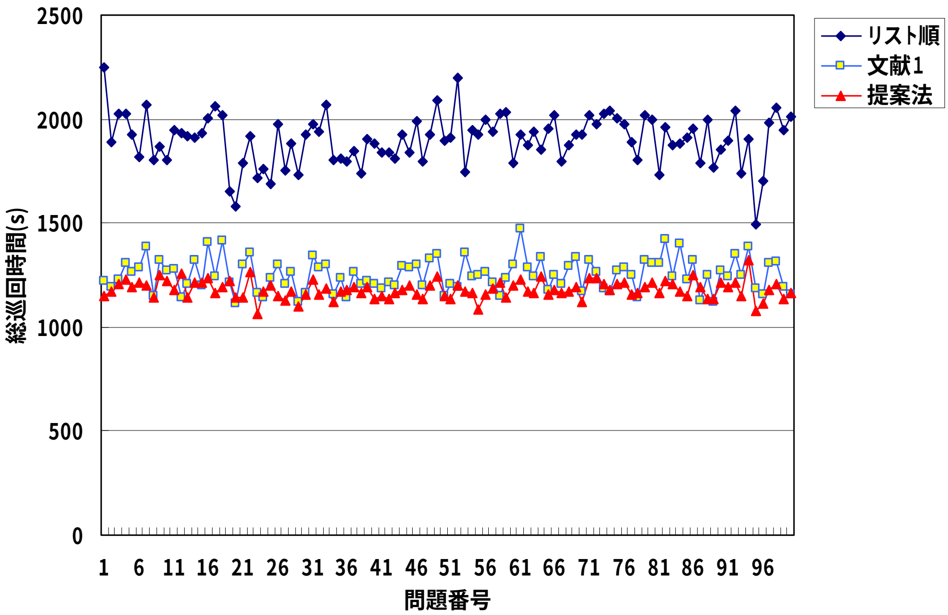

図 5.1　100 問題による各解法の総巡回時間

６．まとめ

本研究では，文献[1]で取り上げられた自動倉庫における入出庫スケジューリング問題を取り上げ，まず，問題を改めて定式化した．次に，文献[1]の解法では，入庫の順序のみからスケジューリングを行い，かつ分割した各グループ内でのルーティング方法は一つしか考慮していなかったことに着目し，新たに解法を提案した．提案した解法では入庫と出庫両方からスケジューリングを行うようにした．さらに，各グループ内でのルーティングは，全てのパターンを試すことで最小巡回時間となるようにした．この結果，扱う解の空間が広がり，文献[1]の解法で探索しきれていなかったよりよい解を探し出すことができ，文献[1]の解法よりよい解を与える解法を得ることが出来た．

今後の課題として，現在の解法では近傍解集合内に総巡回時間を更に短くする解がなくなると，探索を終了してしまうので，解の改悪を許し局所的な最適解を脱出する方法を組み込むことが必要であると考えている．

主要参考文献

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

[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.