自動倉庫の入出庫スケジューリング問題に対する厳密解法

久保田　善経　（沼田　一道　准教授，田中　健一　助教）

1　はじめに

物流センターでは，短時間でより多くの品物を正確に処理する必要がある．そのため，大規模な物流センターでは自動倉庫を導入して入出庫作業の効率化を図っているところが多い．自動倉庫は，平行に配置された複数の立体棚と，立体棚間の通路にそれぞれ 1 台ずつ設置されたスタッカークレーン (Stacker Crane，以下 SC) と呼ばれる搬送機から構成される．SC は高々数個の品物しか同時に積載できないので，入出庫すべき品物が多数ある場合には複数回に分けて処理を行う．このような SC 単体の運行計画を扱った入出庫スケジューリング問題 (Input/Output Scheduling Problem，以下 IOSP) に関しては様々な研究が行われてきた．文献 [2] ではこの問題を取り上げ，厳密解法を提案している．この解法 (以下既存解法) は，最適解であることを事後的に保証できる場合が多々あるというものであり，一般的な厳密解法とは異なる．また，最適性を保証するために SC の移動時間データを量子化するため，扱える問題サイズには求解時間とは別の制限が生ずる．本研究では，このような問題点を克服する方向で一般的な厳密解法の可能性を探ることを目的とする．

2　問題設定

2.1　本研究で扱う自動倉庫

自動倉庫は図 1 のように 1 つの立体棚と 1 台の SC から成るものとする．立体棚は多数のセルと 1 つの入出庫口で構成される．SC は入出庫口で入庫品を載せて出発し，運搬中に入庫品をセルに収納し，空いたスペースに出庫品を載せて入出庫口に戻るという巡回を繰り返す．本研究では，各入庫品を収納する場所が指定されている保管場所指定方式を扱う．

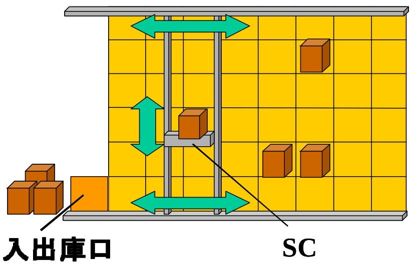

図 1：自動倉庫の例

2.2　IOSP

入出庫品がそれぞれ M 個ずつ与えられるとする．SC に同時に積載できる品物の個数を容量と呼ぶと，容量 m の SC は高々 m 個の品物しか同時に積載できないので，複数回に分けて巡回を行う必要がある．また，より少ない巡回回数で品物を運ぶことで稼働時間の節約ができるため，全ての入出庫品を処理するのに必要な最小回数の巡回を考えれば十分である．SC は毎回 m 個の入庫品と m 個の出庫品を処理できるので，合計 M/m 回の巡回がこれを実現する．IOSP はそのような巡回の仕方全体の中で，SC の総巡回時間を最小化させる運行スケジュールを決定する問題である．ただし，入出庫品の個数は等しく，M/m は整数と仮定する．これは問題を扱いやすくするためもので，ダミー仕事を補うことにより一般性を失うことなく仮定できる．

3　既存解法 [2]

保管場所指定方式の場合, IOSP は容量制約付き配送経路問題 (Capacitated Vehicle Routing Problem, 以下 CVRP) の一種として捉えることができる．CVRP とは，積載量の制限を持つ複数台の車両を用いて，全ての顧客の要求を満足する配送経路をコストが最小になるように決定する問題である．文献 [2] では IOSP を CVRP の一種として扱っている．CVRP は 2 つの問題を同時に考慮する問題である．1 つは入出庫品を同一の巡回で処理する品物ごとにグループ分けすること，もう 1 つはその分けられたグループごとに巡回経路を決定することである．

既存解法は，近似解 (上界値) の改善を繰り返し，それが最適であることを保証するために緩和解 (下

界値) の改善を行い両者を一致させることを目指すというアルゴリズムである．この解法は上下界値の一致を判定するために数値を量子化しているので，扱えるデータの規模が制限される．そのため，扱える問題サイズ (入出庫品の個数や SC の容量) が限られてくる．さらに，扱える問題サイズであっても，CVRP そのものが解き難い問題であるため，入出庫品の個数が多くなるにつれて上下界値が一致しなくなる場合が多くなる．そこで，文献 [2] では上下界値が一致しなかった場合の処理についても言及しており，得られた上下界値から最適解になり得ない変数を除去し，その後は汎用 MIP ソルバーを用いて最適解を求めるという方針を示している．つまり，既存解法は上下界値の更新に重きを置いているため，上下界値が一致しない場合については閉じた解法になってない．

4　提案法の基礎となる定式化

SC は小容量の搬送機であるため，グループごとの最適な処理順序は全列挙で短時間に求められる．つまり，IOSP は入出庫品のグループ分けを中心に扱える特殊な CVRP だといえる．そこで，本研究ではこの特徴を活かし IOSP に対し CVRP とは異なる定式化を行う．

入出庫品を容量 m 個ずつ割当てたグループ分け全体の集合を $L = \{L_1, L_2, .., L_p\}$ とする．グループ分け L_j の採用/不採用を表す決定変数を x_j，L_j を最短で処理する場合にかかる巡回時間を c_j，入出庫仕事 i が巡回 L_j に含まれるとき 1，含まれないとき 0 となる係数を a_{ij} として定式化すると

$$
\text{(SPP)}\left|\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{j=1}^{p} c_j x_j & (1)\\
\text{subject to} & \displaystyle\sum_{j=1}^{p} a_{ij} x_j = 1 \quad \forall i \in \{1, ..., 2M\} & (2)\\
 & x_j \in \{0, 1\} \quad \forall j \in \{1, ..., p\} & (3)
\end{array}\right.
$$

という集合分割問題 (Set Partitioning Problem) になる．ここで，(1) 式は SC の総巡回時間の最小化を表す目的関数，(2) 式は全ての入出庫要求は一度ずつ処理しなければならないという制約条件をそれぞれ表している．また，仕事の数 M の値が増加するとともに，グループ分けの数 p，つまり (2) 式左辺の係数行列の列数，および変数の数が指数的に増えることが (SPP) の特徴として挙げられる．

5　提案する厳密解法

5.1　解法の流れ

本研究で提案する厳密解法の概略は以下の通りである．まず，(SPP) の (3) 式を連続緩和した問題

$$
(\text{SPP}_{\text{LP}})\left|\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{j=1}^{p} c_j x_j & (4)\\
\text{subject to} & \displaystyle\sum_{j=1}^{p} a_{ij} x_j = 1 \quad \forall i \in \{1, ..., 2M\} & (5)\\
 & 0 \le x_j \le 1 \quad \forall j \in \{1, ..., p\} & (6)
\end{array}\right.
$$

を解き，上界値と下界値を計算する．求めた上下界値の情報から最適解になり得ない変数を除き，残った変数の中から間接列挙法により最適解を求める．ここで，上下界値の計算方針，及び問題規模の縮小は既存解法と同様であるが，本研究では SC の巡回経路は考慮せずにグループ分けのみを扱う．

5.2　下界値の計算

(SPP_{LP}) は巨大な列数の係数行列を持つため，(SPP_{LP}) を部分的に何度も解くことにより緩和解を求める列生成法を用いる．具体的には以下のステップを繰り返す．

Step0: $(\mathrm{SPP_{LP}})$ の実行可能解を含む適当な列集合を生成し，当該列から成る問題を $(\mathrm{SPP_{LP}})'$ とする．

Step1: $(\mathrm{SPP_{LP}})'$ を解き，$\{(\mathrm{SPP_{LP}})$ の列集合 $\}\backslash\{(\mathrm{SPP_{LP}})'$ の列集合 $\}$ から負の被約費用を持つグループ分けを探す．

Step2: 1 本以上見つかればそれらを $(\mathrm{SPP_{LP}})'$ に追加し，Step1 へ．見つからなければ現在の解が最適なので終了．

次に，緩和最適解を実行可能解に近づけるために制約条件を強化することを考える．このために導入する不等式制約をカットと呼ぶが，既存解法は CVRP に対するカットの生成をパッケージ化した CVRPSEP を用いて下界値の改善を図っている．これに対し，本研究ではグループ分けの特徴を活かしたカットを導入する．各変数はそれぞれグループ分けに対応しており，1 の値を持つとそのグループ分けを採用することを，0 の値を持つと採用しないことを意味する．また，$(\mathrm{SPP_{LP}})$ の解における各変数は 0 以上 1 以下の実数値をとる．ここで図 2 のように，i，j，k という 3 つのグループ分けが得られたとする．これらは同一の入出庫品を共有しているため，(SPP) の実行可能解として採用されるのは 1 つ以下である．すなわち，$x_i + x_j + x_k \leq 1.0$ というカットを得る事ができる．本研究では緩和問題を解く際の $(\mathrm{SPP_{LP}})'$ に上記のカットを追加し，得られる緩和最適値を下界値として用いる．

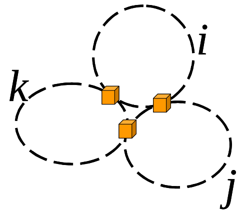

図 2：カットの例

5.3 上界値の計算

下界値の計算を行うとともに，その緩和解を利用して上界値の計算を行う．例えば，2 つの変数について，図 3 のように緩和解が得られたとする．まず，上の解の値を順回経路の各枝に割り当てる．他の変数と共有する枝を持つならば重複する値を合計し，それを枝の点数とする．さらに，変数が持つ枝の点数の総和を変数の点数とする．本研究では点数の和が最大の実行可能解を $(\mathrm{SPP_{LP}})'$ の解の中から探し，上解値の改善を図る．

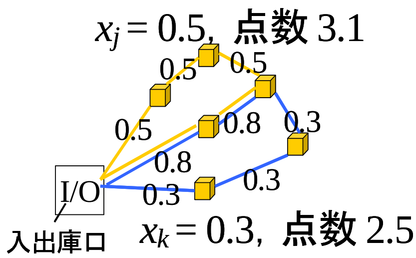

図 3：点数の付け方の例

5.4 釘付けテスト

緩和最適解において上下界値のギャップ以上の被約費用を持つ変数は最適解においてその値が 1 となることはない．つまり，求まったギャップよりも小さい被約費用を持つグループ分けのみを考慮すればよい．本研究では (SPP) の変数の数が多いため，最適解を求める前段階としてこの性質を利用し問題規模の縮小を図る．

5.5 間接列挙法に基づく厳密解法

規模縮小の後，残った変数の集合 X の要素から構成される (SPP) の実行可能解の中で，総巡回時間が最小となるものを探す．具体的には以下のステップを繰り返す．尚，値を 1 に固定した変数の集合を S，S の要素の変数に関係する入出庫品を含まない変数の集合を T と表す．

Step0: $S := \phi$，$T := X$，求めた上界値を暫定値とする．

Step1: $x_j \in T$ に対し $S := S \cup \{x_j\}$ とし，T を更新する．この追加により $\mid S \mid = M/m$ となれば Step5 へ．それ以外ならば Step2 へ．

Step2: T の要素で構成した緩和問題を解き, 緩和解の値と S の要素の総巡回時間の和を LB とし Step3 へ.

Step3: LB が暫定値以下か $T \neq \phi$ ならば Step1 へ．それ以外ならば Step4 へ．

Step4: 最後に追加した変数を S から除き $(S := S \setminus \{x_j\})$，$T$ を更新して Step1 へ．

Step5: S の要素の総巡回時間を計算し，その値が暫定値よりも小さければ更新する．全ての場合を (間接的に) 列挙したならば終了．それ以外ならば Step4 へ．

6 実験と考察

自動倉庫の諸元は文献 [1] に従う．立体棚は高さ方向に 20 段，水平方向に 50 連のセルから成るものとする．また，各セルの大きさは，高さ 1.50(m)，幅 1.50(m) とする．SC は水平，垂直方向に独立に移動可能で水平方向の最大速度，加減速度をそれぞれ，1.67(m/s)，0.30(m/s^2)，垂直方向の最大速度，加減速度をそれぞれ，0.50(m/s)，0.30(m/s^2) とし，SC の容量を $m = 2$ として実験を行った．また，プログラムは Delphi6 で作成し，$(\mathrm{SPP}_{\mathrm{LP}})'$，および間接列挙法で用いる連続緩和問題の計算は，GNU Linear Programming Kit Version4.9 を用いて解いた．例題としては文献 [2] で扱われた 60 問 (各 M ごとに 10 問) を用いた．既存解法の結果を表 1 に，提案法の結果を表 2 に示す．ここで，M は入出庫品の個数，p は変数の数，columns は生成された列の本数，cuts は生成された不等式制約の本数，UB は上界値の更新が行われた例題の数，opt は (上界値)=(下界値) となった例題の数をそれぞれ表す．また，本研究では上下界値を問題の規模縮小のために使用するので，opt ではなく pegging の項目を用意した．pegging は全変数のうち釘付けした変数の割合を表す．columns，cuts，pegging は 10 題の例題の平均を示している．

表 1：既存解法の結果

M	p	columns	cuts	UB	opt
10	16200	45.1	3.7	10	10
20	288800	147.5	107.7	10	9
30	1513800	250.9	271.6	10	4
40	4867200	347.4	488.5	10	6
50	12005000	474.5	687.0	10	5
60	25063200	591.3	816.3	10	4

表 2：提案法の結果

M	p	columns	cuts	UB	pegging
10	2025	61.1	18.3	6	0.97
20	36100	190.0	120.0	9	0.92
30	189225	343.8	223.2	7	0.86
40	608400	520.0	374.6	5	0.69
50	1500625	711.7	473.5	5	0.56
60	3132900	835.8	659.3	7	0.42

提案法は釘付け後に考慮すべき変数が 10000 程度残っている場合，現状では最適解を求めるのに約 2 時間程度かかる．そのため，$M = 30$ あたりからは，現実的時間内には解けない例題も現れる．さらに，$M = 60$ の場合，約 4 割の変数しか釘付けできないので，今後は更なる改良が必要である．既存解法の場合も，M の増加に伴い最適解が求まり難くなっていることが表 1 から読み取れる．但し，既存解法では釘付け後に汎用ソルバーを用いることで短時間に全ての例題に対し最適解を求めている．一方，提案法は最適解を得るまでに多くの時間を必要とする．これは，本研究において釘付けテストの効果が小さいためである．グループ分けをまとめて扱うよりも，各仕事間の枝の採用/不採用を変数として扱う方が，CVRPSEP の効果もあり上下界値の更新には適していると判断される．しかし，グループ分け中心の定式化は p の数が文献 [2] の 1/8 で済むため，釘付けテストに必要な時間は明らかに短縮される．IOSP を厳密に扱う際は，既存解法で上下界値を求め，その結果を利用し (SPP) に釘付けテストを施すことでより一層効果が得られるのではないかと考えられる．

7 まとめ

本研究では，文献 [2] で提案された既存解法の問題点を指摘し，独自の定式化を基本とする厳密解法を提案した．提案した方法は，扱った例題の範囲では既存解法の性能に遠く及ばないが，新しいアプローチの可能性を検証した点で意義がある．また，グループ分け中心の考え方は発見的解法との相性が良いので，本研究の結果を発見的解法の初期解生成に利用できる可能性があるが，その検討は今後の課題である．

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,No4,156/163 (2005)

[2] 田中俊二, 荒木光彦:”立体自動倉庫の入出庫スケジューリング問題に対する厳密解法”, 計測自動制御学会論文集 vol.42,No9,1058/1066 (2006)