東海大学紀要情報通信学部
Vol.11,No.2,2018,pp.21-26

論文

クロスドック入出荷口割当問題に対するタブー探索法

金高 健弥[*1]，浅江 季和[*2]，森山 弘海[*3]

A Tabu Search Heuristic for the Cross-dock Door Assignment Problem

by

Kenya KANETAKA[*1], Suetomo ASAE[*2] and Hiroumi MORIYAMA[*3]

(received on Sep. 28, 2018 & accepted on Jan. 10, 2019)

あらまし

クロスドッキングとは複数の供給点からクロスドック（物流センター）へ入荷した商品を在庫として保管することなく直ちに需要点別に仕分けて出荷する仕組みである．クロスドックには，通常，複数の入荷口と出荷口が設置される．そのため，クロスドッキングの運用においては，クロスドック内の輸送移動量の総和が最小となるように，各供給点（からの入荷トラック）の入荷口への割当と各需要点（への出荷トラック）の出荷口への割当を決定するクロスドック入出荷口割当問題が発生する．当研究では，この問題に対するタブー探索法を提案し，提案法が従来法よりも精度の高い解を求めることを数値実験で検証する．

Abstract

In a cross-dock, goods are directly transferred by material-handling equipment from incoming trucks at the receiving dock doors to outgoing trucks at the shipping dock doors. In this study, we consider the cross-dock door assignment problem in which the assignment of incoming trucks to receiving dock doors and outgoing trucks to shipping dock doors is determined so as to minimize the total transfer distance of goods inside the cross-dock. We propose a tabu search heuristic for this problem and demonstrate the effectiveness of our heuristic through computational experiments.

キーワード: クロスドッキング，入出荷口割当，タブー探索法，0−1整数計画法
Keywords: *Cross-docking, Dock Door Assignment, Tabu Search, 0-1 Integer Programming*

1. はじめに

サプライチェーンにおける物流高度化のための有効な手段の一つにクロスドッキング（cross-docking）がある．クロスドッキングは複数の供給点からクロスドック（物流センター）へ入荷した商品を在庫として保管することなく直ちに需要点別に仕分けて出荷する仕組みであり，この導入により，輸送コストの削減とリードタイムの短縮を同時に実現することができる[1]．

クロスドックには，通常，複数の入荷口と出荷口が設置される．そのため，クロスドック内における各商品の輸送移動量（輸送量と移動距離の積）は，各供給点からの入荷トラックがどの入荷口へ商品を入庫し，各需要点へ向かう出荷トラックがどの出荷口から商品を出庫するかに依存する．他方，各入出荷口が一定時間以内に処理できる輸送量には上限があるため，商品の入出庫においては，各入出荷口の最大輸送処理量を考慮する必要がある．したがって，クロスドッキングの運用においては，各入出荷口の最大輸送処理量を考慮した上で，クロスドック内の輸送移動量の総和が最小となるように，各供給点（からの入荷トラック）の入荷口への割当と各需要点（への出荷トラック）の出荷口への割当を同時に決定するクロスドック入出荷口割当問題（cross-dock door assignment problem）が発生する．そこで当研究では，この問題のタブー探索法に基づく近似解法を提案する．

クロスドック入出荷口割当問題に関しては，これまでにいくつかの研究がある．例えば，Tsuiら[2]は分枝限定法に基づく，Bozerら[3]は模擬アニーリング法に基づく，Ohら[4]は3段階ヒューリスティック解法と遺伝的アルゴリズムに基づく解法を提案している．しかし，前者の二つの研究で取り扱われている問題は，各供給点（需要点）と各入荷口（出荷口）の１対１の割当を決定する問題であり，各入荷口（出荷口）の最大輸送処理量を考慮してない．また，後者の研究で取り扱われている問題は，需要点の出荷口への割当のみを決定する問題であり，供給点の入荷口への割当を考慮していない．他方，Zhuら[5]，Guignardら[6]，Nassiefら[7]は，当研究と同等なクロスドック入出荷口割当問題を設定し，その最適および近似解法を提案している．Zhuらは，この問題を双

*1 情報通信学研究科情報通信学専攻 修士課程
Graduate School of Information and Telecommunication Engineering, Course of Information and Telecommunication Engineering, Master's Program
*2 情報通信学部経営システム工学科
School of Information and Telecommunication Engineering, Department of Management Systems Engineering
*3 情報通信学部経営システム工学科 教授
School of Information and Telecommunication Engineering, Department of Management Systems Engineering, Professor

線形整数計画問題へ定式化したあと，それを一般化二次3次元割当問題（generalized quadratic 3-dimensional assignment problem）へ変換し，それに対する既存の最適解法を利用した解法を提案している．Guignardらは，この問題に含まれる一般化割当問題の構造を利用した近似解法と，線形制約条件付き非線形整数計画問題に対する凸包緩和を利用した近似解法を提案している．そして，これらの２つの近似解法がZhuらの提案法よりも小規模な問題例を除いて有効であることを数値実験で検証している．Nassiefらは，この問題の0−1整数計画問題への定式化を提示し，その構造を利用したラグランジュ緩和法に基づく近似解法を提案している．そして，提案した近似解法がGuignardらの近似解法よりも有効であると共に，小規模な問題例においては定式化を汎用の数理計画ソルバーで解く方法が，大規模な問題例においては提案した近似解法が有効であることを数値実験で検証している．これに対して当研究では，タブー探索法に基づくこの問題の近似解法を提案する．そして，提案法が従来法の中で最良と考えられるNassiefらの近似解法より精度の高い解を短時間で求めることを数値実験で検証する．

以下では，次の2.でクロスドック入出荷口割当問題を0−1整数計画問題に定式化する．次いで，3.においてタブー探索法に基づく近似解法を提案する．そして4.では，提案法の有効性を数値実験を通して検証する．

2. 0−1整数計画問題への定式化

供給点の集合を $M=\{1,2,\cdots,m\}$，需要点の集合を $N=\{1,2,\cdots,n\}$，入荷口の集合を $S=\{1,2,\cdots,s\}$，出荷口の集合を $T=\{1,2,\cdots,t\}$，輸送の集合を $Q=\{1,2,\cdots,q\}$ とする．また，

w_{ij} ： 供給点 $i\in M$ から需要点 $j\in N$ への輸送量

$f_i\left(=\sum_{j\in N}w_{ij}\right)$ ： 供給点 $i\in M$ からの供給量

$r_j\left(=\sum_{i\in M}w_{ij}\right)$ ： 需要点 $j\in N$ への需要量

d_{kl} ： 入荷口 $k\in S$ から出荷口 $l\in T$ への移動距離

F_k ： 入荷口 $k\in S$ の最大輸送処理量

R_l ： 出荷口 $l\in T$ の最大輸送処理量

w_h ： 輸送 $h\in Q$ の輸送量

$i(h)$ ： 輸送 $h\in Q$ の供給点

$j(h)$ ： 輸送 $h\in Q$ の需要点

とする．ここで，$Q_i^+=\{h\in Q\mid i(h)=i\},i\in M$，$Q_j^-=\{h\in Q\mid i(h)=j\},j\in N$ とおくと，$f_i=\sum_{h\in Q_i^+}w_h,i\in M$，$r_j=\sum_{h\in Q_j^-}w_h,j\in N$ である．このとき，変数 $\boldsymbol{x}=(x_{ik})$，$\boldsymbol{y}=(y_{jl})$，$\boldsymbol{z}=(z_{hkl})$ を

$$x_{ik}=\begin{cases}1, & \text{供給点 } i \text{ を入荷口 } k \text{ に割り当てる場合}\\ 0, & \text{その他}\end{cases},\quad i\in M,k\in S$$

$$y_{jl}=\begin{cases}1, & \text{需要点 } j \text{ を出荷口 } l \text{ に割り当てる場合}\\ 0, & \text{その他}\end{cases},\quad j\in N,l\in T$$

$$z_{hkl}=\begin{cases}1, & \text{輸送 } h \text{ を入荷口 } k \text{ と出荷口 } l \text{ に割り当てる場合}\\ 0, & \text{その他}\end{cases},\quad h\in Q,k\in S,l\in T$$

とすると，前述のクロスドック入出荷口割当問題は，0−1整数計画問題：

$$\text{(P)}\quad \min.\quad \sum_{h\in Q}\sum_{k\in S}\sum_{l\in T}w_hd_{kl}z_{hkl} \tag{1}$$

$$\text{s. t.}\quad \sum_{k\in S}\sum_{l\in T}z_{hkl}=1,\quad h\in Q \tag{2}$$

$$\sum_{l\in T}z_{hkl}=x_{i(h)k},\quad h\in Q,k\in S \tag{3}$$

$$\sum_{k\in S}z_{hkl}=y_{j(h)l},\quad h\in Q,l\in T \tag{4}$$

$$\sum_{i\in M}f_ix_{ik}\le F_k,\quad k\in S \tag{5}$$

$$\sum_{j\in N}r_jy_{jl}\le R_l,\quad l\in T \tag{6}$$

$$x_{ik}\in\{0,1\},\quad i\in M,k\in S \tag{7}$$

$$y_{jl}\in\{0,1\},\quad j\in N,l\in T \tag{8}$$

$$z_{hkl}\in\{0,1\},\quad h\in Q,k\in S,l\in T \tag{9}$$

に定式化される[4]．ここで，式(1)は目的関数で輸送移動量（輸送量×移動距離）の総和を最小化することを表す．そして式(2)は，各輸送をいずれかの入荷口と出荷口に割り当てることを規定する．また式(3)は，輸送 h の供給点 $i(h)$ を入荷口 k に割り当てるならば，輸送 h が入荷口 k といずれかの出荷口に割り当てられることを規定する．同様に式(4)は，輸送 h の需要点 $j(h)$ を出荷口 l に割り当てるならば，輸送 h が出荷口 l といずれかの入荷口に割り当てられることを規定する．さらに，式(5)は各入荷口に割り当てた供給点の総輸送量がその入荷口の最大輸送処理量以下であることを，式(6)は各出荷口に割り当てた需要点の総輸送量がその出荷口の最大輸送処理量以下であることを規定する．

3. タブー探索法の構築

供給点の集合 M を入荷口 S の集合に移す写像 $a:M\to S$ を入荷口割当と呼び，需要点の集合 N を出荷口 T の集合に移す写像 $b:N\to T$ を出荷口割当と呼ぶと，(P)の実行可能解を求めることは，実行可能な入荷口割当 a と出荷口割当 b を定めることに他ならない．そこで以下では，そのような解 (a,b) を求めることを考えるが，解 (a,b) の近傍を $N(a,b)$ と記すと，タブー探索法は，探索した解を一定時間保持するタブーリストリスト Ξ を準備し，$N(a,b)\backslash\Xi$ 内の最良解を探索することを基本操作とする．タブー探索法の手順は，概略，以下の通りである．ただし，次の手順における TL はタブーリストの長さである．

手順　タブー探索法

S0.　$p=0$ とおき，初期解 (a^p,b^p) を生成する．$\Xi=\emptyset$ とおく．

S1.　$N(a,b)\backslash\Xi$ 内の最良解 (a^{p+1},b^{p+1}) を求める．

S2. 終了条件が満たされれば探索を終了し，探索中に得られた最良解（暫定解）を出力する．
S3. $\Xi := \Xi \cup \{(a^p, b^p)\} \setminus \{(a^{p-TL}, b^{p-TL})\}$ とする．
S4. $p := p+1$ として S1 へ．

3.1 近傍構造の設計

当研究では，解 (a,b) の近傍 $N(a,b)$ に次の4つの近傍の和集合を用いるものとし，(P) の制約式(5),(6)を満たさない実行不可能な解 (a,b) も探索空間に含めるものとする．

(a) ある供給点を割り当てた入荷口を別の入荷口にシフトする入荷口シフト−近傍：

$$N^M_{shift}(a,b) = \{(a',b) \mid a'(i_1) \neq a(i_1), a'(i_2) = a(i_2), (\forall i_2 \in M \setminus \{i_1\}), i_1$$

(b) ある需要点を割り当てた出荷口を別の出荷口にシフトする出荷口シフト−近傍：

$$N^N_{shift}(a,b) = \{(a,b') \mid b'(j_1) \neq b(j_1), b'(j_2) = b(j_2), (\forall j_2 \in N \setminus \{j_1\}), j_1$$

(c) ある供給点を割り当てた入荷口と別のある供給点を割り当てた入荷口を交換する入荷口交換−近傍：

$$N^M_{swap}(a,b) = \{(a',b) | a'(i_1) = a(i_2), a'(i_2) = a(i_1), a'(i_3) = a(i_3), (\forall i_3 \in M \setminus \{i_1, i_2\}$$

(d) ある需要点を割り当てた出荷口と別のある需要点を割り当てた出荷口を交換する出荷口交換−近傍：

$$N^N_{swap}(a,b) = \{(a,b') | b'(j_1) = b(j_2), b'(j_2) = b(j_1), b'(j_3) = b(j_3), (\forall j_3 \in N \setminus \{j_1, j_2\}$$

3.2 評価関数の設計

当研究のタブー探索法においては，上述のように実行不可能な解 (a,b) も探索空間に含まれるため，その評価関数は実行不可能な解 (a,b) にペナルティを与えるように設定する．具体的には，解 (a,b) の制約式(5),(6)に対するペナルティをそれぞれ

$$p^A_k(a) = \max\{\sum_{\substack{i \in M \\ a(i)=k}} f_i - F_k, 0\}, \quad k \in S \tag{10}$$

$$p^B_l(b) = \max\{\sum_{\substack{j \in N \\ b(j)=l}} r_j - R_l, 0\}, \quad l \in T \tag{11}$$

とし，ペナルティ重み $\alpha_k, k \in S$ と $\beta_l, l \in T$ を用いて，解 (a,b) の評価関数を

$$c(a,b) = \sum_{i \in M}\sum_{j \in N} w_{ij} d_{a(i),b(j)} + \sum_{k \in S} \alpha_k \cdot p^A_k(a) + \sum_{l \in T} \beta_l \cdot p^B_l(b) \tag{12}$$

と設定する．α_k と β_l には正の初期値を設定し，一定回数 $\zeta(>0)$ の繰り返しごとに，直前の ζ 回の探索中に得られた解の中の最良解 (a',b') を用いて更新する．直前の ζ 回の探索中に 1 度も実行可能解が見つからなかった場合には，ペナルティ重みが小さいと判定し，α_k と β_l をそれぞれ

$$\alpha_k := \alpha_k(1 + \gamma^A p^A_k(a')), \quad k \in S \tag{13}$$

$$\beta_l := \beta_l(1 + \gamma^B p^B_l(b')), \quad l \in T \tag{14}$$

と更新する．γ^A と γ^B の値はそれぞれ，パラメータ $\Delta^+(0 < \Delta^+ < 1)$ を用いて，$0 < \gamma^A p^A_k(a') < \Delta^+, k \in S$ と $0 < \gamma^B p^B_l(b') < \Delta^+, l \in T$ を満たす最大の値に定める．一方，直前の ζ 回の探索中に 1 度でも実行可能解が見つかった場合には，ペナルティ重みが大きい（十分）と判定し，α_k と β_l をそれぞれ，パラメータ $\Delta^-(0 < \Delta^- < 1)$ を用いて，

$$\alpha_k := \alpha_k(1 - \Delta^-), \quad k \in \{k \in S \mid p^A_k(a') = 0\} \tag{15}$$

$$\beta_l := \beta_l(1 - \Delta^-), \quad l \in \{l \in T \mid p^B_l(b') = 0\} \tag{16}$$

と更新する．なお，この α_k と β_l の更新方法は，一般化割当問題（generalized assignment problem）に対する柳浦ら[8]の方法を参考に設計したものである．

3.3 タブーリストの設計

タブーリスト Ξ は解をそのまま保持するのではなく，解の属性（attribute）を保持するのが一般的である．当研究では，上述の 4 つの近傍に対して以下のような解の属性を保持する．

(a) 入荷口シフト−近傍
供給点 i を割り当てた入荷口 $a(i)$ を他の入荷口にシフトした場合には，ペア $(i, a(i))$ をタブーリストに保持する．そして，タブーリストにペア (i,k) が含まれている場合には，供給点 i を割り当てたある入荷口を入荷口 k にシフトすることを禁止する．

(b) 出荷口シフト−近傍
需要点 j を割り当てた出荷口 $b(j)$ を他の出荷口にシフトした場合には，ペア $(j, b(j))$ をタブーリストに保持する．そして，タブーリストにペア (j,l) が含まれている場合には，需要点 j を割り当てたある出荷口を出荷口 l にシフトすることを禁止する．

(c) 入荷口交換−近傍
供給点 i を割り当てた入荷口 $a(i)$ と供給点 i' を割り当てた入荷口 $a(i')$ を交換した場合には，ペア $(i, a(i))$ とペア $(i', a(i'))$ をタブーリストに保持する．そして，タブーリストにペア (i,k) が含まれている場合には，供給点 i を割り当てた入荷口 k とある供給点を割り当てたある入荷口を交換することを禁止する．

(d) 出荷口交換−近傍
需要点 j を割り当てた出荷口 $b(j)$ と需要点 j' を割り当てた出荷口 $b(j')$ を交換した場合には，ペア $(j, b(j))$ とペア $(j', b(j'))$ をタブーリストに保持する．そして，タブーリストにペア (j,l) が含まれている場合には，需要点 j を割り当てた出荷口 l とある需要点を割り当てたある出荷口を交換することを禁止する．

なお，タブーリストの実装には，通常，リストをそのまま用いるのではなく配列を用いる[8]．まず，入荷口割当 a に対して配列 $A^M[i,k](i \in M, k \in S)$ を，出荷

口割当 b に対して配列 $A^N[j,l](j \in N, l \in T)$ を用意し，これらの配列の全ての要素の値を $-\infty$ に初期化する．そして，タブー探索法の反復回数 p において，供給点 i を割り当てた入荷口 k が変更（シフトまたは交換）された場合に $A^M[i,k] := p$ と更新し，需要点 j を割り当てた出荷口 l が変更（シフトまたは交換）された場合に $A^N[j,l] := p$ と更新する．各反復においては，

$$(\text{そのときの反復回数}) - A^M[i,k] \leq TL$$

ならば，供給点 i を割り当てた入荷口を入荷口 k に変更（シフトまたは交換）することを禁止し，

$$(\text{そのときの反復回数}) - A^N[j,l] \leq TL$$

ならば，需要点 j を割り当てた出荷口を出荷口 l に変更（シフトまたは交換）することを禁止する．これにより，タブーリストに解の属性が含まれているかどうかを $O(1)$ で判定できる．

3.4 長期メモリの導入

探索がいくつかの解を経由したあとに元の解に戻る現象を一般にサイクリング（cycling）と呼ぶ．上述のタブーリストの導入により，短期的なサイクリングを防ぐことが可能になるが，ここではさらに長期的なサイクリングを防ぐための長期メモリを導入する[8]．

入荷口割当 a に対して長期メモリ $LTM_a[i,k](i \in M, k \in S)$ を，出荷口割当 b に対して長期メモリ $LTM_b[j,l](j \in N, l \in T)$ を導入し，$LTM_a[i,k]$ には供給点 i を割り当てた入荷口 k を変更（シフトまたは交換）した回数を保持し，$LTM_b[j,l]$ には需要点 j を割り当てた出荷口 l を変更（シフトまたは交換）した回数を保持する．そして，パラメータ $\omega(>0)$ を与え，供給点 i を割り当てた現在の入荷口を入荷口 k に変更した場合には，式(12)の評価関数値 $c(a,b)$ に $\omega \times LTM_a[i,k]$ を加えた値を改めてその評価関数値とし，需要点 j を割り当てた現在の出荷口を出荷口 l に変更した場合には，式(12)の評価関数値 $c(a,b)$ に $\omega \times LTM_b[j,l]$ を加えた値を改めてその評価関数値とする．これにより，頻繁に変更（シフトまたは交換）を繰り返す供給点 i を割り当てた入荷口 k と需要点 j を割り当てた出荷口 l に解の変更をされにくくすることが可能となる．

3.5 暫定解からの再出発

タブー探索法においては，タブーリストを用いて近傍を制限しているため，本来の局所最適解が求まらない場合がある．そこでここでは，式(12)の評価関数 $c(a,b)$ における暫定解が更新されない繰返し数が一定回数 $\eta(>0)$ を超えた場合には，タブーリストを空にした上で，暫定解から探索を再出発するものとする．

3.6 初期解の生成と終了条件設定

タブー探索法の初期解はランダムに生成するものとする．すなわち，各供給点を割り当てる入荷口は $[1,s]$ の一様整数乱数で定め，各需要点を割り当てる出荷口は $[1,t]$ の一様整数乱数で定めるものとする．

また，終了条件としては，式(12)の評価関数 $c(a,b)$ における暫定解が更新されない繰り返し数と，本来の目的関数における実行可能な暫定解が更新されない繰り返し数が一定回数 $\xi(>0)$ を超えた場合に探索を終了するものとする．

4. 数値実験による検証

当研究で提案したタブー探索法に基づく近似解法（以下 TS と呼ぶ）を

- Nassiefら[7]が提案したラグランジュ緩和に基づく近似解法（以下 LR と呼ぶ）
- (P) を数理計画ソルバーで解く方法（以下 MIP と呼ぶ）

と比較するために，数値実験を実施した．実験で使用した問題例はNassiefら[7]の問題例の作成方法を踏襲し，以下の(1)～(3)で作成した．また，実験環境については以下の(4)～(9)の通りである．

(1) 供給点 $i \in M$ から需要点 $j \in N$ への輸送量 w_{ij} は次のように定めた．まず，$M \times N$ の部分集合 E を次の(a)，(b)，(c)で定めた．

(a) 供給点 $i \in M$ ごとの需要点 $j(i) \in N$ をランダムに発生させ，$E = \{(i, j(i)) \in M \times N\}$ とおく．

(b) 需要点 $j \in \{j \in N \mid j \neq j(i)\}$ ごとの供給点 $i(j) \in M$ をランダムに発生させ，$E := E \cup \{(i(j), j) \in M \times N\}$ とする．

(c) 供給点から需要点への輸送の密度を表すパラメータ ρ を与え，$|E| = \lceil m \times n \times \rho \rceil$（$\lceil \cdot \rceil$ は・以上の最小の整数）となるまで次を繰り返す．供給点と需要点の組み $(i,j) \in (M \times N) \setminus E\}$ をランダムに発生させ，$E := E \cup \{(i,j)\}$ とする．

次に，供給点 $i \in M$ から需要点 $j \in N$ への輸送量 w_{ij} を次式で定めた．

$$w_{ij} = \begin{cases} U[10,50], & (i,j) \in E \\ 0, & \text{その他} \end{cases}, \quad i \in M, j \in N \qquad (17)$$

ただし，$U[10,50]$ は $[10,50]$ の一様整数乱数である．

(2) 入荷口 $k \in S$ から出荷口 $l \in T$ への移動距離 d_{kl} は次式で定めた．

$$d_{kl} = 8 + |k - l|, \quad k \in S, l \in T \qquad (18)$$

(3) 入荷口 $k \in S$ の最大輸送処理量 F_k と出荷口 $l \in T$ の最大輸送処理量 R_l は，最大輸送処理量の余裕率を表すパラメータ ϕ を与え，次式で定めた．

$$F_k = F = \left\lceil (1+\phi) \cdot \frac{\sum_{i \in M} \sum_{j \in N} w_{ij}}{s} \right\rceil, \quad k \in S \qquad (19)$$

$$R_l = R = \left\lceil (1+\phi) \cdot \frac{\sum_{i \in M} \sum_{j \in N} w_{ij}}{t} \right\rceil, \quad l \in T \qquad (20)$$

ただし，$\lceil \cdot \rceil$ は・以上の最小の整数である．

(4) 使用計算機は Intel® Core™i7-4770 CPU 3.40GHz の CPU と 8GB のメモリを搭載した PC（OS：

Table 1. Computational Results

m	n	s	t	ρ	ϕ	TS		LR	MIP
						ofv	time	ofv	ofv
25	25	10	10	0.25	0.10	43039	34.962	43178	44611
					0.20	42404	34.580	42480	42983
					0.30	41967	36.715	42005	43485
				0.50	0.10	99314	66.602	99664	103070
					0.20	96851	72.470	96925	99566
					0.30	94860	62.415	94956	98473
				0.75	0.10	155342	71.422	155863	—
					0.20	149509	93.076	149649	154649
					0.30	146741	91.582	146762	152250
50	50	20	20	0.25	0.10	228758	362.445	231110	248528
					0.20	220306	411.626	222979	241517
					0.30	214805	392.508	216868	252384
				0.50	0.10	502812	561.446	518686	—
					0.20	475218	684.317	478415	524106
					0.30	461565	470.647	463102	507034
				0.75	0.10	—	314.068	—	—
					0.20	728374	890.255	735610	784711
					0.30	718383	554.134	718879	756802
75	75	30	30	0.25	0.10	625917	1192.080	636430	768936
					0.20	599174	1354.652	607081	712473
					0.30	578461	1339.410	585873	684962
				0.50	0.10	—	1159.159	—	—
					0.20	1249310	2450.085	1263639	1405464
					0.30	1208002	1879.300	1215980	1336229
				0.75	0.10	—	1062.741	—	—
					0.20	1949901	3266.429	1967816	2116496
					0.30	1922140	2334.181	1922175	2044066

Windows 7 professional 64bit SP1）で，使用言語はC#（処理系：Microsoft Visual Studio 2015 Pro）である．

(5) LRの終了条件は計算時間3600秒とした．

(6) MIPに使用した数理計画ソルバーはGurobi Optimizer Version 6.5[9]である．

(7) MIPの計算は計算時間3600秒で打ち切りとした．

(8) TSのパラメータは，予備実験の結果より，以下のように定めた．

- タブーリストの長さTLは$TL = 15$に設定した．
- 評価基準におけるペナルティ重み$\alpha_k, k \in S$と$\beta_l, l \in T$の初期値は1.0とし，その調整パラメータζ，Δ^+，Δ^-はそれぞれ$\zeta = 1.0$，$\Delta^+ = 0.1$，$\Delta^- = 0.1$に設定した．なお，ペナルティ重み$\alpha_k, k \in S$と$\beta_l, l \in T$が0.001より小さい値に更新された場合には0.001とし，1000より大きい値に更新された場合には1000とした．
- 長期メモリの重み付けパラメータωは$\omega = 2$に設定した．
- 暫定解の再出発パラメータηはη = 2000に設定した．
- 終了条件の停止パラメータξはξ = 10000に設定した．

また，TSは以上のパラメータ設定の下で（初期解を変えて）30回実行し，その中の最良値をTSで求まった実行可能解の目的関数値とした．

以上の環境の下で実施した数値実験の結果をTable 1に示す．ここで，Table 1のofvは求まった実行可能解の目的関数値であり，timeはTSを30回実行した計算時間の合計で単位は秒である．ただし，計算終了までに実行可能解が求まらなかった場合には，ofvに「－」を記した．また，MIPの結果はいずれも計算時間3600秒以内に最適解が求まらず打ち切りとなった場合である．

この数値実験の結果は，TSの方がLRとMIPよりも，より少ない計算時間でより良い解が求まることを強く示唆している．また，m，n，s，tの問題規模が大きくなるほど（当然のことながら）TSの計算時間が増大し，問題規模が同じならば，パラメータρが大き

くなるほど（実行可能解が求まった問題例において）TSの計算時間が増大する傾向にあると共に，パラメータ ϕ を 0.1 または 0.3 に設定した場合よりも 0.2 に設定した場合の方がTSの計算時間が増大する傾向にあることを示している．

5. おわりに

当研究では，クロスドック入出荷口割当問題を取り上げ，タブー探索法に基づくこの問題の近似解法を提案した．そして，提案した解法の有効性を検証するための数値実験を実施した．その結果，提案法が従来研究の中で最良と考えられる Nassief ら[7]の近似解法よりも精度の高い解を短時間で求めることを検証した．

しかしながら，提案法は問題規模が大きくなるにつれて所要計算時間が大幅に増加することから，その実用性を高めるためには，所要計算時間の更なる短縮が必要であると思われる．それゆえ今後，より効果的な近傍構造を設計すると共に，より効率的な近傍探索方法を開発し，提案法の一層の効率化を図る必要がある．

参考文献

1) D. Simchi-Levi, P. Kaminsky and E. Simchi-Levi, Designing and Managing the Supply Chain, McGraw-Hill, 2000

2) L. Y. Tsui and C. -H. Chang, "An Optimal Solution to a Dock Door Assignment Problem", *Computers and Industrial Engineering*, Vol. 23, Issues 1-4, pp.283-286, 1992

3) Y. A. Bozer and H. J. Carlo, "Optimizing Inbound and Outbound Door Assignments in Less-than-truckload Crossdocks", *IIE Transactions*, Vol. 40, Issue 11, pp. 1007-1018, 2008

4) Y. Oh, H. Hwang, C. N. Cha and S. Lee, "A Dock-door Assignment Problem for the Korean Mail Distribution Center", *Computers & Industrial Engineering*, Vol. 51, Issue 2, pp. 288-296, 2006

5) Y. -R. Zhu, P. M. Hahm, M. Guignard, "New Approach for the Cross-dock Door Assignment Problem", In Anais do XLI Simposio Brasileiro de Pesquisa Operacional, Porto Seguro, Bahia, Brazil, 2009

6) M. Guignard, P. M. Hahn, A. Alves Pessoa and D. Cardoso da Silva, "Algorithms for the Crossdock Door Assignment Problem", In *Proceedings of the Fourth International Workshop on Model-based Mataheuristics*, Brazil, 2012

7) W. Nassief, I. Contreras and R. Asad, "A Mixed-integer Programming Formulation and Lagrangean Relaxation for Cross-dock Door Assignment Problem", *International Journal of Production Research,* Vol.54, No.2, pp.494-508, 2016

8) 柳浦睦憲,茨木俊秀「組合せ最適化-メタ戦略を中心として-」朝倉書店, 2001

9) Gurobi Optimization, Inc. Gurobi Optimizer Reference Manual Version 6.5, 2016