経営科学(日本オペレーションズ・リサーチ学会邦文機関誌) 第19巻 増補 (1975年12月)

コンテナ荷役機械のスケジューリング†

越智利夫*
三根久**

1. はじめに

コンテナ専用列車とストレイジ(コンテナの1時保管場所)の間の荷役を行なう荷役機械のスケジューリングの問題を考える．いま，図1のように荷役機械線路にそった1次元の座標系をとると列車，ストレイジ上のコンテナの位置は完全に指定できる．ここで列車上にはその駅でおろすコンテナがあり，積みこむことのできる空き場所がある．また，ストレイジ側にも列車につみこむべきコンテナと列車からおろすことのできる空き場所がある．この問題を複雑にする条件の1つは，列車からコンテナを降ろしたあとの空き場所はストレイジからの積みこみに利用できるし，ストレイジのコンテナを列車に積みこんだあとの空き場所は，列車からの降ろしに利用できることにある．

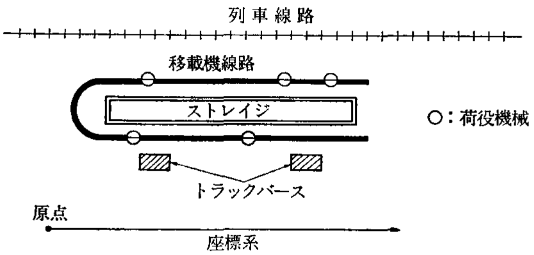


図1 荷役システムの例

荷役機械は図1に示すように，列車線路に並行に設置されたレール上に1台または複数台設置されており，スケジューリングの目的は列車停止時間内に効率よく荷役を行なうことであるが，この問題は時間的要素と空間的要素を含むため，一度に総合的問題として解くことは困難である．そこで次のステップを取ることを提案する．

ステップ1：列車のコンテナをどこに降ろし，ストレイジのコンテナをどこに積みこむかをコンテナの移動距離の総和を最小にする条件のもとで決定する．

ステップ2：ステップ1で定まったコンテナの移動を効率よく行なうための荷役機械のスケジュールを定める．

ここでステップ1は，コンテナ移動距離の総和を最小にすることを目的としてコンテナ移動場所を定める一種の割り当て問題である．

本論文では，ステップ1を，(i)整数形線形計画問題として定式化し，暫定解を求め，(ii)その暫

---

† 1975年3月6日受理.
* 日立製作所日立研究所.
** 京都大学工学部.

定解に矛盾が含まれている場合は，その矛盾を順次解消していく方法を提案する．ここで矛盾とはすでにコンテナの置いてある場所にコンテナを運ぶことであって，本論文で提案する矛盾解消法は，実用的な近似解を与えるものである．この方式の詳細を第**2**章で論ずる．

なお，この問題は，割り当て問題あるいは巡回セールスマン問題 [1] に変形できるもので Eastman [2], Bellmore [3] らによる解法を適用すれば理論的には正確な解が得られるが，現実には問題のもつ特殊性から計算に非常に多くの時間を要する欠点がある．本論文で実用的解法を提案したのはこのためである．なお，正確な解を得る方法，およびその際の問題点の概要を第**2・4**節で論ずる．

ステップ2は，ステップ1で定まったコンテナの移動を効率よく行なうための荷役機械のスケジュールを定めるもので，分枝限定法（以下 B-B 法と略称する）を適用することにより解を得ることができる．この問題を第**3**章で論ずる．

なお，本論文では荷役機械のスケジューリングをステップ1とステップ2に分けて解くことを提案したが，荷役機械が1台の場合には，その移動距離を最小にすることを目的として，コンテナの移動場所の決定と荷役機械の最適スケジュールを同時に求める方法がある．それは，コンテナの移動の可能性をゲームの木の形に展開してその木を最終の枝まで探索するものであるが，計算時間を非常に多く要して実用的でない．しかし，この場合ゲームの木の枝の探索を1段あるいは2段程度で中止することは直観的な方法としてすでに考えられているものと一致する [4]．本論文ではゲームの木を p 段探索することを p 手読み法と呼び，p 手読み法を動的計画法を使って定式化するとともに，1手読み法および2手読み法と本論文で提案したステップ1とステップ2に分ける方法との比較を行なう．この問題を第4章で論ずる．

2. コンテナ移動場所の決定

2・1 暫 定 解

列車から降ろすコンテナの数を n_1 個，列車の空き場所の個数を n_2 個，ストレイジから列車に積み込むべきコンテナの数を m_1 個，ストレイジの空き場所の数を m_2 個とし，

$$n_3 = n_1 + n_2$$

$$m_3 = m_1 + m_2$$

とする．したがって空き場所の関係から次の条件を得る．

条件 1. この問題が解をもつのは

$$n_3 \geq m_1$$

$$m_3 \geq n_1$$

$$n_3 + m_3 \geq n_1 + m_1 + 1$$

のときである．

以下，この条件は満たされているとする．

荷役機械線路にそって図1のような1次元座標系をとり，降ろすコンテナ位置を $\alpha_1, \cdots\cdots, \alpha_{n_1}$,

列車上の空き場所の位置を$\alpha_{n_1+1},\cdots\cdots,\alpha_{n_3}$，積み込むコンテナ位置を$\beta_1\sim\beta_{m_1}$，ストレイジの空き場所の位置を$\beta_{m_1+1},\cdots\cdots,\beta_{m_3}$とする.

コンテナの移動においては移動距離の総和を最小にすることを目的とすると，次のような方法を考えることができる.

まず，x_{ij}を列車のα_iの位置のものをストレイジのβ_jの位置に運ぶとき1，運ばないとき0をとる決定変数とし，y_{ij}をストレイジβ_iのものを列車α_jの位置に運ぶとき1，運ばないとき0をとる決定変数とする.

また，α_iとβ_jの間の距離を$\rho_{\alpha_i\beta_j}$で表わす．このとき次のことがいえる.

荷物がすでに存在しているところに運ぶ矛盾を許すと，列車側，ストレイジ側，それぞれ独立に計画を立てることができる．それは次のように表現できる.

定理1. 列車のコンテナの移動は，

$$\sum_{j=1}^{m_3} x_{ij}=1 \qquad (i=1\sim n_1)$$

$$\sum_{i=1}^{n_1} x_{ij}\leq 1 \qquad (j=1\sim m_3)$$

$$0\leq x_{ij}\leq 1,\ x_{ij}:\text{整数}$$

の条件のもとで，

$$\sum_{i=1}^{n_1}\sum_{j=1}^{m_3}\rho_{\alpha_i\beta_j}x_{ij}\longrightarrow \text{最小}$$

を求めることであり，ストレイジ上のコンテナの移動は，

$$\sum_{j=1}^{n_3} y_{ij}=1 \qquad (i=1\sim m_1)$$

$$\sum_{i=1}^{m_1} y_{ij}\leq 1 \qquad (j=1\sim n_3)$$

$$0\leq y_{ij}\leq 1,\ y_{ij}:\text{整数}$$

の条件のもとで，

$$\sum_{i=1}^{m_1}\sum_{j=1}^{n_3}\rho_{\beta_i\alpha_j}y_{ij}\longrightarrow \text{最小}$$

を求めることである.

これらの問題は整数形線形計画法，あるいは割り当て問題に変形してHichcockのアルゴリズム[3]で解くことができる．以上で列車から降ろすコンテナとストレイジから積み込むコンテナの仮の移動場所が定まったわけであるが，この結果はコンテナのはいっているところにコンテナを運ぶという矛盾を含んでいることがある．そのためこの解を暫定解と呼ぶことにする．暫定解が矛盾を含んでいなければそれが正解であるが，矛盾を含むときは以下のように解消をする.

2・2 暫定解の改良

暫定解の矛盾を以下のステップで解消する．なお，xの位置のものをyの位置に運ぶことを

$x \to y$ と書く．また，$x \to y$ かつ $y \to x$ を $x \leftrightarrow y$ と書くことにする．

定理 2.

2つの移動 $\alpha_{i_1} \to \beta_{j_1}$，$\alpha_{i_2} \to \beta_{j_2}$ が $\alpha_{i_1} < \alpha_{i_2}$，$\beta_{j_2} < \beta_{j_1}$ のとき交叉をしているという．この場合はこの計画を $\alpha_{i_1} \to \beta_{j_2}$，$\alpha_{i_2} \to \beta_{j_1}$ と変更すると交叉の除去ができる．そのさい，$\rho_{\alpha_{i_1}}{}^{\beta_{j1}} + \rho_{\alpha_{i_2}}{}^{\beta_{j2}} \geq \rho_{\alpha_{i_1}}{}^{\beta_{j2}} + \rho_{\alpha_{i_2}}{}^{\beta_{j1}}$ が成立する．この関係は，$\alpha_{i_1} > \alpha_{i_2}$，$\beta_{j_2} > \beta_{j_1}$ のときも同様である．また β と α を逆にしても成立する．

証明.　$\alpha_{i_1} < \alpha_{i_2}$，$\beta_{j_2} < \beta_{j_1}$ の場合について示す．他も同様である．このとき，

(1)　$\rho_{\alpha_{i_1}}{}^{\beta_{j1}} + \rho_{\alpha_{i_2}}{}^{\beta_{j2}} = \rho_{\alpha_{i_1}}{}^{\beta_{j2}} + \rho_{\alpha_{i_2}}{}^{\beta_{j1}}$

　　if $\beta_{j_1} \leq \alpha_{i_1}$ または $\beta_{j_2} \geq \alpha_{i_2}$ のとき

(2)　$\rho_{\alpha_{i_1}}{}^{\beta_{j1}} + \rho_{\alpha_{i_2}}{}^{\beta_{j2}} > \rho_{\alpha_{i_1}}{}^{\beta_{j2}} + \rho_{\alpha_{i_2}}{}^{\beta_{j1}}$

　　その他のとき

が成立する．

系 2・1　1つの移動が複数の移動と交叉するとき，距離を増大せず交叉の解除ができる．

暫定解の中には

(1)　$\alpha_{i_1} \longleftrightarrow \beta_{j_1}$

(2)　$\alpha_{i_1} \longrightarrow \beta_{j_1}$，$\beta_{j_1} \longrightarrow \alpha_{j_2}$，$\alpha_{j_2} \longrightarrow \beta_{j_3}$，……，$\beta_{j_k} \longrightarrow \alpha_{i_1}$

のように，すでにコンテナのあるところに運ぶという矛盾した組み合わせのあることがある．

定義 1.　暫定解に現われた局所的に矛盾している移動の組み合わせを矛盾作業と呼ぶ．とくに(1)のタイプの矛盾作業を単純な矛盾作業と呼ぶ．

定義 2.　(1)，(2)のタイプの関係をもつ，α 点，β 点を矛盾作業点と呼ぶ．矛盾作業点以外の点を利用可能点と呼ぶ．

この定義によると利用可能点は，空き場所（運び込むことのできる所），および矛盾作業に含まれない移動で荷物を運び込んでいる場所を指す．

定理 2，系 2・1 より，(2)のタイプの矛盾作業はループを作っているが，交叉を除去することにより，距離を増大させないで単純な矛盾作業に分解できる．

定義 3.　単純な矛盾作業が，その間の利用可能な点や，矛盾作業に含まれない移動をもたず連続しているとき，これをまとめて（単純な）矛盾作業集合と呼ぶ．

図2では，3つの単純な矛盾作業集合 S_1, S_2, S_3 ができたことになる．

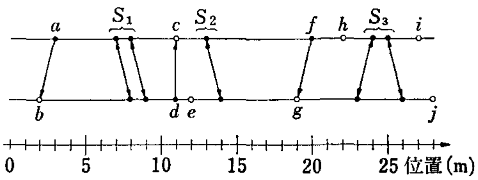


図2　矛盾作業の例

定理 3.　単純な矛盾作業 $\alpha_{i_1} \longleftrightarrow \beta_{j_1}$ に対し，α_{i_1} と β_{j_1} ではさまれる区間を矛盾作業の範囲と呼ぶ．矛盾作業の範囲に矛盾作業に含まれない移動の終点のあるときは距離を増大させず，矛盾作業を解消できる．

問題を単純化するために次のような条件をつける．

条件 2.

(1)　単純な矛盾作業集合の左側（あるいは右側）の列車側およびストレイジ側に利用可能点が1個以上あっても，この矛盾作業集合の解消は左右それぞれの側について列車側の矛盾作業集合に最も近い点かストレイジ側の矛盾作業集合に最も近い点を使用し，しかも，左右それぞれの側において使用される利用可能点の数はたかだか1個とする．したがって，図2のように矛盾作業集合 S_1 の左右の列車側，ストレイジ側にそれぞれ a, b, c, d の利用可能点のあるとき同時に使用される可能性のあるのは (a, c)，(a, d)，(b, c)，(b, d) の4ケースであるとする．

(2)　単純な矛盾作業集合 $\alpha_{i_1} \longleftrightarrow \beta_{j_1}, \cdots\cdots, \alpha_{i_n} \longleftrightarrow \beta_{j_n}$ を点 $\alpha_{i_{n+1}}(\alpha_{i_1} < \cdots\cdots < \alpha_{i_n} < \alpha_{i_{n+1}}$，$\beta_{j_1} < \cdots\cdots < \beta_{j_n})$ だけを使って解消する方法は $\alpha_{i_1} \longrightarrow \beta_{j_1}$，$\beta_{j_1} \longrightarrow \alpha_{i_2}, \cdots\cdots, \alpha_{i_n} \longrightarrow \beta_{j_n}$，$\beta_{j_n} \longrightarrow \alpha_{i_{n+1}}$ とする．その他の点 $\beta_{j_{n+1}}(\beta_{j_n} < \beta_{j_{n+1}})$，$\alpha_{i_0}(\alpha_{i_0} < \alpha_{i_1})$，$\beta_{j_0}(\beta_{j_0} < \beta_{j_1})$ に関しても同様とする．

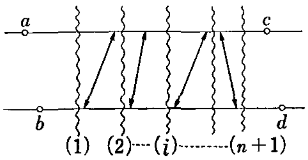


図3　矛盾作業集合の分割

いま，矛盾作業集合の境界を図3で示すように(1)～$(n+1)$ とし，これを (a, c) で解消することを考える．このとき(1)での分割は c で矛盾作業集合を解消するものであるし，(i) での解消は左側の $i-1$ 個の単純な矛盾作業を a で右の $n-i+1$ 個の単純な矛盾作業を c で解消することになる．(i) の場所によっては a 単独や c 単独での解消を含む．

(a, c) を使い (i) で分割したときの距離の増分を $\delta_i(a, c)$ で表わす．また，単純な矛盾作業に左から番号をつけ，列車側の荷物の位置座標 α_i，ストレイジ側の荷物の位置座標 β_i，$(i=1 \sim n)$ とし，a, b, c, d の位置座標をおのおの α_0，β_0，α_{n+1}，β_{n+1}（n は単純な矛盾作業の個数）とすると次のことが容易にいえる．

定理 4.　$\delta_i(a, c)$ は次のように表現できる．

$$\delta_i(a, c) = \sum_{k=1}^{n} |\beta_k - \alpha_{k+\sigma_i(k)}| - \sum_{k=1}^{n} |\beta_k - \alpha_k|$$

ここで，

$$\sigma_i(k) = \begin{cases} +1 & k \geq i \\ -1 & k < i \end{cases}$$

$\delta_i(a, b)$，$\delta_i(b, c)$，$\delta_i(b, d)$ も同様に表現できる．

この矛盾作業集合を (a, c) を使って解消するときの方法は，

$$\delta(a, c) = \min \delta_i(a, c)$$

により得られる．他も同様である．

定義 4.　隣り合った2つの矛盾作業集合間の列車側（あるいはストレイジ側）に空き場所が1つあるいはそれ以下のとき，この2つの矛盾作業に競合関係にあるという．

ある矛盾作業集合が左右どちらの矛盾作業集合とも競合関係にないとき，その矛盾作業集合は単独で解消できる．解消に使用する対は，次の関係，

$$\min\{\delta(a, c),\ \delta(a, d),\ \delta(b, c),\ \delta(b, d)\}$$

を満たすものである.

矛盾作業集合の両端の利用可能点に空き場所でないもの（荷物を運びこんでいる点）があり，その点を矛盾作業の解消に使用するときは，この点の代替行き先を考えねばならない．代替行き先は，利用可能点の左側に求めるものと右側に求めるものがある.

どの方向にも競合関係をもたない矛盾作業集合は独自に解消できるので，残った矛盾作業集合はどこかに競合関係をもつ．この残った矛盾作業集合を左から$S_1, \cdots\cdots, S_h$とする．まず，S_jに注目する．S_jの両端に利用可能点がa, b, c, dの4つあるとする．このとき，S_j解消のための可能性は(a, c)，(a, d)，(b, c)，(b, d)の組み合わせの中にある.

以下，(a, c)を例にとり，S_jの解消のときの損害を示すが(a, d)，(b, c)，(b, d)に対しても同様である.

なお，損害の計算に使う記号は次のようにする．まず矛盾作業集合S_jに関しては**定理4**で使った記号を使う．また，aより左にあるa側の空き場所の座標をα_{-1}（なければ$\alpha_{-1}=-\infty$），aが空き場所でないときaへの荷物はbより来ているとする．すなわち，$\beta_0 \longrightarrow \alpha_0$とする.

同様に，cが空き場所でないときは荷物はdから来ているとする．すなわち，$\beta_{n+1} \longrightarrow \alpha_{n+1}$.
また，c側でS_{j+1}までに空き場所があればこの点をα_{n+2}とする．S_{j+1}までに空き場所のないとき，S_{j+1}のc側のいちばん左の点をα_{n+2}とする．また，矛盾作業集合S_jを(a, c)を使って解消するときの損害を$P_j(a, c, U)$と書く．ここでUはcを使うか否かに関するもので，

$U=\underline{b}$……cを使わない.

$U=*$……cを使う.

$U=L$……cは荷物のある点であるが，c点は矛盾作業集合S_jの解消に使い，cの代替点をcの左に求める.

表1　損害の計算

c	U	a	$P_j(a, c, U)$
c–1	$\underline{b}$	a–1	d_{n+1}
		a–2	$\min(d_{n+1}+A,\ d_{n+1}+B_{n+1})$
	$*$	a–1	$\min\limits_{j=1\sim n} d_j$
		a–2	$\min[\min\limits_{i=2\sim n} d_i+A,\ \min\limits_{i=2\sim n}(d_i+B_i),\ d_1]$
c–2	$\underline{b}$	a–1	d_{n+1}
		a–2	$\min(d_{n+1}+A,\ d_{n+1}+B_{n+1})$
	L	a–1	$\min\limits_{i=1\sim n}(d_i+C_i)$
		a–2	$\min[\min\limits_{i=2\sim n}(d_i+C_i)+A,\ \min\limits_{i=2\sim n}(d_i+B_i+C_i),\ d_1+c_1]$
	R	a–1	$\min\limits_{i=1\sim n} d_i+D$
		a–2	$\min[\min\limits_{i=2\sim n} d_i+D+A,\ \min\limits_{i=2\sim n}(d_i+B_i)+D,\ d_1+D]$

$U=R$……cは荷物のある点であるが，c点は矛盾作業集合S_jの解消に使い，cの代替点をcの右に求める．

とする．次にaとcの状態に関して，

$(a\text{-}1)$……aは空き場所である．

$(c\text{-}1)$……cは空き場所である．

$(a\text{-}2)$……aは空き場所でない．

$(c\text{-}2)$……cは空き場所でない．

とすると各損害は表1のようになる．なお表1では，

$$d_i=\delta_i(a,c)$$

と略記するとともに次の記号を使用した．

$$A=|\alpha_{-1}-\beta_0|-|\alpha_0-\beta_0|$$

$$B_i=|\alpha_{i-1}-\beta_0|-|\alpha_0-\beta_0|$$

$$C_i=|\beta_{n+1}-\alpha_i|-|\alpha_{n+1}-\beta_{n+1}|$$

$$D=|\alpha_{n+2}-\beta_{n+1}|-|\alpha_{n+1}-\beta_{n+1}|$$

さて，矛盾作業集合S_1,……,S_hの解消は次のようにツリー展開を行なう．まず，S_1に対し，a,b,c,dを選び$P_1(x,y,U)$を計算する．ここで(x,y)は(a,c)，(a,d)，(b,c)，(b,d)でUはyが空き場所のとき$\underline{b}$か＊をとり，yが代替を要する点のとき$\underline{b}$かLかRをとる．この組み合わせをツリーの第1段に並べ，下限値として$P_1(x,y,U)$の値を与える．

次に，このようにしてできた下限値の中の最小の下限値をもつものを見いだし，その先の展開を行なう．そのさい，次の点に注意する．ここでは，S_{j+1}の展開を行なう場合につき述べる．

S_jの記号は前項の記号を使い，S_{j+1}のa,b,c,dをa',b',c',d'とする．S_jのcとS_{j+1}のa'の関係について述べるがdとb'の関係も同様である．なお，$P_j(x,c,U)$のUを単にUと書く．

(1) Uが$\underline{b}$のときは(a',c')，(a',d')の計算は通常のとおりでよい．

(2) Uが＊かLのとき，a'とcが一致するときは(a',c')，(a',d')のケースはなくなり(b',c')，(b',d')のみでよい．a'とc'が一致しないときは通常の計算を行なう．

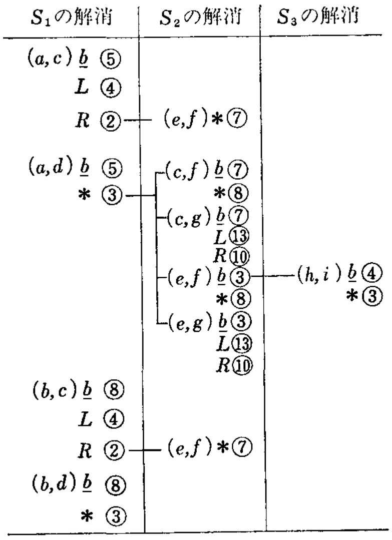
(3) UがRのとき，α_{n+2}がa'より左のときは通常の組み合わせで計算するが，α_{n+2}がa'点かその右のときは(a',c')，(a',d')はなくなる．

(4) UがRでα_{n+2}がS_{j+1}内のとき(b',c')のみとなる．S_jに対する$P_j(a,c,R)$は暫定計算であるのでこの場合(b',c')のときの損害計算で補正する．それには，$\delta_i(b',c')$の代わりに，

$$\delta_i^*(b',c')=\delta_i(b',c')+|\alpha_{n+2}-\alpha_i|$$

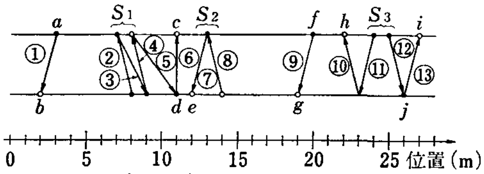
を使う．なお，この場合，c'を使わない，すなわち$(a',c',\underline{b})$のケースはない．

以上の配慮をしながら各ステップで最小の値をもつ枝の先を展開し，展開した枝の下限値には，親枝の下限値と展開した損害$P_j(x,y,U)$の和を与える．その他，B-B法のアルゴリズムに従う．P_jは正の値であり，下限値は単調増加するので，第h段まで進んだ枝で，その枝の下限値がどの枝の下限値より大きくないものがあればそれが1つの解を与える．



○内の数字は損失値

図4 ツリー展開



○内の数字は仕事番号を表わす

図5 矛盾作業の解消結果

2・3 数 値 計 算

図4は図2の簡単な例題に対するツリー展開を示したものであり，図5はそれによる矛盾作業の解消結果を示したものである．なお，以上の操作で得られた解は，ただ単にある位置のコンテナを他の位置に運ぶというだけでなく，部分的に処理順序の制限が加えられる．図5ではそれらは $6 \prec 5 \prec 4 \prec 3 \prec 2$，$7 \prec 8$，$10 \prec 11$，$13 \prec 12$ となっている．ここで，記号 $A \prec B$ は，B は A のあとで行なうことを示す．

2・4 コンテナ移動場所の最適解

以上，本論文ではコンテナの移動場所を定めるための近似解法を提案したが，もともとこの問題の正解は次のようにして求めることができる．それは Eastman [2] が割り当て問題に対し B-B 法を適用した手法を次のように応用すればよい．まず，最初の整数形線形計画問題を解き，矛盾がなければ正解となるところは同じである．矛盾 $\alpha_{i_1} \longleftrightarrow \beta_{j_1}$，$\alpha_{i_2} \longleftrightarrow \beta_{j_2}$ があるとしこのときコンテナ全移動距離を η_1 とする．この計画で第1の $\alpha_{i_1} \longleftrightarrow \beta_{j_1}$ の矛盾をなくするには $\alpha_{i_1} \longrightarrow \beta_{j_1}$ をなくするか $\alpha_{j_1} \longleftarrow \beta_{j_1}$ をなくするしかないのでB-B法を使い，最初の出発点を①点，その損害を η_1 とし①点より $\alpha_{i_1} \longleftarrow \beta_{j_1}$ を含まない道順と $\alpha_{i_1} \longrightarrow \beta_{j_1}$ を含まない道順の2つの分枝を作る．$\alpha_{i_1} \longleftarrow \beta_{j_1}$ を含まない分枝を②とし $\alpha_{i_1} \longrightarrow \beta_{j_1}$ を含まない分枝を③とする．そして分枝②に対しては変数 y_{j_1, i_1} を $y_{j_1, i_1}=0$ として整数形線形計画問題を解き，そのときのコンテナ移動距離を η_2 としてこれを分枝②の損害とする．また，分枝③に対しても同様に $x_{i_1, j_1}=0$ として整数形線形計画問題を解きそのときのコンテナ移動距離を η_3 として，これを分枝③の損害とする．明らかに $\eta_2 \geq \eta_1$，$\eta_3 \geq \eta_1$ である．

以下，分枝②，③の損害の小さいほうを取り，たとえばそれが分枝③だとすると分枝③での解に矛盾があるか否かを調べ，矛盾がなければ正解となる．あれば矛盾の1つをとりその原因とな

っているコンテナの動きそれぞれに対し分枝を作り，上記②，③を計算したのと同じステップをくり返す．そして各分枝を行った小枝の先の損害の値の中で，損害が最も小さくかつ矛盾のないものがみつかればそれが正解である．

この方法は正解の求まる長所はあるが分枝の数が多くなり，各分枝点で整数形線形計画問題を解くために非常に多くの計算時間を要することになる．分枝の数が多くなる理由は次による．たとえば，矛盾が $\alpha_{i_1}\longleftrightarrow\beta_{j_1}$，$\alpha_{i_2}\longleftrightarrow\beta_{j_2}$，$\alpha_{i_3}\longleftrightarrow\beta_{j_3}$ でありこれ以外の利用可能点はかなりはなれたところにしかないとする．このとき $\alpha_{i_1}\longrightarrow\beta_{j_1}$ を除いた解はたとえば $\alpha_{i_1}\longrightarrow\beta_{j_3}$，$\beta_{j_3}\longrightarrow\alpha_{i_3}$，$\alpha_{i_3}\longrightarrow\beta_{j_2}$，$\beta_{j_2}\longrightarrow\alpha_{i_2}$，$\alpha_{i_2}\longrightarrow\beta_{j_1}$，$\beta_{j_1}\longrightarrow\alpha_{i_1}$ のような矛盾を作る．この点で分枝が6個できる．かりにここで $\beta_{j_1}\longrightarrow\alpha_{i_1}$ を除いて，すなわち前回と合わせて $\alpha_{i_1}\longrightarrow\beta_{j_1}$，$\beta_{j_1}\longrightarrow\alpha_{i_1}$ を除いたとしても，たとえば $\alpha_{i_1}\longleftrightarrow\beta_{j_2}$，$\alpha_{i_2}\longleftrightarrow\beta_{j_1}$，$\alpha_{i_3}\longleftrightarrow\beta_{j_3}$ などの矛盾ができ，分枝が多くなり毎回整数形線形計画問題を解くのがたいへんとなる．したがってこの解法は厳密ではあるが，場合によっては多くの計算時間を要し実用的でないことがある．

Eastman のこのような欠点を除く1つの考えは Bellmore [3] により提案されている．ここでは整数形線形計画問題から得られた1つの矛盾作業集合をとりだしこれを除去するときに，この矛盾作業からかならず脱出することができるような分枝の方法をとっている．しかし，この方法は費用行列が非対称で，個々の要素間に相関のないときは有効であるが，本問題のように1部対称性をもち，しかも要素間に三角不等式が成りたつ場合には Eastman と同じような問題がやはり残り分枝が多くなると予想される．

3. 荷役機械の動き

前述の方法で定まったコンテナの移動を荷役機械で行なうことを仕事と呼ぶ．この仕事の数を n 個とし，第 i 番目の仕事 g_i を次の形で表わす．

$$g_i=(x_i, y_i) \qquad (i=1\sim n)$$

これは位置 x_i のコンテナを位置 y_i に移すことを意味する．また，g_i の集合を G で表わす．G の各作業内には，処理順序の指定されたものがある．これらの順序は前後関係がこの規則を満たせばよく，それらの間に他の仕事があってもよい点に特徴をもつ．

(i) 荷役機械が1台の場合

集合 G に含まれる n 個の仕事 $g_1\sim g_n$ の処理に要する時間 T は，

$$T=\sum_i(g_i \text{ の処理に要する時間})+\sum(\text{仕事と仕事の間の移動のための時間})$$

と表わすことができる．ここで，第1項はスケジュールのいかんにかかわらず一定であるから，第2項，すなわち，1つの仕事が終わって次の仕事のために移動する時間の和を最小にするのが最適スケジュールとなる．そこで仕事 i の次に仕事 j をするための移動の時間，すなわち，むだな動きのための時間 $w_{ij}(i, j=1\sim n)$ を，

$$w_{ij}=\begin{cases}\text{距離 } |x_j-y_i| \text{ を移動するに要する時間}\cdots\cdots i\neq j \text{ のとき}\\ \infty\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots i=j \text{ のとき}\end{cases}$$

と定める．また仮空の $n+1$ 番目の仕事 g_{n+1} を作り，これを他の仕事との距離も 0 となるように，

$$w_{i,n+1} = w_{n+1,i} = 0 \qquad (i = 1 \sim n)$$

$$w_{n+1,n+1} = \infty$$

と定め，$(w_{ij})(i,j=1\sim n+1)$ の作る損失行列を W とする．この行列を距離行列とする巡回セールスマン問題を考えて，これに Little [5] による B-B 法のアルゴリズムを適用する．ただし各分枝点では，これまでに採用された要素で作る順序が順序指定に違反すれば下限に ∞ を与える．このようにして得られた解は g_{n+1} を含んだループを作るので，このループから g_{n+1} の要素をはずせば行列 W に対する最適な動きが得られることになる．

(ii) 荷役機械が複数台のとき

荷役機械を h 台まで利用できるとする．まず，各仕事 $g_i=(x_i, y_i)$ に，

$$E_i = \{\text{点 } x_i, y_i \text{ を両端とする区間の実数の集合}\}$$

を対応させ，仕事 g_i と g_j の共通部分を

$$E_i \cap E_j$$

で定める．

$$E_i \cap E_j = \phi$$

とは，2つの仕事が離れていることを示す．

この定義で集合 G は，

$$G = \bigcup_{i=1}^{m} G_i$$

となるよう分離できる．ここで任意の $i, j(i \neq j)$ に対して G_i, G_j は，

① $G_i \cap G_j = \phi$

② $a, b, (a<b)$ が存在して $g_k \in G_i$ を満たすすべての k に対して，

$$\cup E_k \subset [a, b]$$

かつ $g_l \in G_j$ を満たすすべての l に対して，

$$\cup E_l \cap [a, b] = \phi$$

なる性質をもつ．

ここで，$h<m$ のときは，すなわち，分割された作業集合の数より荷役機械が少ないときは次のようにして区間の統合を行なう．

まず，G_i の仕事を1台の荷役機械でサービスするのに要する時間を(i)項の計算方法で求めこれを t_i とする．

$$\min(t_i + t_{i+1}) \qquad (i = 1 \sim m-1)$$

を満たす i を選び，それに対する集合 G_i と G_{i+1} を合わせて1つの集合とする．そうして，$m=m-1$，すなわち集合の数を1つ少なくするとともに番号をつけかえる．この操作を $h=m$ となるまでくり返す．

以上より，$h \geq m$ となったわけであるが，ここで各 G_i を1台の荷役機械で処理する時間 t_i が列

車停止時間より小の場合は問題はない．$\max t_i$ が列車停止時間より大きいときは，次のような方法が考えられる．

(1) 類似性のある仕事，すなわち，同じ位置から出発し，同じ方向，距離に行くような仕事を探し，これらを複数台で同時処理したのち，上記区間分割の操作に移る．

(2) 列車到着までにストレイジのコンテナを適当に移動しておき，区間分割でスムーズな処理ができるようにする．

(3) 列車から降ろすコンテナを優先し，列車に積みこむコンテナは，ストレイジ停滞時間，緊急性などから優先度をもたせ，停止時間内で処理できる範囲で処理する．

4. ゲーム的方法との比較（荷役機械1台）

この種問題に一般的に考えられる直観的な方法は，最も手近かなものから処理するというものである．この方法はゲームにおける p 手読み法[4]に相当し，次のように定式化できる．

$\mathfrak{A}=\mathfrak{A}_1, \mathfrak{A}_2, \cdots\cdots, \mathfrak{A}_{p-1}$ を荷物の集合，$\mathfrak{B}_1, \mathfrak{B}_2, \cdots\cdots, \mathfrak{B}_{p-1}$ を空き場所の集合とし，α_i, β_i をそれぞれ，荷物のある場所と空き場所の座標とする．$f_k(\beta_{k-1}, \mathfrak{A}_k, \mathfrak{B}_k)=k-1$ 番目まで荷物の移動が終わり，荷役機械の位置が β_{k-1}，荷物の集合が $\mathfrak{A}_k$，空き場所の集合が $\mathfrak{B}_k$ であるとき，$p-k$ 回の最適な移動に必要な移動量とすると，次の動的計画法の関係式が成立することがわかる．

定理 5.

$$\begin{cases} f_k(\beta_{k-1}, \mathfrak{A}_k, \mathfrak{B}_k) = \min\limits_{\substack{\alpha_k \in \mathfrak{A}_k \\ \beta_k \in \mathfrak{B}_k \circledast \alpha_k}} \{\rho_{\beta_{k-1}}{}^{\alpha_k} + \rho_{\alpha_k}{}^{\beta_k} + f_{k+1}(\beta_k, \mathfrak{A}_k - \{\alpha_k\}, \mathfrak{B}_k - \{\beta_k\} + \{\alpha_k\})\} \\ f_{p-1}(*, *, *) = 0 \end{cases}$$

ただし，$\mathfrak{B}_k \circledast \alpha_k$ は $\mathfrak{B}_k$ の空き場所の中，荷物 α_k の移動可能な空き場所の集合とする．

が成りたつ．

これを $k=p$，$p-1, \cdots\cdots, 1$ に対して順次計算し，

$$\min_{\alpha_1 \in \mathfrak{A}} f_1(\alpha_1, \mathfrak{A}_1, \mathfrak{B}_1)$$

を与える α_1 と β_1 を定める．すると，荷役機械の初期位置は α_1 で，最初の仕事は α_1 のものを β_1 に運ぶことになる．

次に，$\mathfrak{A}=\mathfrak{A}-\{\alpha_1\}$ として上記の動的計画法の計算をし，$f_1(\beta_1, \mathfrak{A}_1, \mathfrak{B}_1)$ に対する α_2, β_2 を決め，β_1 から α_2 に移動し，α_2 のコンテナを β_2 に運ぶ．このステップを $\mathfrak{A}$ が空き集合になるまでくり返すのが p 手読み法である．

いま，荷役機械が i の位置から始めて，$p=n$ とした移動距離を z_i とする．そして $\min(z_1, \cdots\cdots, z_n)$ を満たす移動方法をとると，それがこの問題に対する最適解を与えることは明らかである．しかし，この方法はB-B法と違い，すべての可能性を全部検討すること，コンテナの行き先の割り当ても同時にするため組み合わせの数が非常に多いこと，などによりぼう大な計算時間を要する．$p=1$ または2程度が実用的であろう．

そこで，荷役機械の移動範囲を100 m とし，この中に一様乱数で，列車側，ストレイジ側，そ

れぞれ5個の移動を必要とするコンテナと，5個の空き場所をもつ簡単なモデルを20個作り，一台の荷役機械に対し，本論文で提案したアルゴリズムと，1手読み法，2手読み法を比較した．表2はその結果であるが，これより，本論文で提案した解法がゲーム的方法よりよい結果を出していることがわかる．なお，一般に p 手読み法では p を大きくすると時間は幾何級数的に増大するが，かならずしも評価値が改善されるとは限らない．

表2　ゲーム的方法との移動距離の比較

ケース番号	総移動距離		
	本論文の解法(*1)	1手読み法	2手読み法
1	183m	237m	220m
2	139	171	202
3	178	259	285
4	176	215	185
5	173	180	183
6	157	206	220
7	111	163	136
8	161	277	182
9	207	238	267
10	159	170	193
11	149	234	180
12	165	195	218
13	169	176	195
14	147	158	159
15	142	206	178
16	169	178	185
17	150	219	194
18	146	142	150
19	133	222	172
20	168	185	154
平均	159.1m	201.8m	192.8m
計算機所要時間(*2)	15	1	11

(*1)　暫定解は Hichcock のアルゴリズムで求めた

(*2)　1手読み法の時間を1とした場合

5. 結　び

本論文では，コンテナ専用列車とストレイジ間における荷役機械のスケジューリングの問題をとりあげ実用的な解法を提案した．しかし，一般には荷役機械は，同一線路でトラック専用サービス部分と接続されており（図1），対トラックサービスも行なう必要がある．このためには，列車のいないとき荷役機械はトラック・サービスにあたっており，列車到着前に情報システムから到着列車に関する情報，ストレイジの情報をもらって本論文で提案した対列車のスケジューリングの計算を行なう．この結果何台の荷役機械が，いつ列車サービスのためにどの位置に行っているべきかが求まるので，その指示にしたがって荷役機械を転線させればよい．なお，対トラック専用のサービスは，トラック到着がランダムな性格をもつからヒューリスティックなアルゴリズムが有効であろう．この点に関しては別の機会に論じたい．なお，コンテナ荷役システムの具体的問題を提供してくださり，ご指導いただいた日本国有鉄道鉄道技術研究所貨物輸送研究室，中島博士，前田，大沼両主任研究員，および有益なご討論をいただいた京都大学工学部，茨木俊秀助教授，日立本社，油井副技師長，日立研究所，徳増真司，国友佳男，織田村元視，山本茂の各氏に感謝をいたします．

参考文献

[1] Bellmore, M. and G. Nemhauser, "The Travelling Salesman Problem——A Survey," *Operations Research*, **16**, (1968), 538–558.

[2] Lawler, E. L. and D. E. Wood, "Branch and Bounds——A Survey," *Operations Research*, **14**, (1966), 699–719.

[3] Bellmore, M. and J. C. Malone, "Pathology of Travelling-Salesman Subtour-Elimination Algorithm," *Operations Research*, **19**, (1971), 278–307.

[4] 越智ほか, "電子計算機とゲーム", HITAC ユーザ研究会第6回大会記念論文集, (1969), 71–85.

[5] Little, J. D. C., K. G. Murty *et al.*, "An Algorithm for the Travelling-Salesman Problem," *Operations Research*, **11**, (1963), 972–989.