ヤードクレーンスケジューリング問題における総作業時間の比較研究
～クレーンの交差が可能な場合と不可能な場合～

沼田研究室
和田 研輔

1、はじめに

1-1　本研究の背景

大きな貨物の貿易はコンテナを用いた海運が主流

- **交易貨物の多くが主要なコンテナ港を通って運搬されている**

輸送のコスト削減・時間短縮を目指す

- **輸送時間短縮のためにはターンアラウンドタイムの短縮が必要**

ターンアラウンドタイム：船舶の岸壁滞在時間
多くがコンテナの積み降ろしに費やされる

コンテナ積み降ろしの効率化が必要

1-1-1　コンテナ積み降ろし

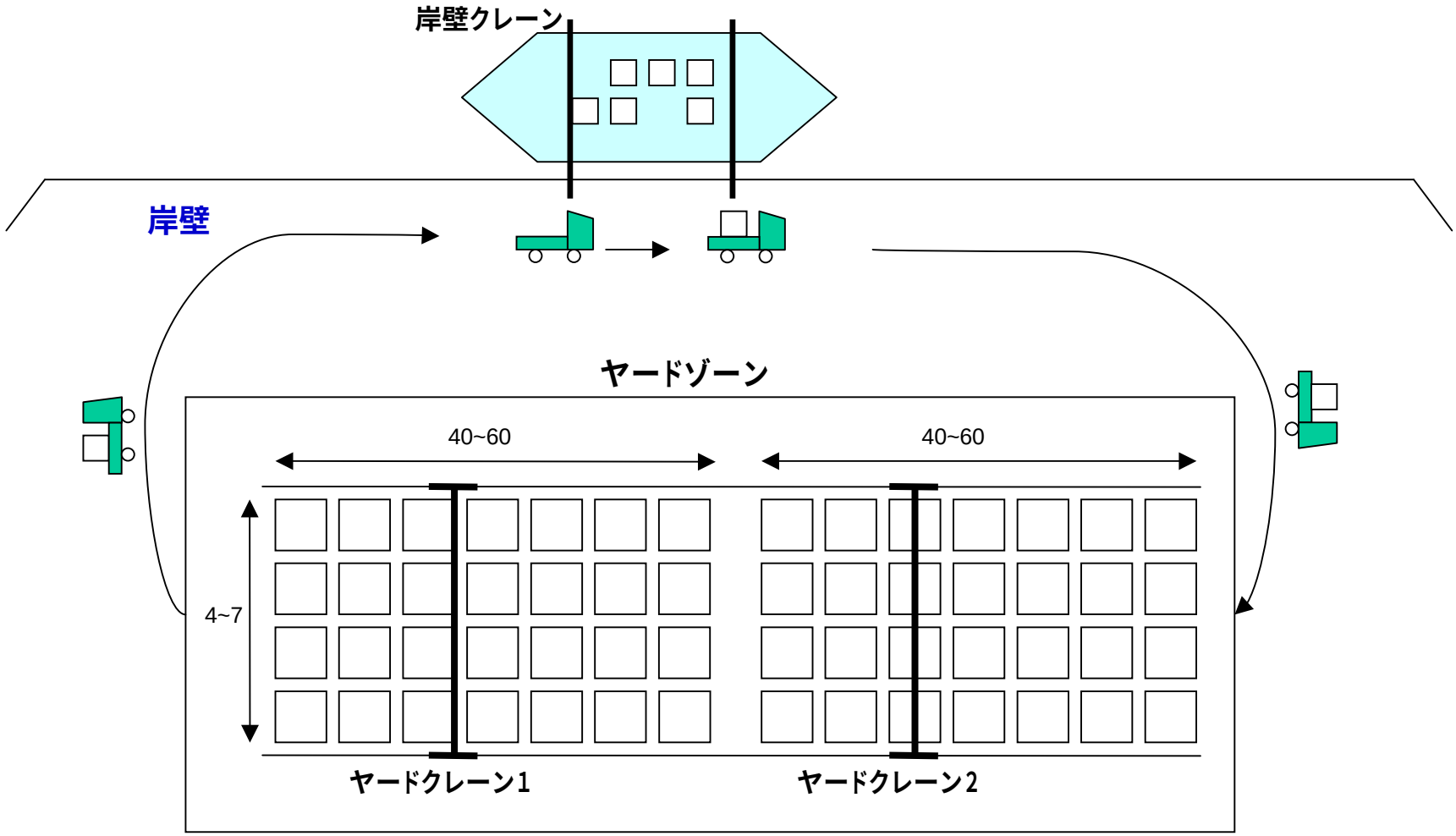


図1-2：コンテナヤード

1-1-2　スケジューリング

ターンアラウンドタイムの短縮

- **岸壁クレーンのスケジューリング**
- **ヤードクレーンのスケジューリング**

を工夫することが考えられる

本研究では、ヤードクレーンのスケジューリングを取り上げる

文献[2]**などの先行研究**

クレーンの交差ができないもとでスケジューリング

文献[1]

交差可能なヤードクレーンの記述が見られる

1-2　研究目的

ヤードクレーンの動作条件が変わることにより，
全コンテナを積み降ろす作業に掛かる時間（総作業時間）
がどの程度変化するかみるため

交差可能な場合と，不可能な場合の総作業時間を比較する

2-1 ヤードクレーンの概要

ヤードクレーン：ヤードゾーンにコンテナを積み降ろす

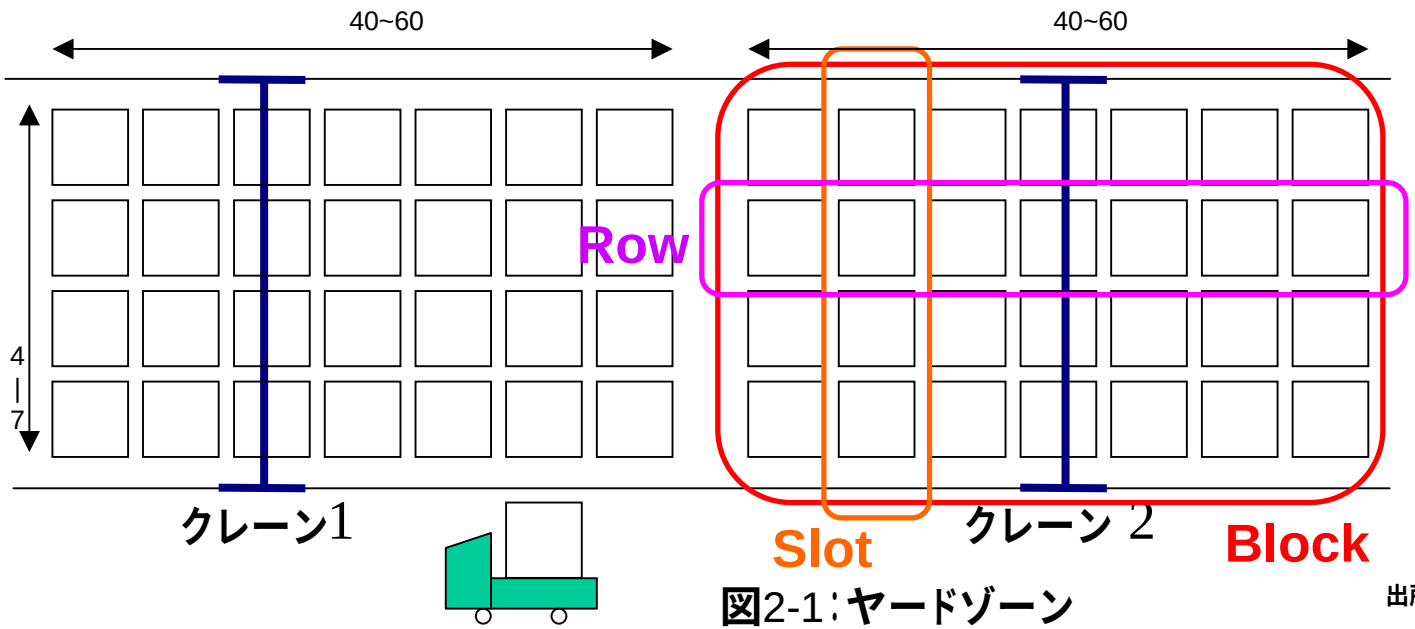


図2-1：ヤードゾーン

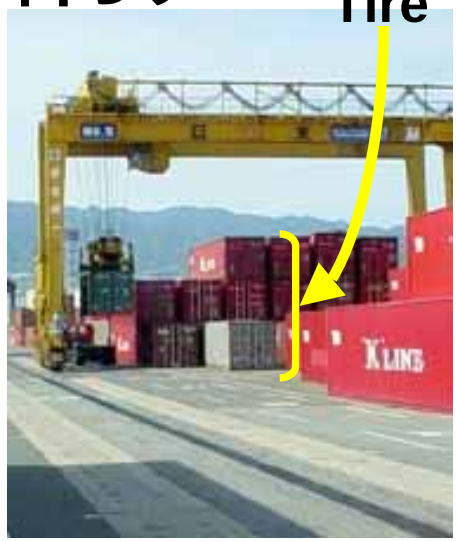


図2-1：ヤードクレーン

出所（http://www.inankuru.com/img/zakki04.jpeg）

各クレーンはレーンを共有している為、交差ができない

本研究では、交差できる場合についても考えていく

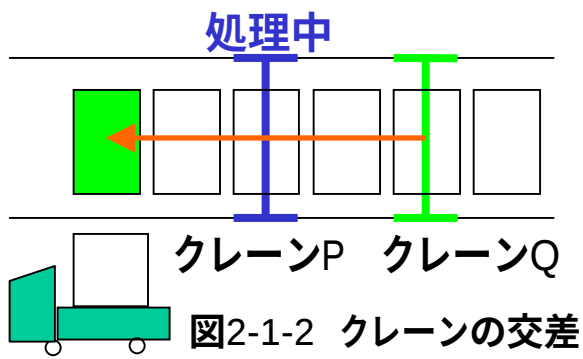


図2-1-2 クレーンの交差

2-2 ヤードクレーンスケジューリング問題

ヤードクレーンスケジューリング問題とは

- **クレーン数**
- **コンテナ数**
- **各コンテナの到着時刻**
- **各コンテナを積み降ろす位置**

が与えられたもとで、

総作業時間が最短となるような各ヤードクレーンのコンテナ処理順序を決定する問題

ただし、本研究でコンテナを積み降ろす位置はSlotのみ扱い、RowやTireは考慮しない（文献[2]同様）

3、定式化

3-1　記号の定義

i　$(i=1,2,\cdots,n)$　：ジョブ番号

k　$(k=1,2,\cdots,m)$　：クレーン番号

θ　$(l=1,2,\cdots,l)$　：スロット番号

α_k　：クレーン k の初期位置

r_i　：ジョブ i のトラック到着時刻

β_i　：ジョブ i のコンテナが運ばれるスロット

H　：ヤードクレーンでのジョブ処理時間

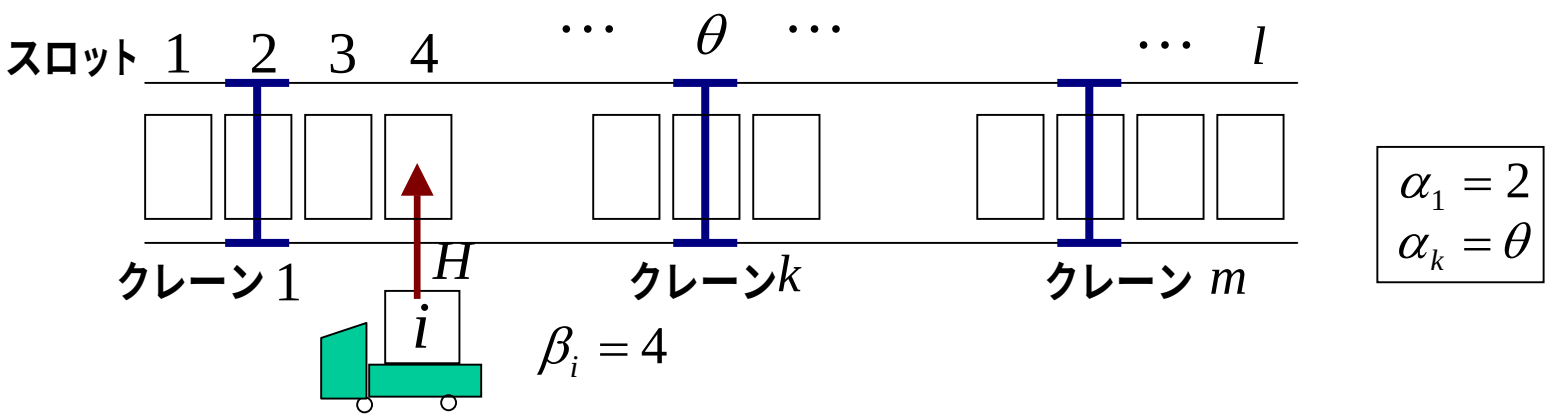


図3-1：記号定義

3-2 目的関数

$$x_{i,p,k} = \begin{cases} 1 & : \text{ジョブ } i \text{ をクレーン } k \text{ で } p \text{ 番目に処理する} \\ 0 & : \text{それ以外} \end{cases}$$

スロット間の移動時間 d_{hi}　　但し $d_{0(k)i}$ ：初期位置 α_k から β_i への移動時間

ジョブ i の処理完了時刻

$$f_i \geq \max\left\{ r_i, \quad \sum_{k=1}^{m} \left\{ d_{0(k)i} \cdot x_{i,1,k} + \sum_{h=1}^{n} \sum_{p=2}^{n_k} \left(f_h + d_{hi} \right) \left(x_{h,p-1,k} \cdot x_{i,p,k} \right) \right\} \right\} + H$$

前のジョブの完了時刻

あるクレーンでジョブ h の後 i を処理するとき 1

但し $f_0 = 0$　　$x_{0,q,k} = \begin{cases} 1 & q = 0 \\ 0 & q \neq 0 \end{cases}$

n_k ：クレーン k が処理するジョブ個数

目的関数 $\min \quad \max_{1 \leq i \leq n} \{ f_i \}$

3-3 制約条件

$$\text{s.t.} \quad \sum_{i=1}^{n} x_{i,p,k} = 1 \qquad \forall k, p \qquad (1)$$

$$\sum_{k=1}^{m} \sum_{p=1}^{n_k} x_{i,p,k} = 1 \qquad \forall i \qquad (2)$$

$$\left| f_i - f_j \right| \geq H \qquad \forall i, j \qquad (\beta_i = \beta_j) \qquad (3)$$

$$x_{i,p,k} \in \{0,1\} \qquad \forall i, p, k \qquad (4)$$

(1) **あるクレーンのある順番で処理されるジョブはただ1つ**
(2) **全てのジョブはどれかのクレーンのどこかの順番で処理される**
(3) **同じスロットで複数のクレーンが同時に処理を行わない**
(4) **決定変数は0か1である**

4-1 計算量

組合せ最適化問題

ヤードクレーンスケジューリング問題 → **列挙法により最適解を算出できる**

- m 台のクレーンに n 個のジョブを割り当てる方法　　m^n 通り
- 割り当てられたジョブからクレーン1台の最短処理完了時刻を求める

最悪の場合　$n!$ 通り

n が大きくなると列挙法により許容時間内で解く事は困難

解法を工夫する必要がある

本研究では，最適性の保証はないが良質の解をできるだけ短い時間で求められる発見的解法を用いて解く．

4-2 動作条件の違いに対する解法

- クレーンの交差が不可能な場合の解法
  - 文献[2]の解法を用いる
- クレーンの交差が可能な場合の解法
  - 2つ解法（解法A，解法B）を提案する．
    - 解法A：文献[2]をもとに，条件を緩和したもの
    - 解法B：新たに提案する貪欲的に解を構成する

4-2-1 文献[2]の解法

- **ヤードクレーンスケジューリング問題を解くために、文献[2]では2段階の発見的解法を提案している**
  - **第1段階**
    - **ヤードゾーンを分割することによって、m個の独立したヤードクレーンスケジューリング問題に簡単化する**
  - **第2段階**
    - **第1段階で簡単化しため良いスケジュールが失われる可能性があるので，第1段階で得られるスケジュールをもとに、ジョブの入れ替えを行い，良い解を探索する**

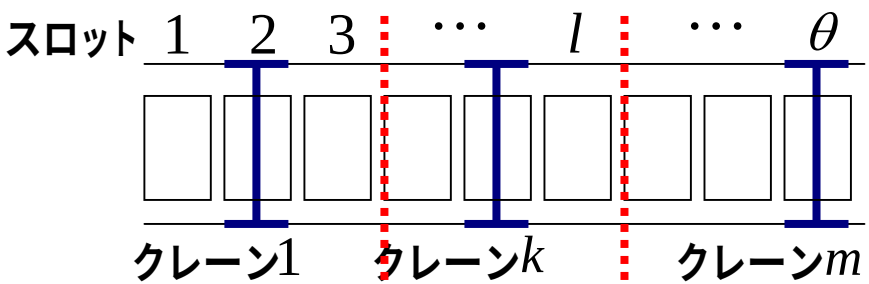


図4-2-1-1：第一段階

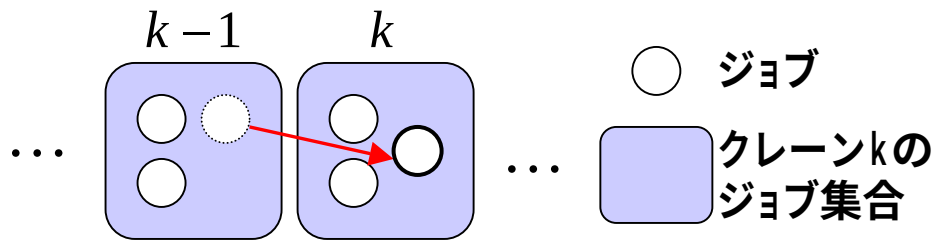


図4-2-1-2：第二段階

4-2-2 解法A

文献[2]の解法と同様，2段階に分けてに解く

ただし、

・第2段階でジョブの入れ替えを行う相手クレーン

文献[2]	隣のクレーンと限定
↓	
解法A	任意のクレーン同士で可能

・入れ替えを許す条件

文献[2]	クレーンの交差が起きない
↓	
解法A	クレーンが同時刻に同Slotで処理を行わない

4-2-3 解法B

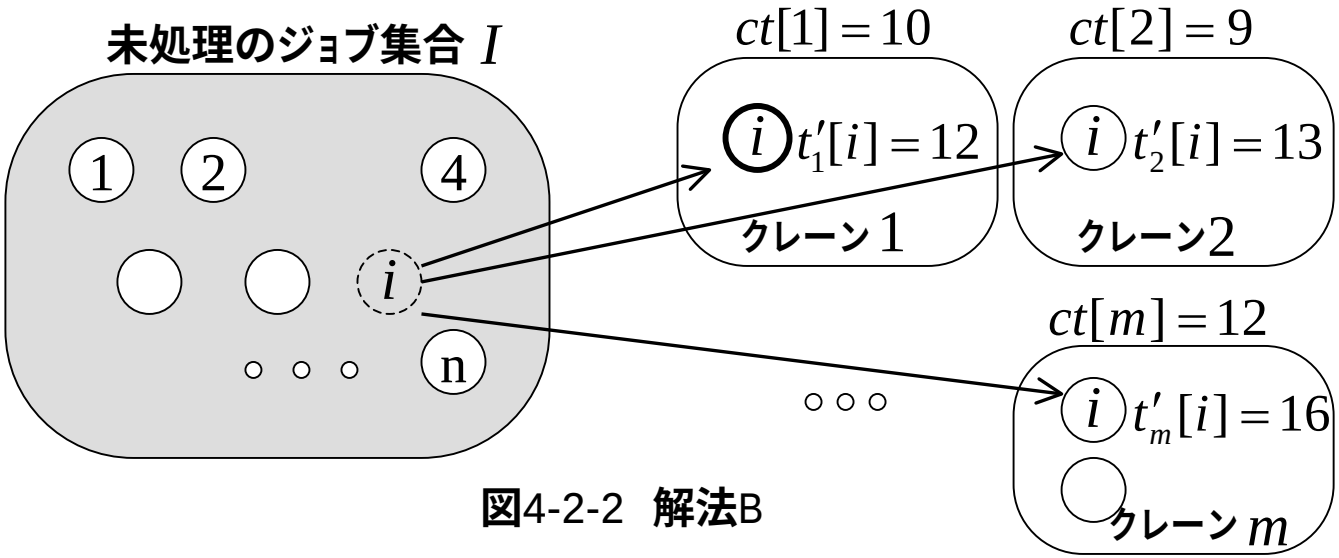


図4-2-2 解法B

①全てのクレーンに対して，そのとき未処理のジョブを処理し終わる時刻を各ジョブについて算出する
② (クレーン数) × (未処理のジョブ数) 個の中で最も小さい値を持っているジョブとクレーンの組合せで処理する

すべてのジョブの処理が終わるまで繰り返す

5、数値実験

5-1　実験概要

ジョブ到着時刻
ジョブ到着Slot
} ランダムで与えたジョブ100個生成
＝ジョブリスト

Slot数10と40の場合、各10個のジョブリストを作成

各解法（文献[2]、解法A、解法B）にもとづき作成したプログラムで全てのジョブを処理し終わるまでに掛かる時間を算出，比較

クレーン数：3台
ジョブ1つの処理時間：$H = 3$（一定）
Slot間移動時間：$d_{hi} = \left|\beta_h - \beta_i\right|$

プログラムはBorland社製Delphi6で作成

5-2 結果・考察

計算結果

表5-2-1 数値実験結果（Slot数：10）

実験番号	文献[2]	解法A	比A	解法B	比B
1	120	110	0.917	113	0.942
2	111	110	0.991	125	1.126
3	122	111	0.910	121	0.992
4	125	124	0.992	126	1.008
5	117	112	0.957	128	1.094
6	118	119	1.008	126	1.068
7	118	112	0.949	116	0.983
8	111	111	1.000	119	1.072
9	120	120	1.000	125	1.042
10	123	122	0.992	127	1.033
平均	118.5	115.1	0.972	122.6	1.036

表5-2-2 数値実験結果（Slot数：40）

実験番号	文献[2]	解法A	比A	解法B	比B
1	146	146	1.000	160	1.096
2	149	149	1.000	167	1.121
3	125	125	1.000	143	1.144
4	125	125	1.000	132	1.056
5	150	150	1.000	164	1.093
6	151	148	0.980	166	1.099
7	216	216	1.000	223	1.032
8	227	227	1.000	245	1.079
9	467	467	1.000	471	1.009
10	475	475	1.000	475	1.000
平均	223.1	222.8	0.998	234.6	1.073

比A：解法A/文献[2]　比B：解法B/文献[2]

5-2 結果・考察

Slot数	文献[2]	解法A	解法B
10	118.5	115.1	122.6
40	223.1	222.8	234.6

表5-2-1 総作業時間の平均

Slot数	比A	比B
10	0.972	1.036
40	0.998	1.073

表5-2-2 比の平均

比A：解法A/文献[2]　比B：解法B/文献[2]

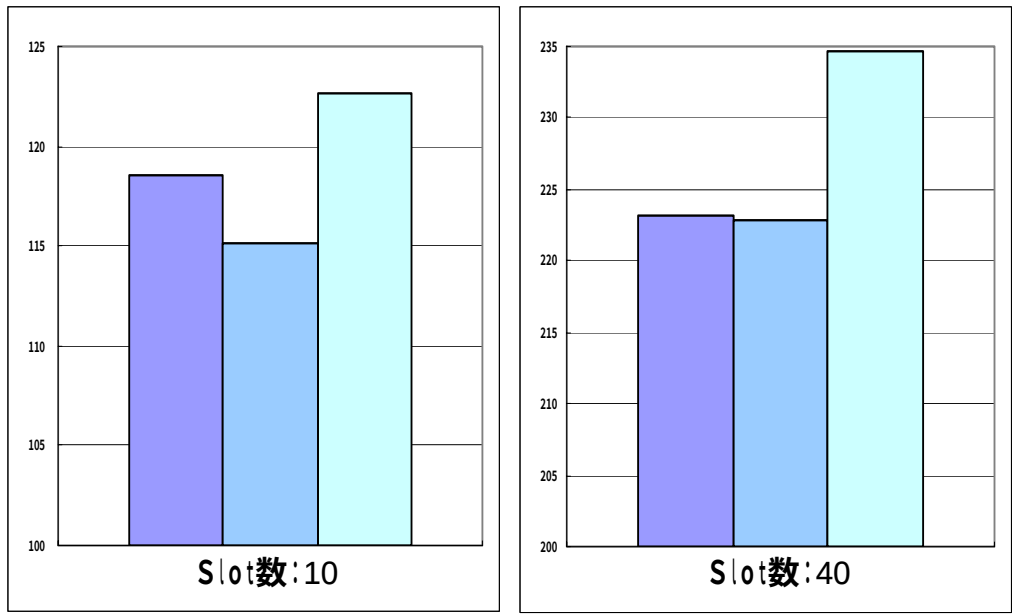


図5-2-1 総作業時間の平均

数値実験の結果より、文献[2]と比較して

解法A　時間の短縮は大きいものではない
解法B　解法の精度が十分でない

ことがわかった

5-2 結果・考察

クレーンの交差が起こる状況

クレーンP：処理中
クレーンQ：新たなジョブの処理に向かえる状態

クレーンQとジョブ i の間にクレーンPがある

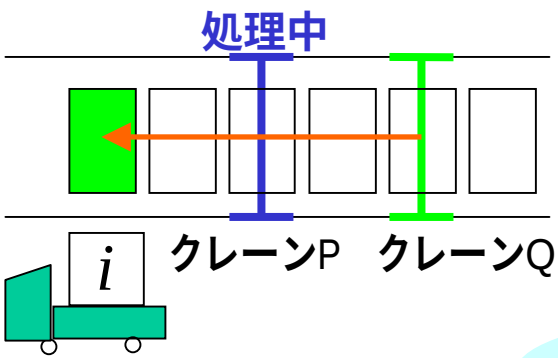


図5.2.2 クレーンの交差

クレーンQがジョブ i を処理するためには、クレーンPの処理状態が終わる前にPを超えなければならない

クレーンPの処理後に超えるならばクレーンPでジョブ i を処理するほうが早く完了する

交差が起こる
→クレーン2台が近いSlotで処理を行っている

5-2 結果・考察

クレーンの交差が起こる状況

クレーンが隣接しているとき交差が起こる可能性がある

クレーンの数に対してSlot数が多くなるとクレーンが隣接している可能性は少なくなる

→ クレーンが交差する可能性も減る

Slot数	比A	比B
10	0.972	1.036
40	0.998	1.073

表5-2-2 比の平均

比の平均をSlot数10と40で比較しても見て取れる.

実際の状況ではクレーン数に対してSlot数が多いため、クレーンの交差が可能であることの効果は少ないと考えられる.

6、まとめ

結論

クレーンの交差が可能になることで総作業時間に
与える効果は少ない

今後の課題

交差不可能な場合を、より現実に近づけるため

- RowやTireも考慮し、処理時間H を積み降ろす位置によって変える
- ジョブの到着をリアルタイムで把握し、処理順序を決定する

ことが考えられる.

7、参考文献

[1] 星野智史,太田順,篠崎朗子,橋本英樹:“港湾物流搬送システムのためのコンテナ蔵置計画とエージェント行動即設計”, 第17回 自律分散システム・シンポジウム,pp.75-80（2005）

[2] W.C.Ng¦ “Crane scheduling in container yards with inter-crane interference”, European Journal of Operational Research 164,pp.64-78 (2005)

[3]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）

[4]掌田津耶乃：「Delphi パーソナルプログラミング」, 毎日コミュニケーションズ (2002)

抄録の訂正

・pp130，18**行目の**(3.1)**式**

(誤) $\min \max_{1 \le i \le i} \{ f_i \}$

↓

(正) $\min \max_{1 \le i \le n} \{ f_i \}$

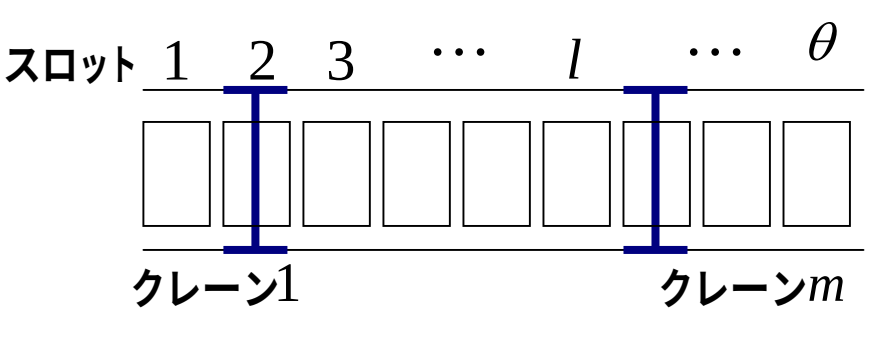
付録

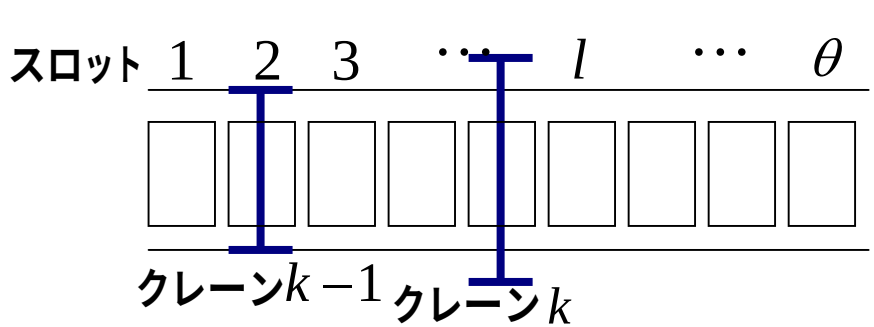
本研究の位置づけ

- スケジューリングの対象
  - 岸壁クレーン
  - ヤードクレーン

ヤードクレーン	交差不可能	交差可能
1台		
複数	文献[2]	本研究

比較





発見的解法 (第一段階)

- 各クレーンが受け持つスロットを範囲で分割し，そこに含まれるジョブを1台のクレーンで全て完了するのにかかる時間を計算する
- 分割方法を変えていき，最もジョブ処理完了時間が短いものを第一段階での解とする

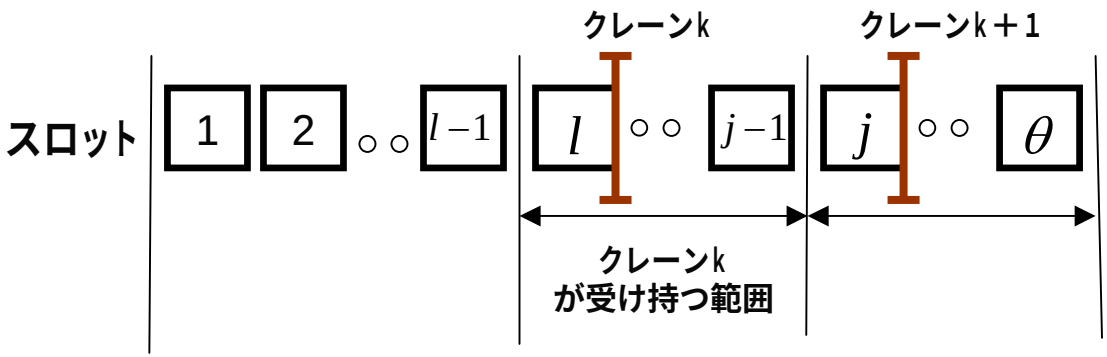


図3：スロットの分割

発見的解法（第二段階）

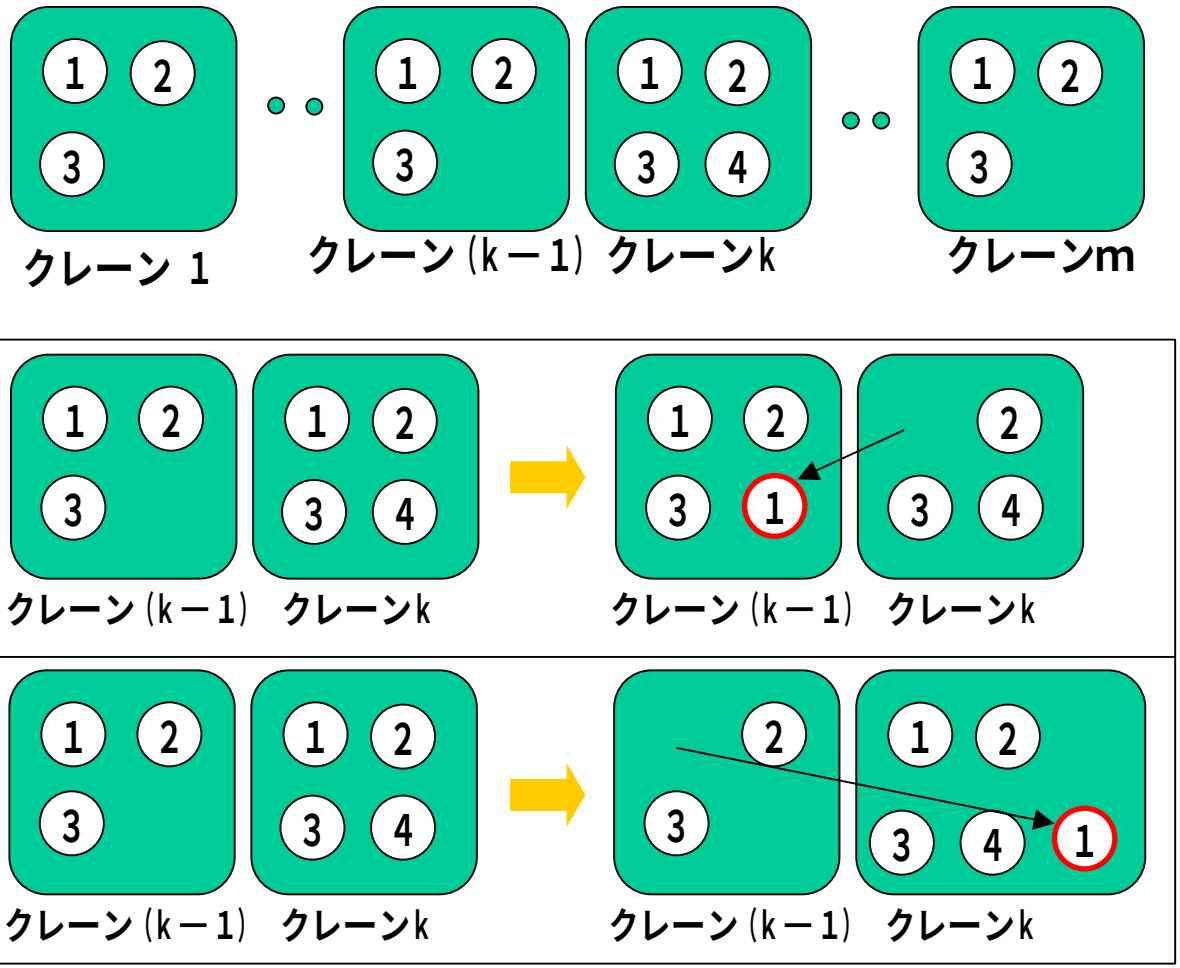


図4：ジョブの再割り当て

より良いスケージュルを見つけるため、前後のクレーンのジョブ集合にジョブ移して行き、作業時間が短縮された場合、ジョブ集合を更新する

これを全てのジョブに対して行う

：ジョブ

：あるクレーンの仕事の集合

近似解法と発見的解法

本研究では，最適性の保証はないが良質の解をできるだけ短い時間で求められる

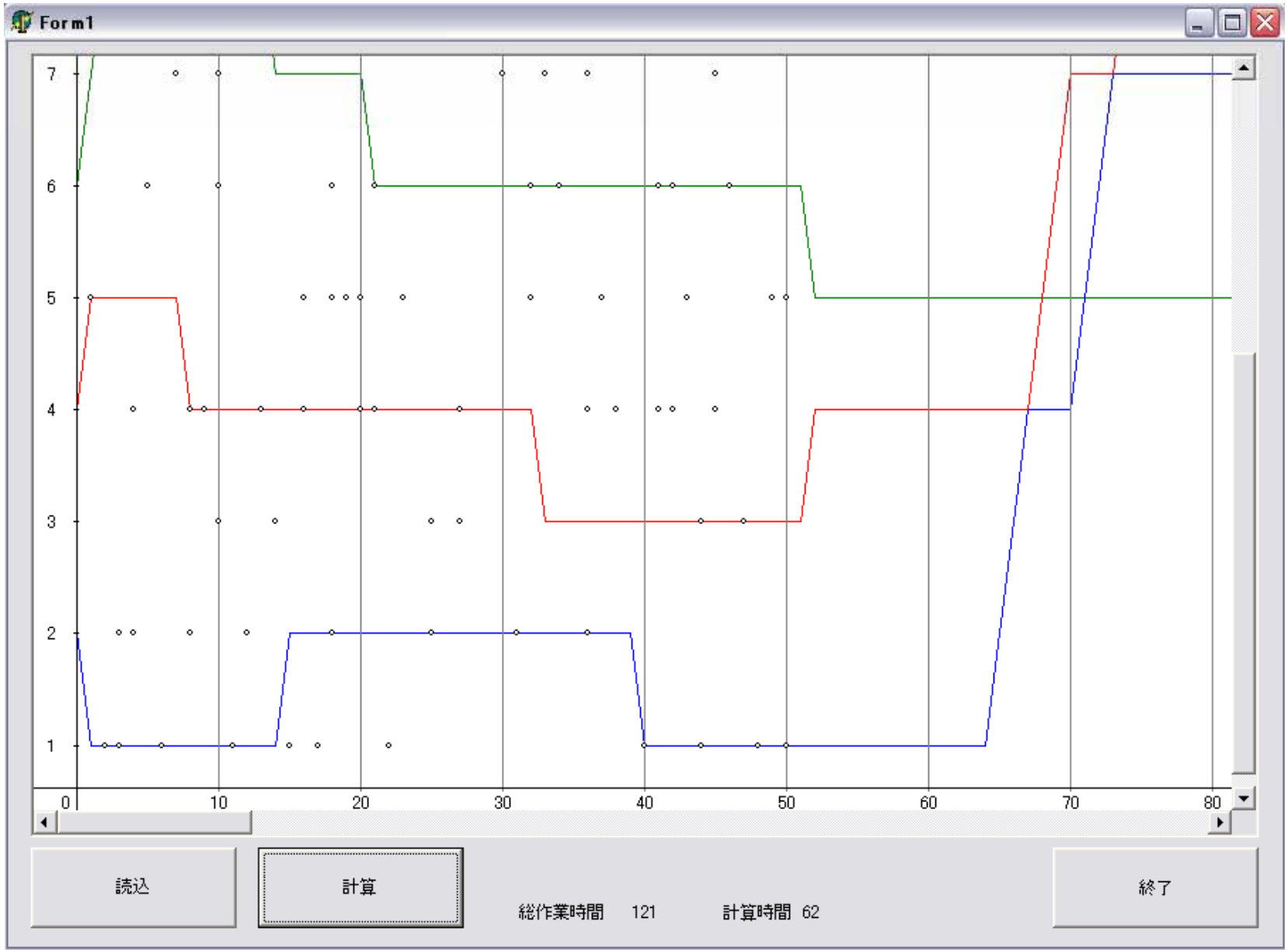
ある解法によって求められる解が，

最適解からある程度の距離しか離れていない事が保証できるとき　**近似解法**

そのような保証ができないとき　**発見的解法**

と呼ぶ事が多い．

ただし、「近似解法」と「発見的解法」の使い分けは，
研究者の間でも必ずしも同意されている訳ではない．


Form1
7
6
5
4
3
2
1
0
10
20
30
40
50
60
70
80
読込
計算
総作業時間 121
計算時間 62
終了

定式化（文献[2]）

$$\min \quad \sum_{i=1}^{n}\sum_{k=1}^{m}\sum_{t=0}^{\pi} tW_{i,k,t} \tag{1}$$

$$s.t \quad \sum_{k=1}^{m}\sum_{t=0}^{\pi} tW_{i,k,t} \geq r_i + h \qquad i=1,2,\cdots,n \tag{2}$$

$$\sum_{k=1}^{m}\sum_{t=0}^{\pi} W_{i,k,t} = 1 \qquad i=1,2,\cdots,n \tag{3}$$

$$\sum_{j=0}^{h} Y_{k,\beta_i,t-j} - h - 1 \geq M(W_{i,k,t} - 1) \qquad k=1,2,\cdots,m : i=1,2,\cdots,n : t=h,h+1,\cdots,\pi \tag{4}$$

$$Y_{k,l,t} \leq \sum_{j\in p(l)} Y_{k,j,t-1} \qquad k=1,2,\cdots,m : l=1,2,\cdots,\theta : t=1,2,\cdots,\pi \tag{5}$$

$$Y_{k,l,t} \leq \sum_{j\in s(l)} Y_{k,j,t+1} \qquad k=1,2,\cdots,m : l=1,2,\cdots,\theta : t=1,2,\cdots,\pi-1 \tag{6}$$

$$M(1-Y_{k,l,t}) \geq \sum_{q=l}^{\theta} Y_{k-1,q,t} \qquad k=2,3,\cdots,m : l=1,2,\cdots,\theta : t=1,2,\cdots,\pi \tag{7}$$

$$\sum_{l=1}^{\theta} Y_{k,l,t} = 1 \qquad k=1,2,\cdots,m : t=1,2,\cdots,\pi \tag{8}$$

$$\sum_{t=h}^{\pi} t(W_{j,k,t} - W_{i,k,t}) \geq h - M(1-X_{i,j,k}) \qquad i,j=1,2,\cdots,n : i\neq j : k=1,2,\cdots,m \tag{9}$$

$$\sum_{t=h}^{\pi} (W_{j,k,t} + W_{i,k,t}) - 1 \leq X_{i,j,k} + X_{j,i,k} \qquad i,j=1,2,\cdots,n : i\neq j : k=1,2,\cdots,m \tag{10}$$

$$X_{i,j,k}, W_{i,k,t}, Y_{k,l,t} \in \{0,1\} \qquad i,j=1,2,\cdots,n : k=1,2,\cdots,m : t=1,2,\cdots,\pi : l=1,2,\cdots,\theta \tag{11}$$

目的関数

- ジョブ i のトラックの待ち時間

$$\sum_{k=1}^{m}\sum_{t=1}^{\pi} tW_{i,k,t} - h - r_i$$

- 全トラックの待ち時間

$$\sum_{i=1}^{n}(\sum_{k=1}^{m}\sum_{t=0}^{\pi} tW_{i,k,t} - h - r_i)$$

  - ここで h や r_i は定数なので

$$\sum_{i=1}^{n}\sum_{k=1}^{m}\sum_{t=0}^{\pi} tW_{i,k,t}$$

の最小化を考える

(2)式は，ジョブの完了時刻はその到着時刻に処理時間を足したものより大きくなることを表している．
(3)式は，ジョブの完了時刻は1度であり，処理するクレーンも1台であることを表している．
(4)式は，ジョブの処理中クレーンはSlotにいなければならないことを表している．
(5)(6)式は，クレーンは急に現れたり，消えたりしてはいけないことを表している．
(7)式は，クレーンはクレーンより常に小さいSlot番号にいなくてはならないことを表している．
(8)式は，クレーンはある時刻には必ずどこかのSlotにいることを表している．
(9)式は，同じクレーンで続けて処理される2つのジョブの完了時間の差は h 以上であることを表している．
(10)式は，同じクレーンで処理されるジョブの関係を表している．
(11)式は，各決定変数がかであることを表している．

計算時間

	文献[2]	解法A	解法B
文献[2]	-	1.367	0.405
解法A	0.732	-	0.296
解法B	2.471	3.377	-