スタック構造を有する造船大組ブロックストックヤードのスケジューリング最適化

九州大学大学院工学研究院
木村　元
トヨタ自動車株式会社
酒井 栄輔

日本船舶海洋工学会春季講演会
2012年5月17日～18日＠神戸

本発表の概要

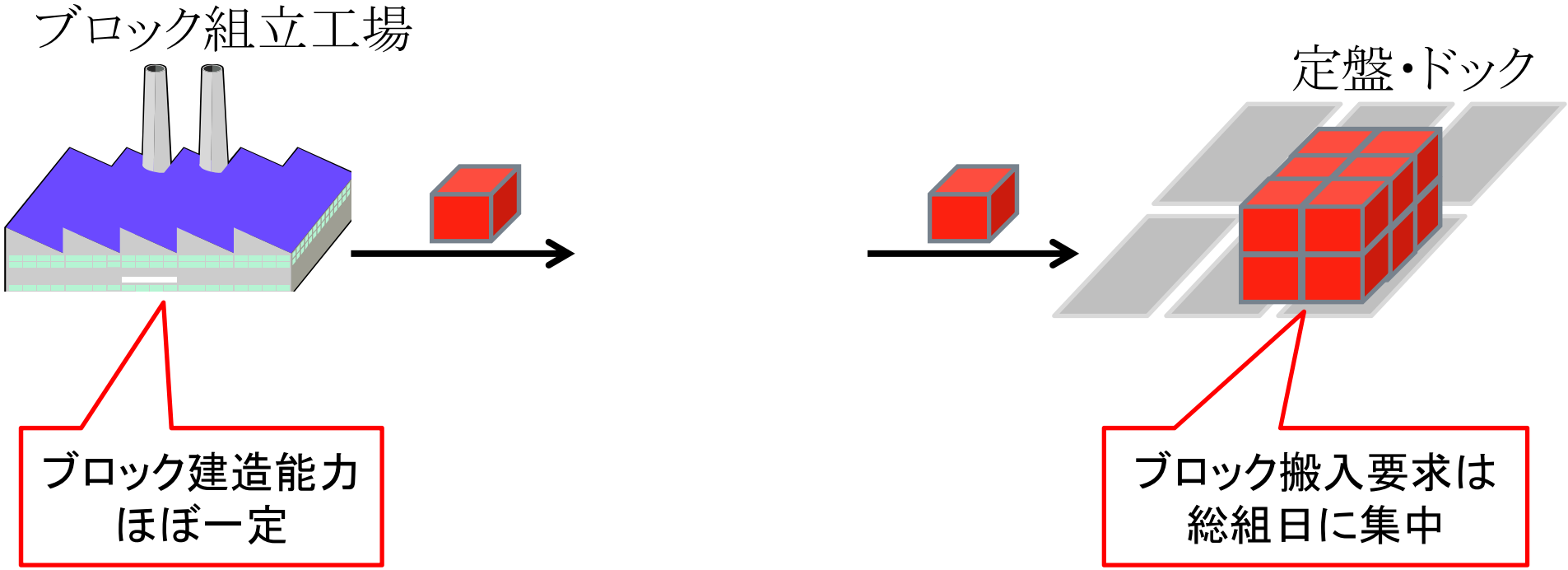
1. 背景と目的
   - 造船所のストックヤードの現状と問題
2. アプローチ
   - 造船所の置場情報とブロック蔵置ルールの設定
   - ブロック割り当てアルゴリズム
3. 結果と考察
   - 従来手法との比較
   - サンプルデータの生成と検討
4. 実用化への取り組み
   - リスケジューリングと属性情報の追加
5. まとめと今後の課題

本発表の概要

1. 背景と目的
   - 造船所のストックヤードの現状と問題
2. アプローチ
   - 造船所の置場情報とブロック蔵置ルールの設定
   - ブロック割り当てアルゴリズム
3. 結果と考察
   - 従来手法との比較
   - サンプルデータの生成と検討
4. 実用化への取り組み
   - リスケジューリングと属性情報の追加
5. まとめと今後の課題

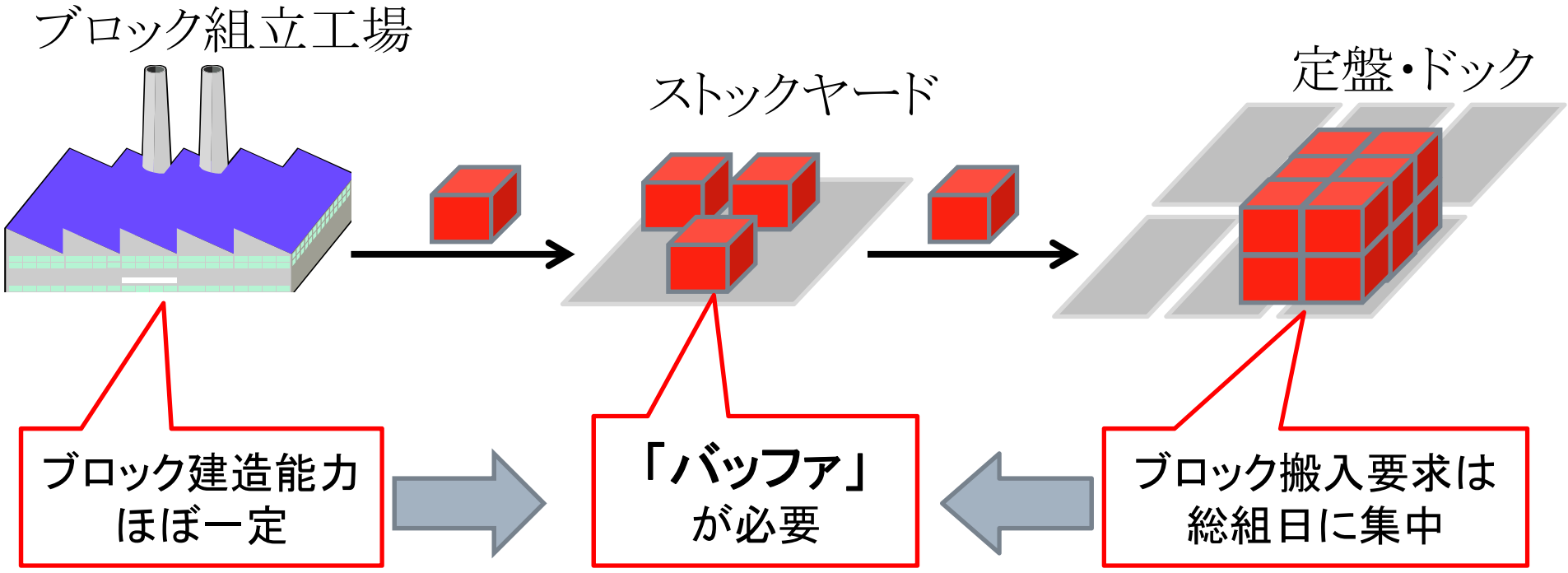
背景(1)

1．ストックヤードの必要性



背景(1)

1．ストックヤードの必要性



背景(2)

1．造船所のブロックストックヤードの特徴

ストックヤードに蔵置される船舶建造ブロックは…

1） 巨大・重量物である

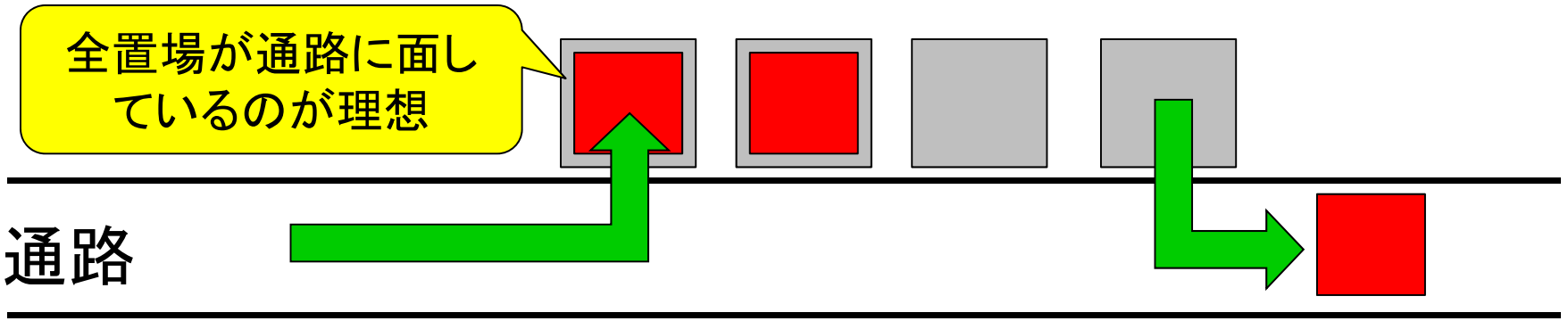
2） (自動車部品のように)ラックにストックできない

3） (コンテナのように)積み重ねたりできない

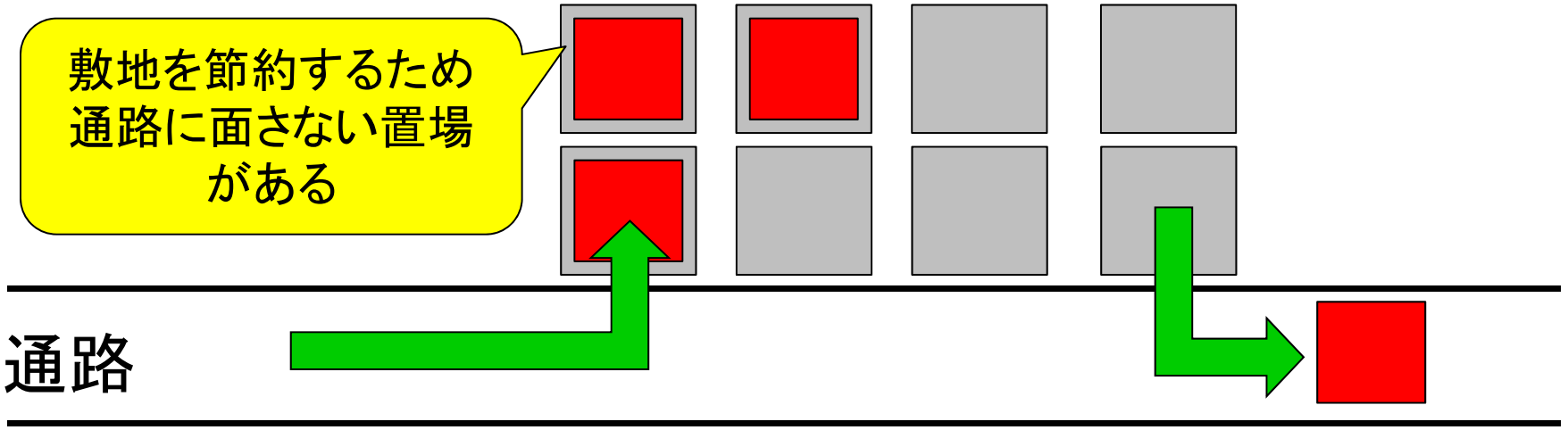
広大な敷地が必要

背景

- 理想的なブロックストックヤード

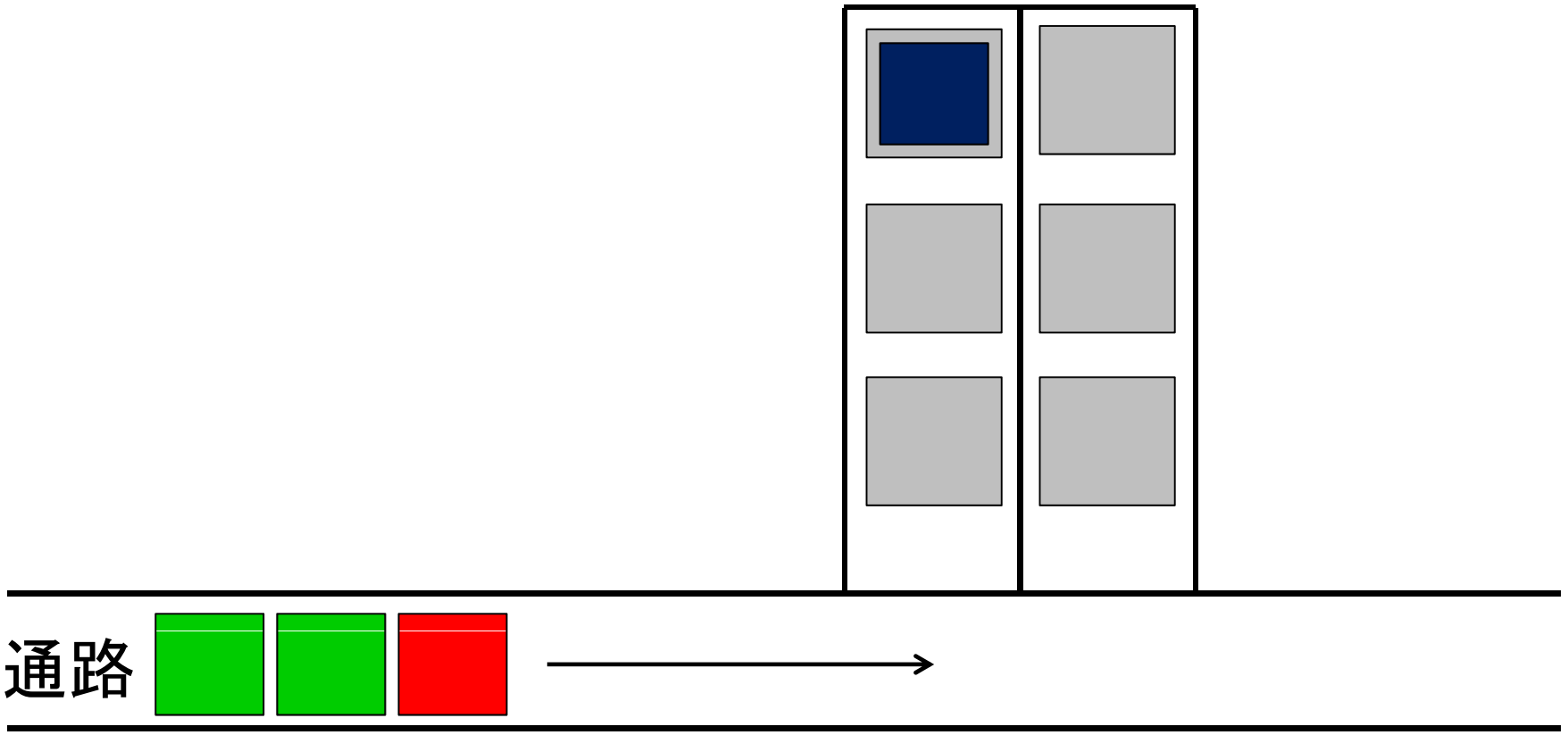


- 実際のブロックストックヤード



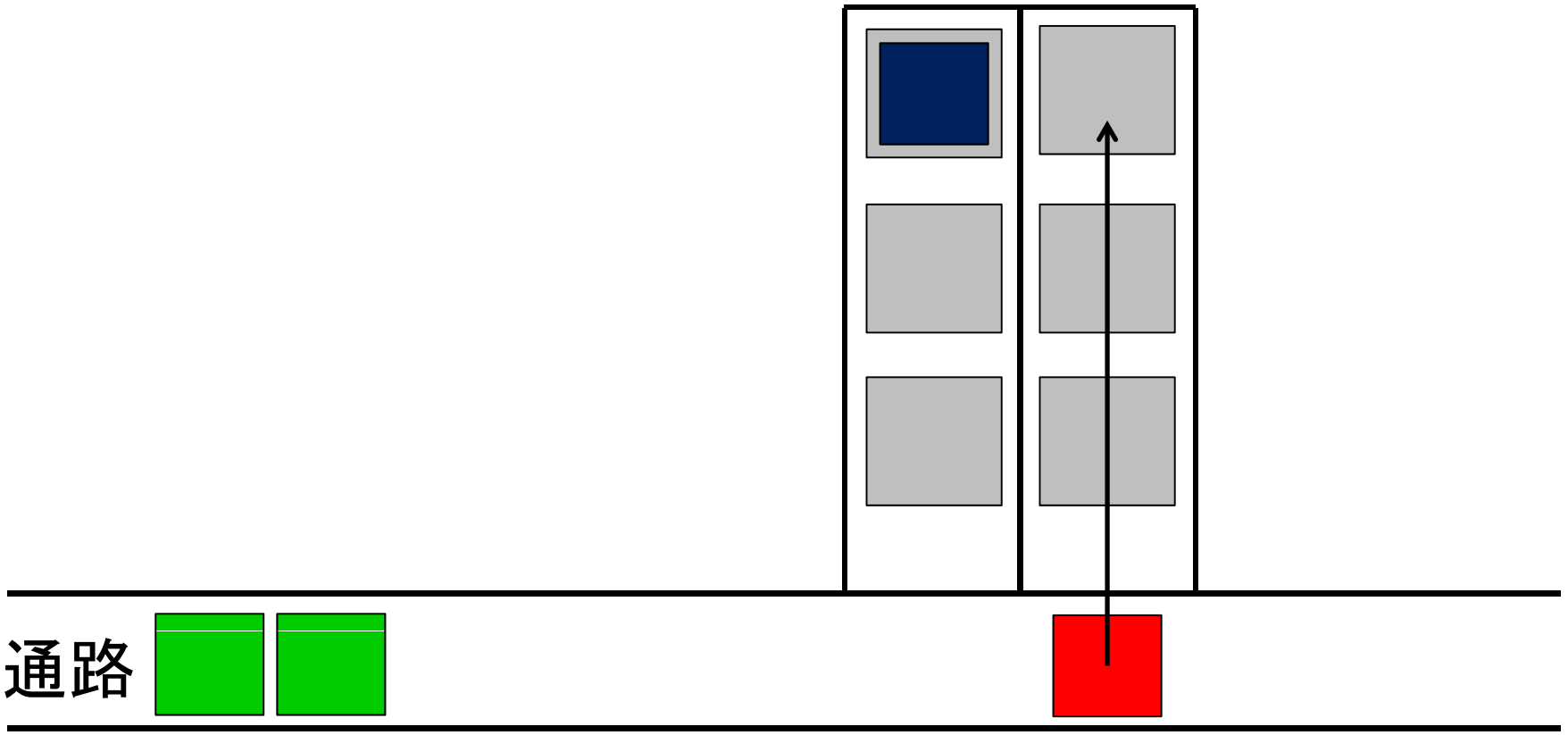
背景

- 先入れ後出し(First In Last Out)の**スタック構造**



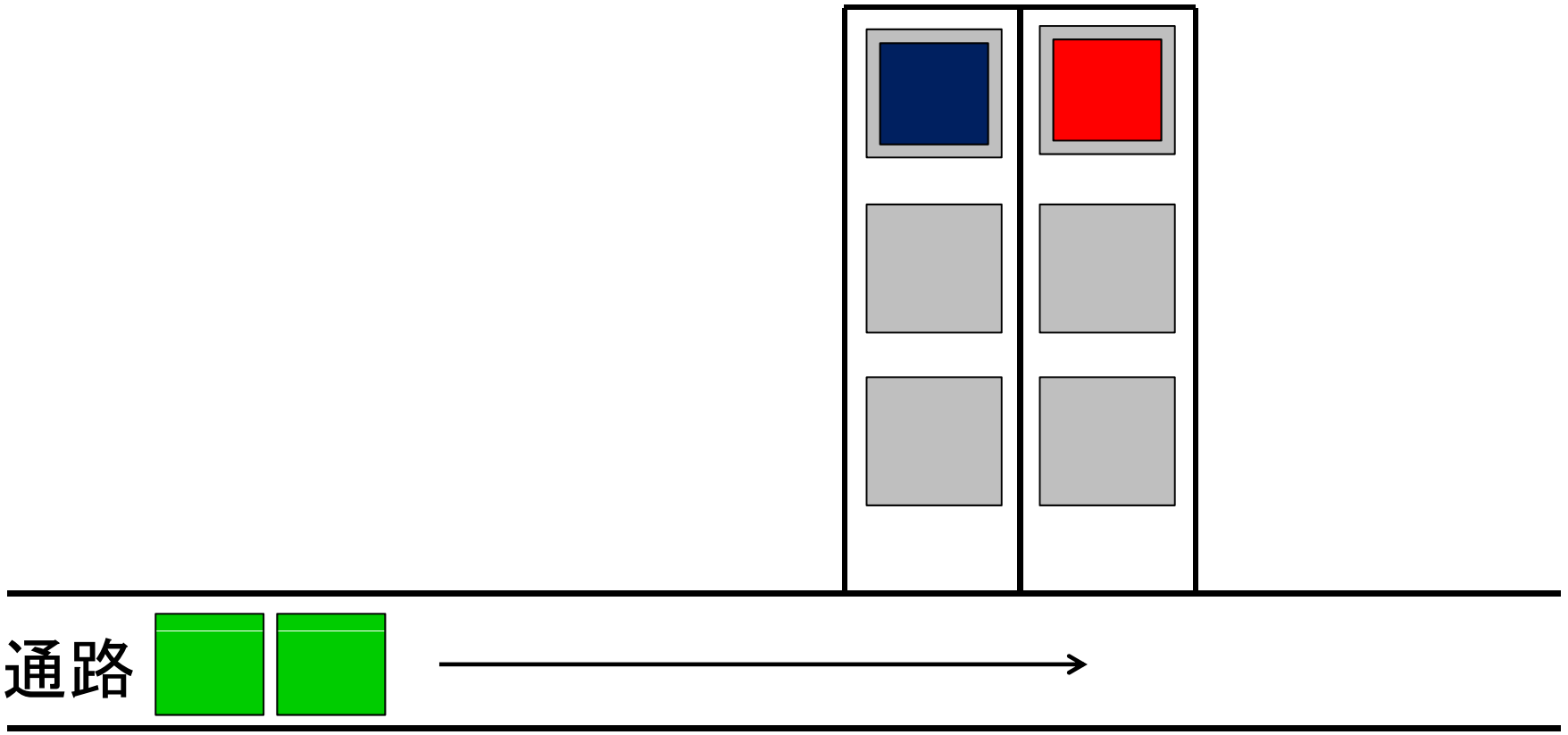
背景

- 先入れ後出し(First In Last Out)の**スタック構造**



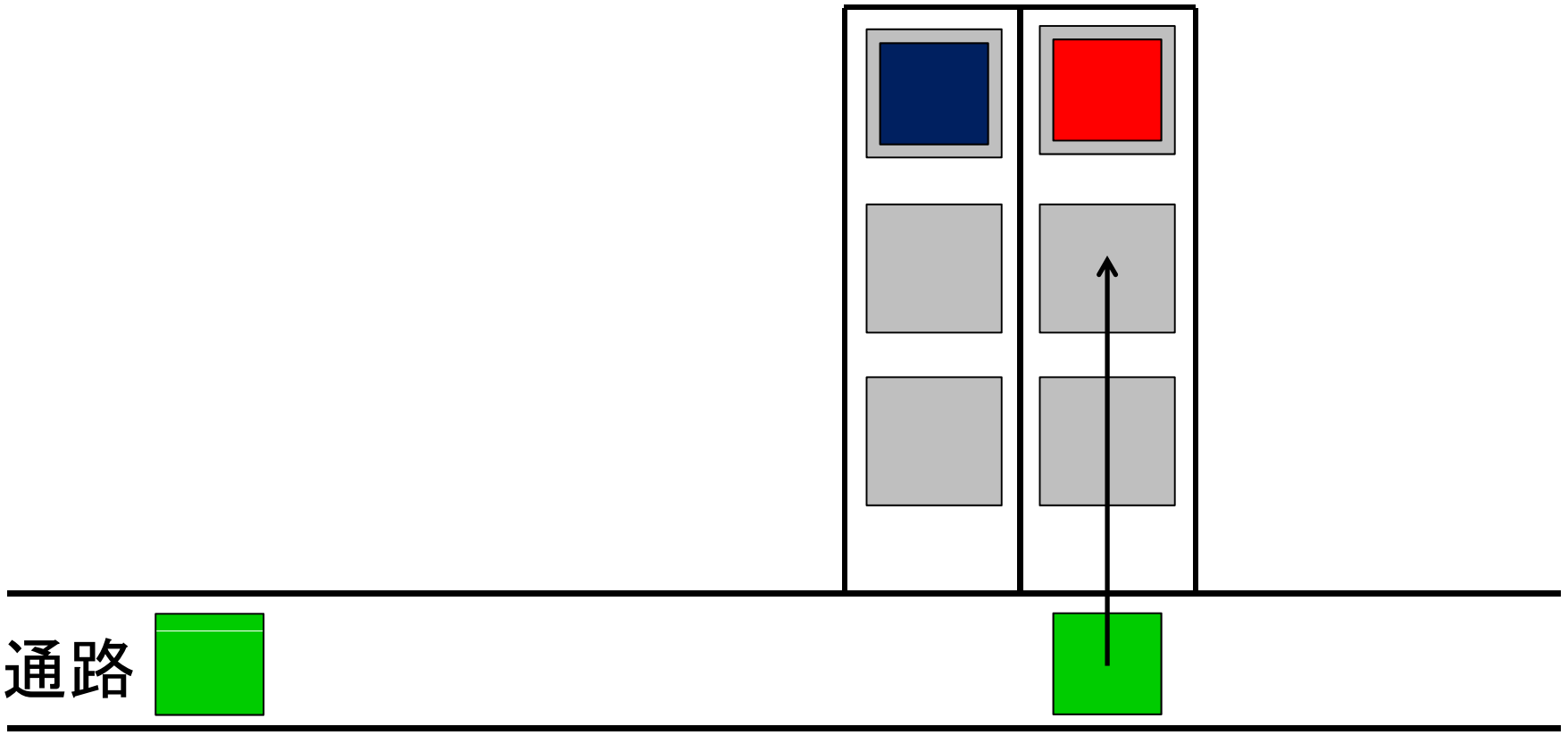
背景

- 先入れ後出し(First In Last Out)の**スタック構造**



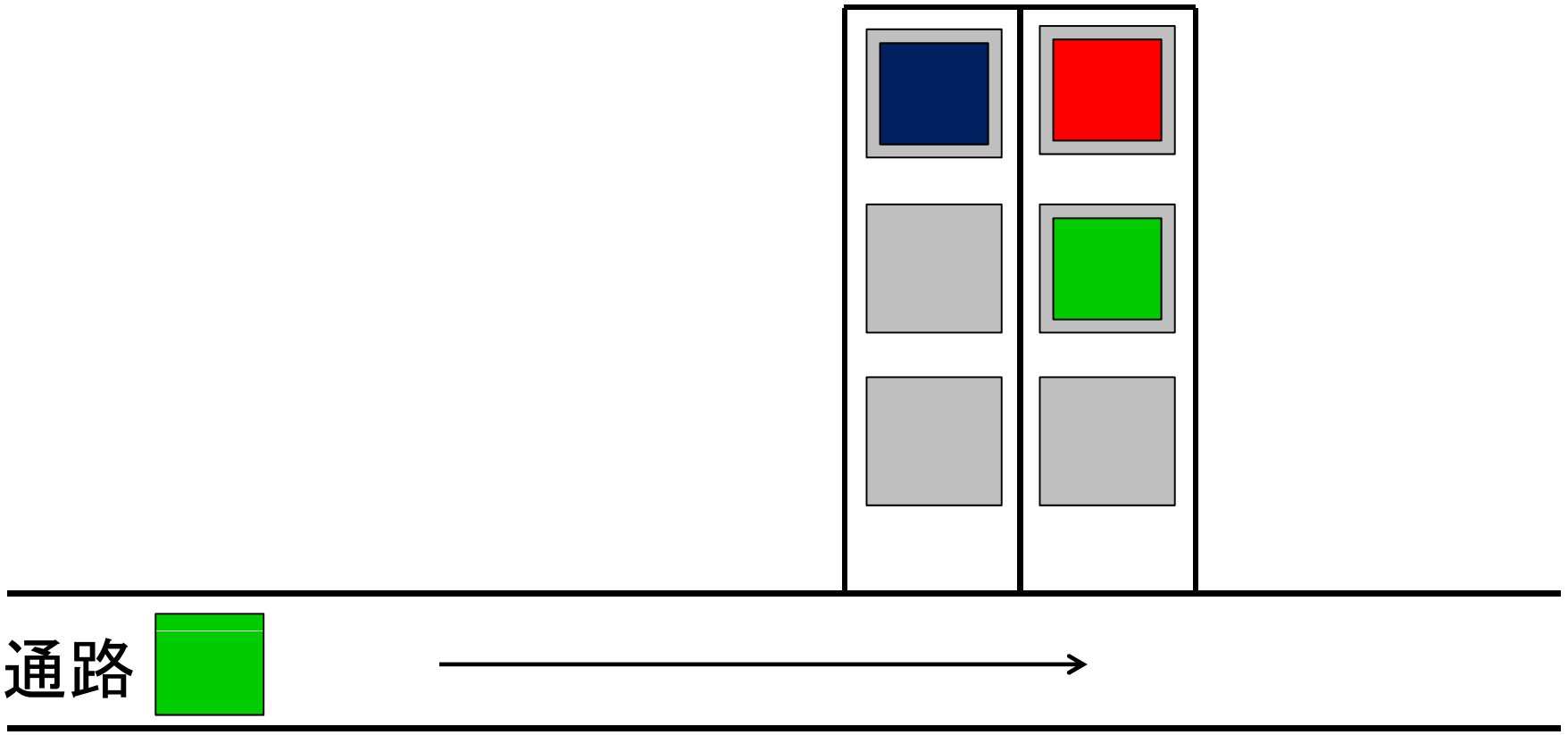
背景

- 先入れ後出し(First In Last Out)の**スタック構造**



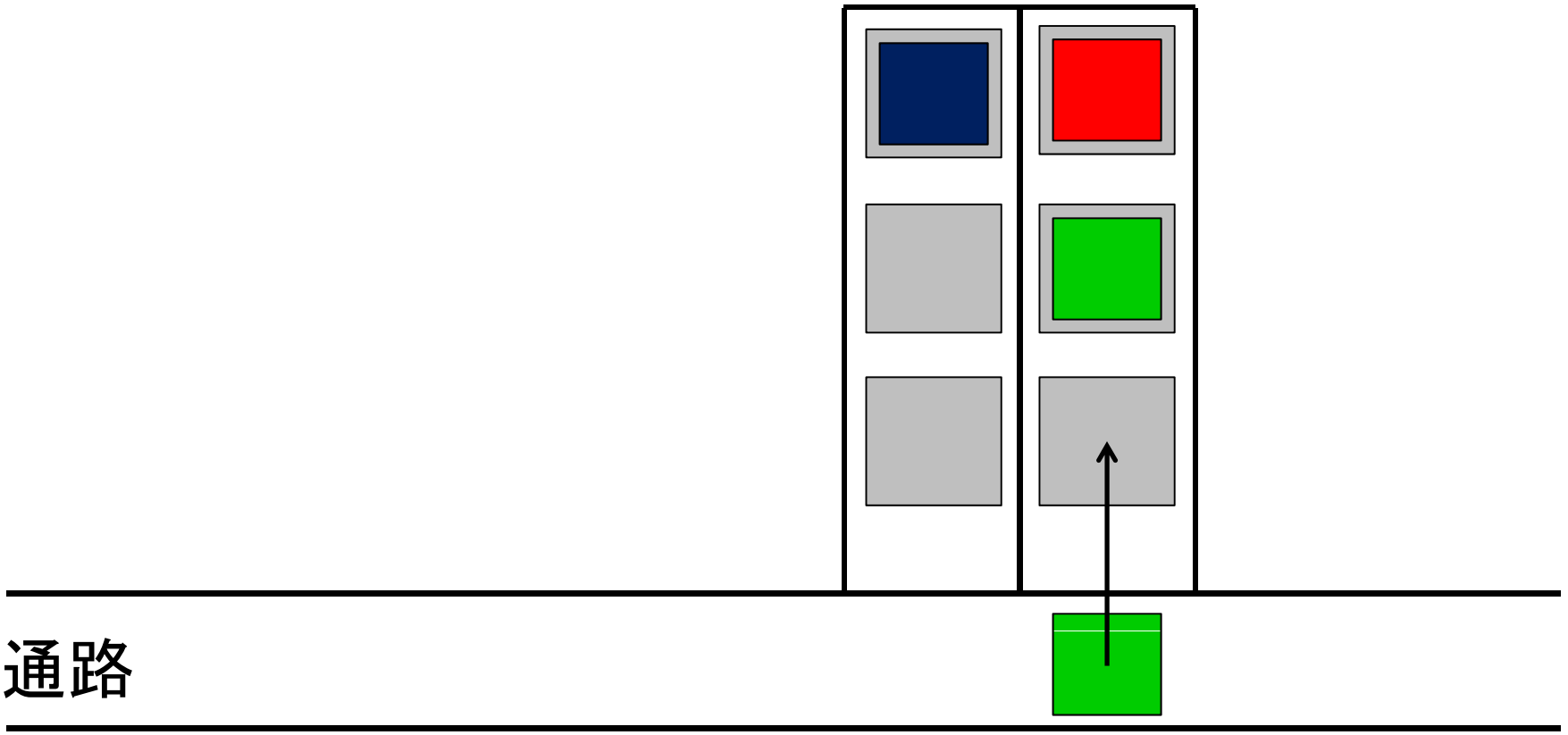
背景

- 先入れ後出し(First In Last Out)の**スタック構造**



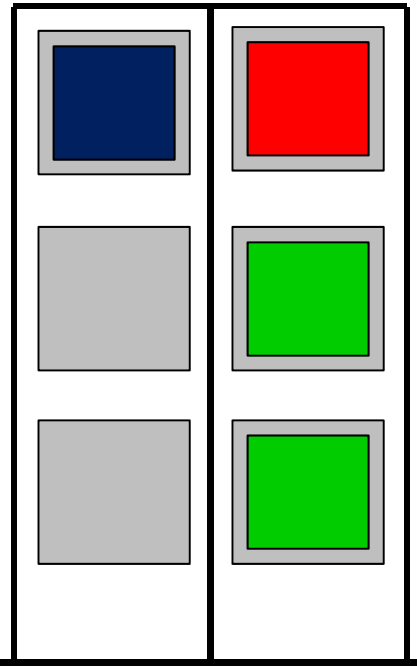
背景

- 先入れ後出し(First In Last Out)の**スタック構造**



背景

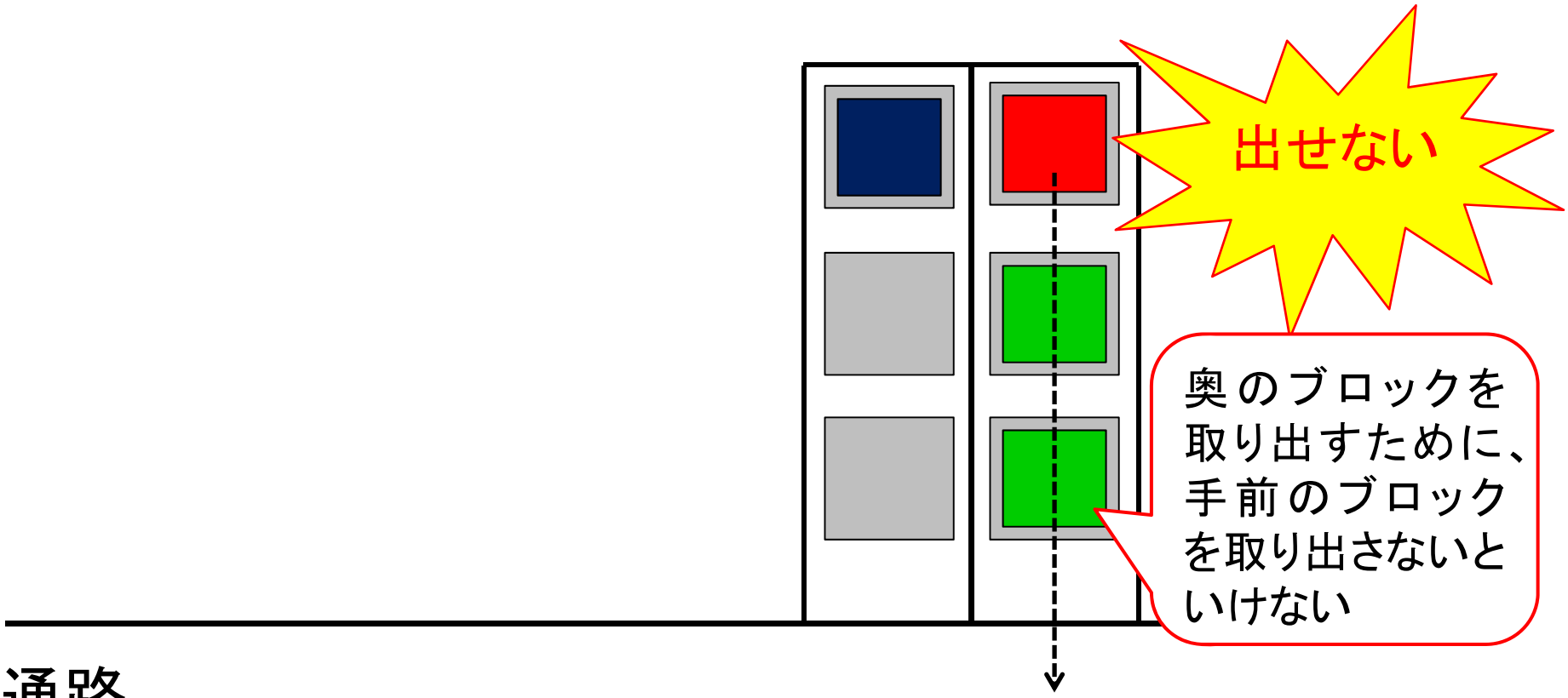
- 先入れ後出し(First In Last Out)の**スタック構造**



通路

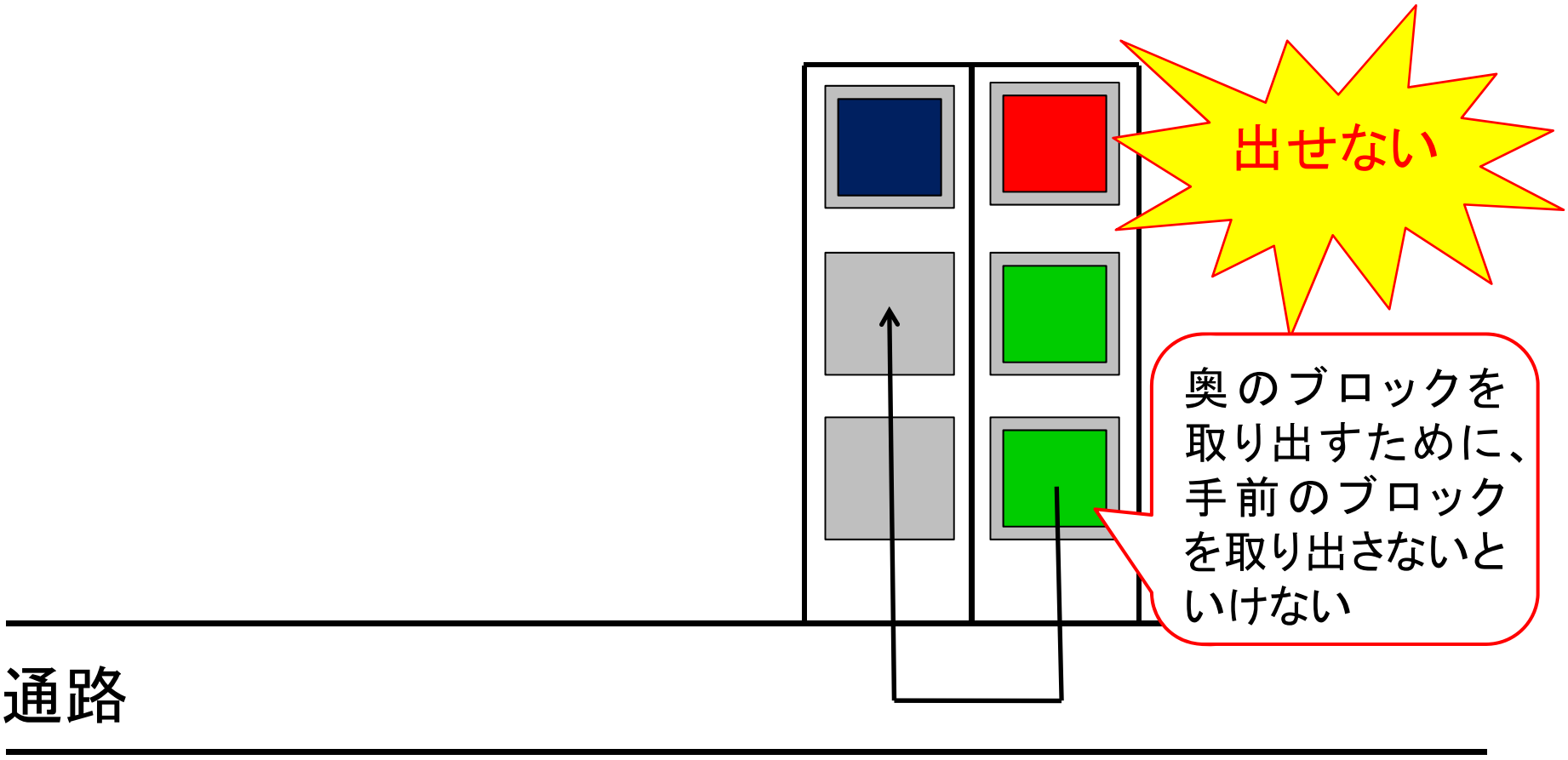
背景

- 先入れ後出し(First In Last Out)の**スタック構造**



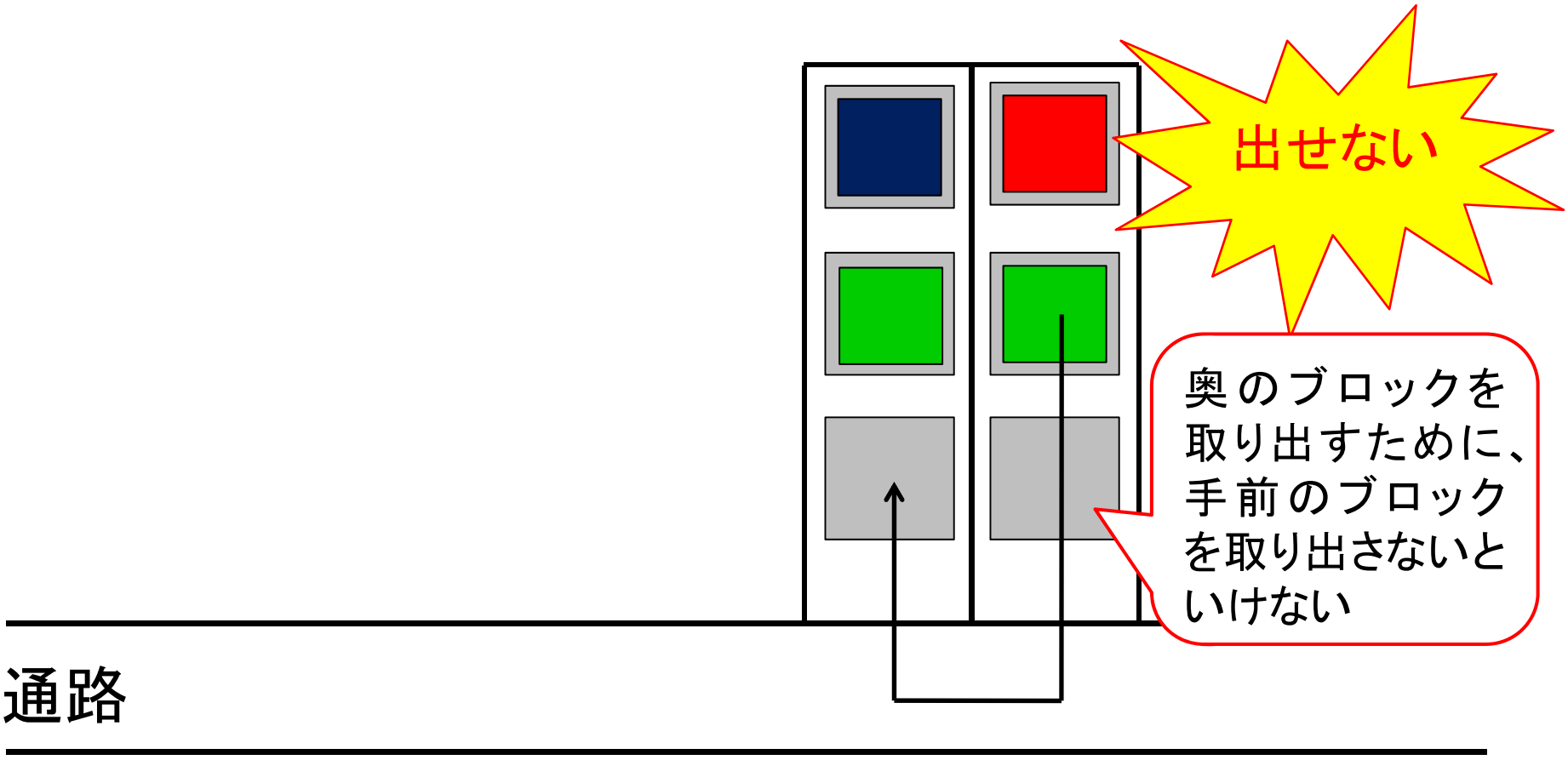
背景

- 先入れ後出し(First In Last Out)の**スタック構造**



背景

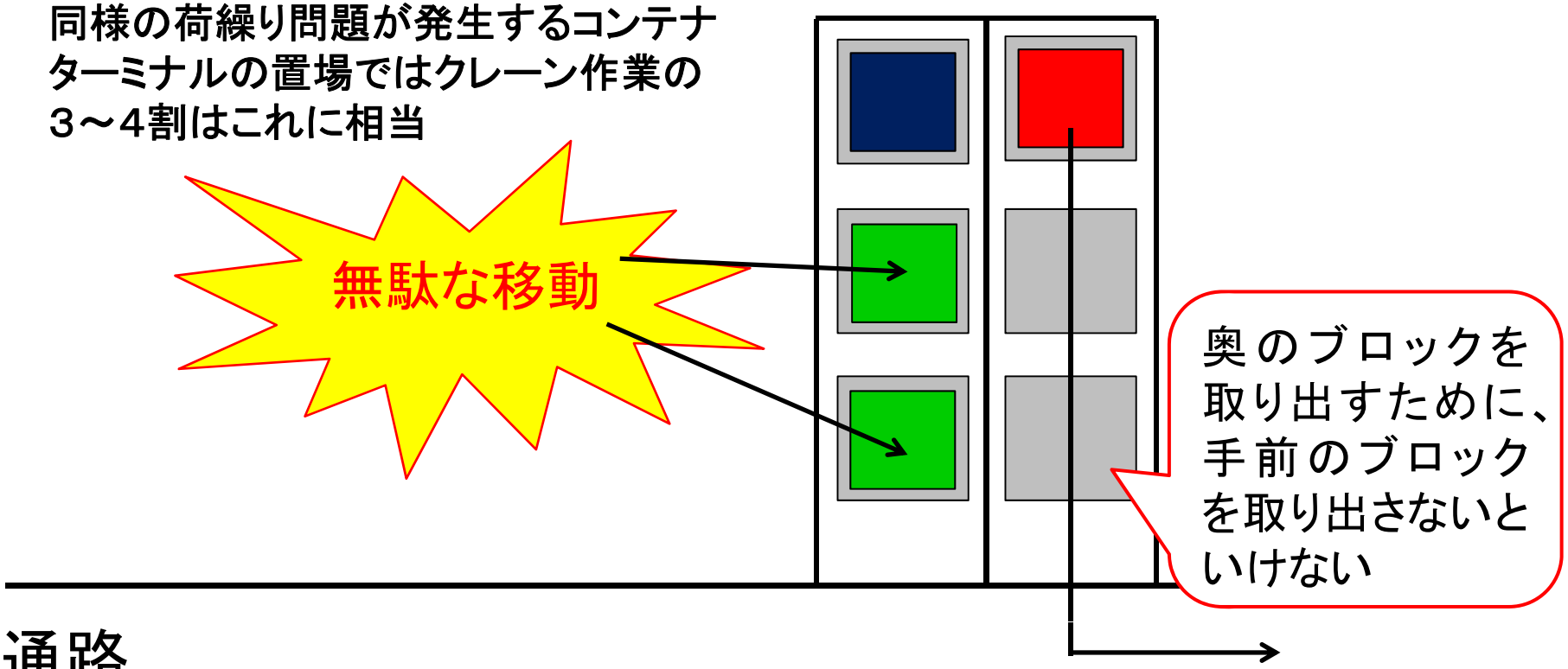
- 先入れ後出し(First In Last Out)の**スタック構造**



背景

- 先入れ後出し(First In Last Out)の**スタック構造**

同様の荷繰り問題が発生するコンテナターミナルの置場ではクレーン作業の3～4割はこれに相当



各ブロックはストックヤードへの搬入日と搬出日が決まっている

置場の指示を適切に行えば無駄な移動を減らせる

造船所の現状

- このスタック構造のために、ブロックがストックヤード内で無駄な移動を繰り返している
- 作業効率の低下につながる
- 今後生産量を増やしブロック数が増加すると、この無駄な移動が更に増える

↓

早急な改善が必須

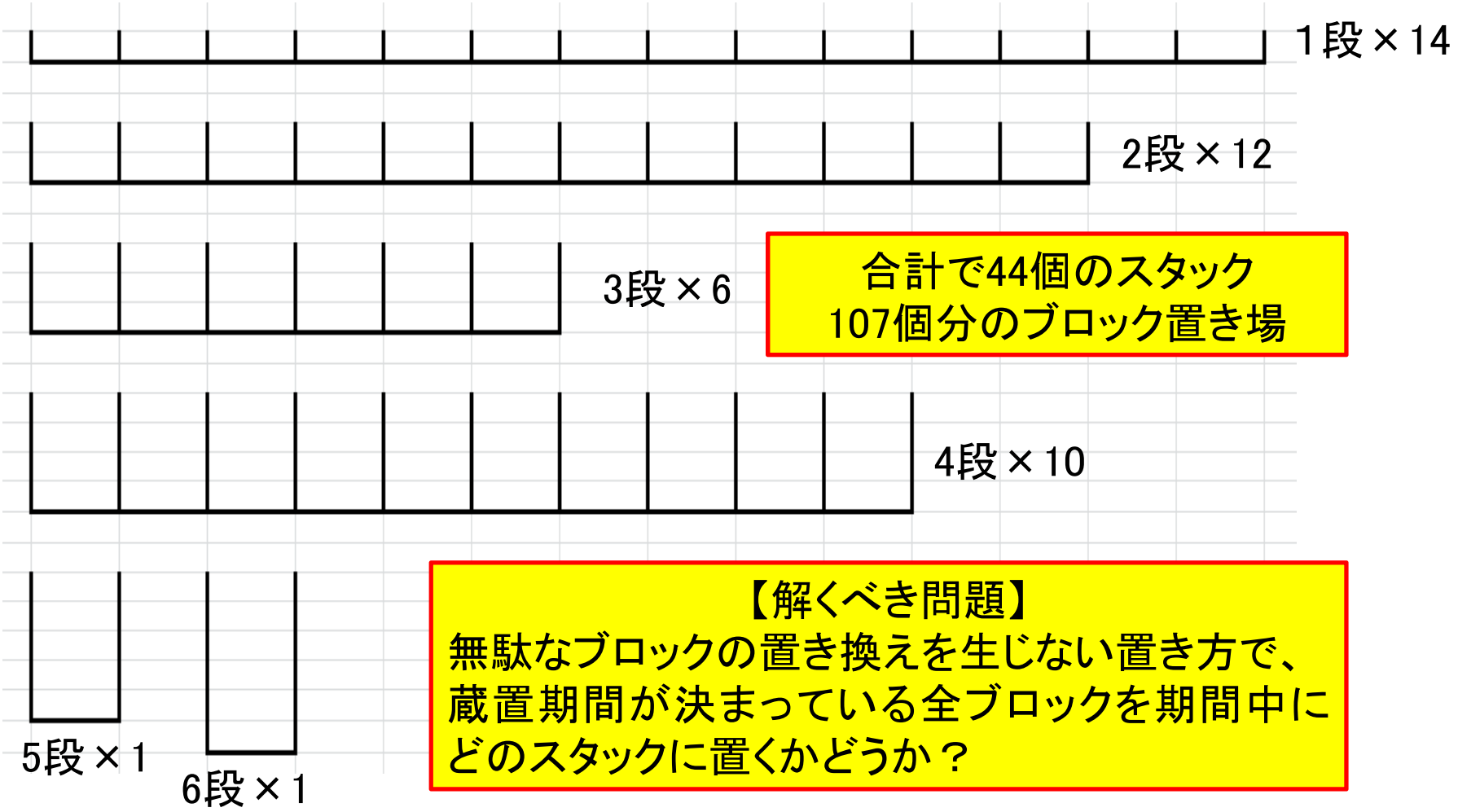
本研究の目的

①ブロック蔵置ルールの設定
②ブロック蔵置スケジューラーの開発

本発表の概要

1. 背景と目的
   - 造船所のストックヤードの現状と問題
2. アプローチ
   - 造船所の置場情報とブロック蔵置ルールの設定
   - ブロック割り当てアルゴリズム
3. 結果と考察
   - 従来手法との比較
   - サンプルデータの生成と検討
4. 実用化への取り組み
   - リスケジューリングと属性情報の追加
5. まとめと今後の課題

置き場の構造



合計で44個のスタック
107個分のブロック置き場

【解くべき問題】
無駄なブロックの置き換えを生じない置き方で、
蔵置期間が決まっている全ブロックを期間中に
どのスタックに置くかどうか？

無駄な置き換えの生じない蔵置方法

従来研究

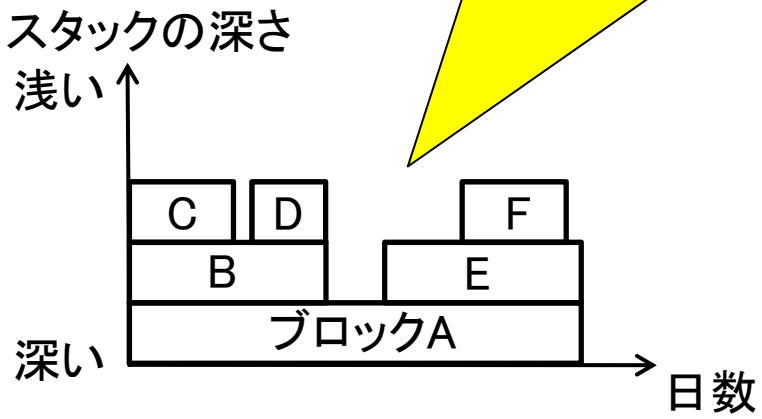
➢ 理想スタック蔵置ルール

ある時刻tにおいてスタックの深さdに置かれているブロックの蔵置を考える
(スタックの深さdは、1が最も深く、数字が増加するほど浅い置場とする)
蔵置開始時刻bs(t,d)
蔵置終了時刻be(t,d)
時刻tと深さdで指定された深さと時刻に何も置かれていない場合は
bs(t,d)＝be(t,d)＝t

ブロックの出し入れがFILOとなる条件：
全期間における任意のtにおいて、
bs(t,d1)≦bs(t,d2)かつbe(t,d1)≧be(t,d2)
(ただし深さd1≦d2)を満たすこと

理想スタック蔵置ルール

同一スタックにおいて、浅い場所に置かれるブロックの蔵置期間は、奥に置かれたブロックの範囲内にある

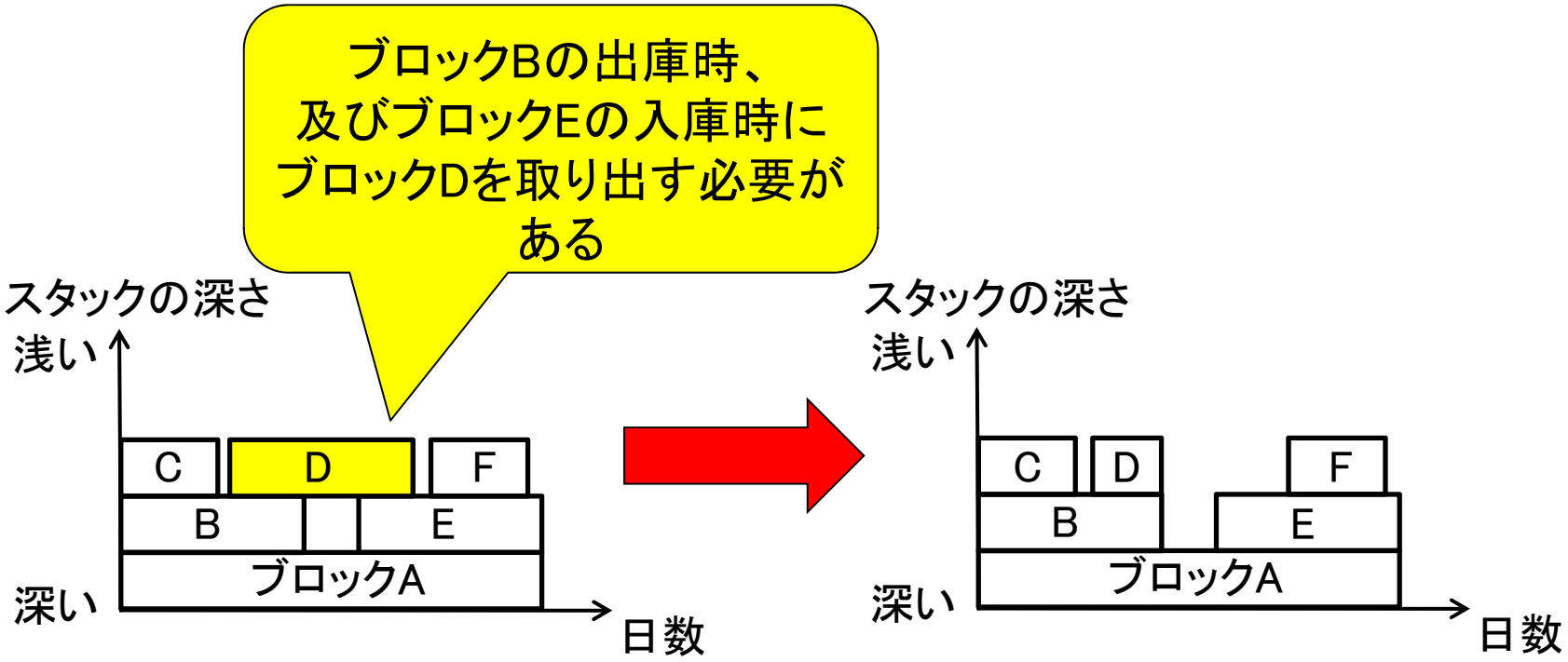


【理想スタック蔵置ルールに従う置き方】

無駄な置き換えの生じない蔵置方法

従来研究

➢ 理想スタック蔵置ルール



【理想スタック蔵置ルールに従わない置き方】　【理想スタック蔵置ルールに従う置き方】

ブロック割り当てアルゴリズム

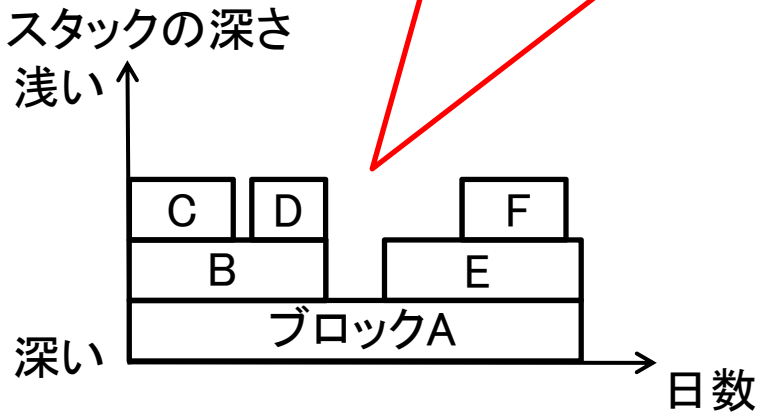
従来研究

【1】蔵置開始日と期間によるブロックデータのソート

ブロックNo.	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85
1000	■	■	■	■													
1001	■	■	■	■													
1002	■	■	■														
1003	■	■	■														
1004	■	■	■														
1005	■	■	■														
1006	■	■	■														
1007		■	■	■	■	■	■	■	■	■							
1008		■	■	■	■												
1009		■	■	■													
1010		■	■	■													
1011		■	■	■													
1012		■	■	■													
1013		■	■	■													
1014		■	■														
1015		■	■														
1016			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
1017			■	■	■	■	■	■	■	■	■	■	■	■	■	■	
1018			■	■	■	■											
1019			■	■	■	■											
1020			■	■	■												
1021			■	■	■												
1022			■	■	■												
1023			■	■	■												
1024			■	■	■												
1025			■	■													
1026				■	■	■	■	■	■	■	■	■	■				
1027				■	■	■	■										
1028				■	■	■	■										
1029				■	■	■	■										
1030				■	■	■											
1031				■	■	■											
1032				■	■	■											
1033				■	■	■											
1034					■	■	■	■	■	■	■	■	■				
1035					■	■	■	■	■	■	■	■					
1036					■	■	■	■	■	■	■	■					
1037					■	■	■										
1038					■	■	■										
1039					■	■	■										
1040					■	■	■										
1041					■	■	■										
1042					■	■											
1043					■	■											

ブロックを蔵置日の早いものかつ、蔵置期間の長いもの順にソート

蔵置開始日が同じブロックは、蔵置期間が長いものが先にスタックの奥へ置かれるべき



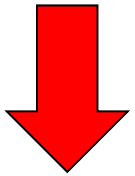
【理想スタック蔵置ルールに従う置き方】

ブロック割り当てアルゴリズム

従来研究

【2】ソートされたブロックの割り当てアルゴリズム

段数の多いスタックほど扱いが厄介



段数の多いスタックから隙間なくブロックを詰め込む

(最適解を得る保証がないヒューリスティクス)

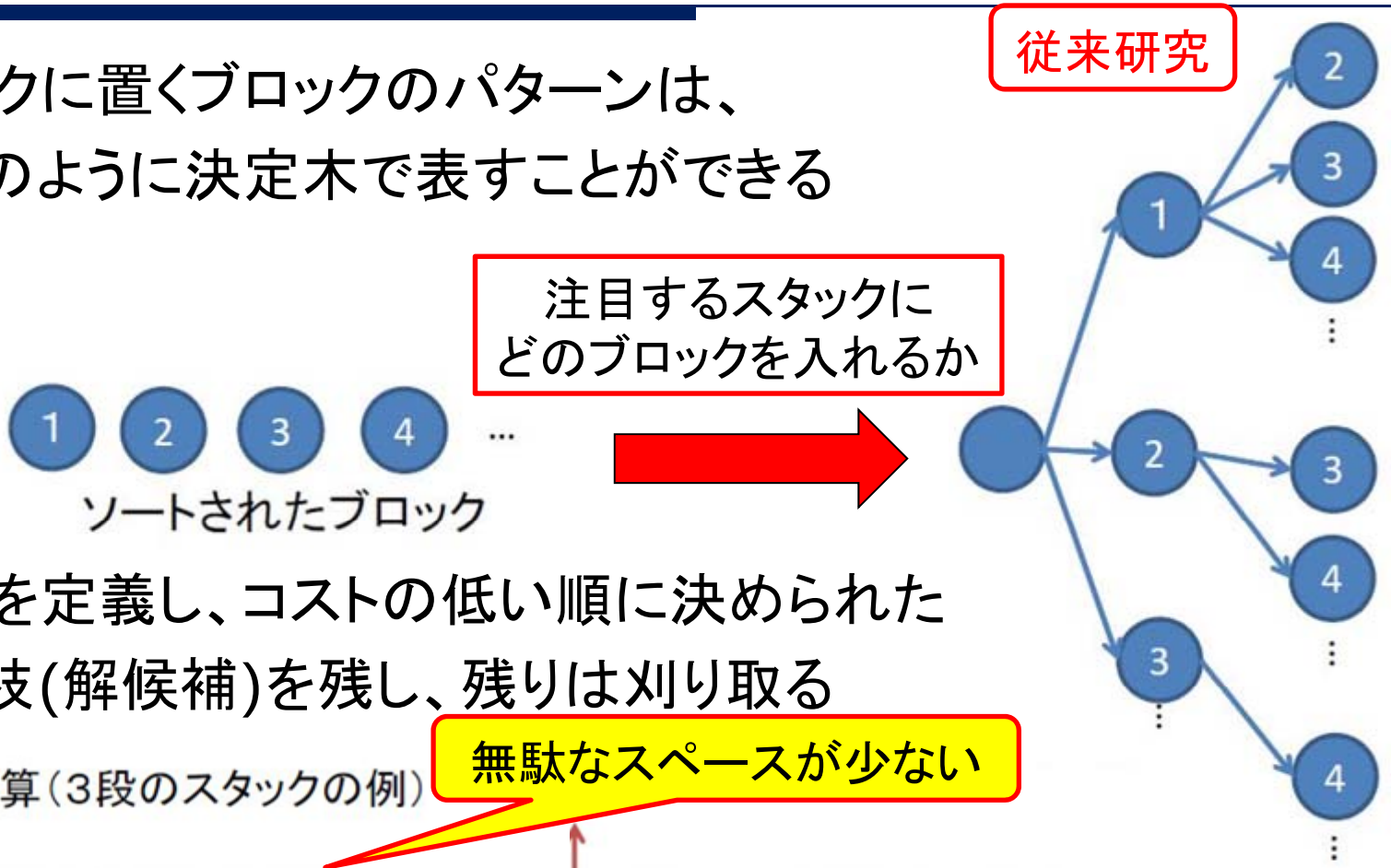
① 深いスタックから着目し、置場の決まっていないブロックを選択してソートした順に割り当てていく

② 理想スタック蔵置ルールを順守

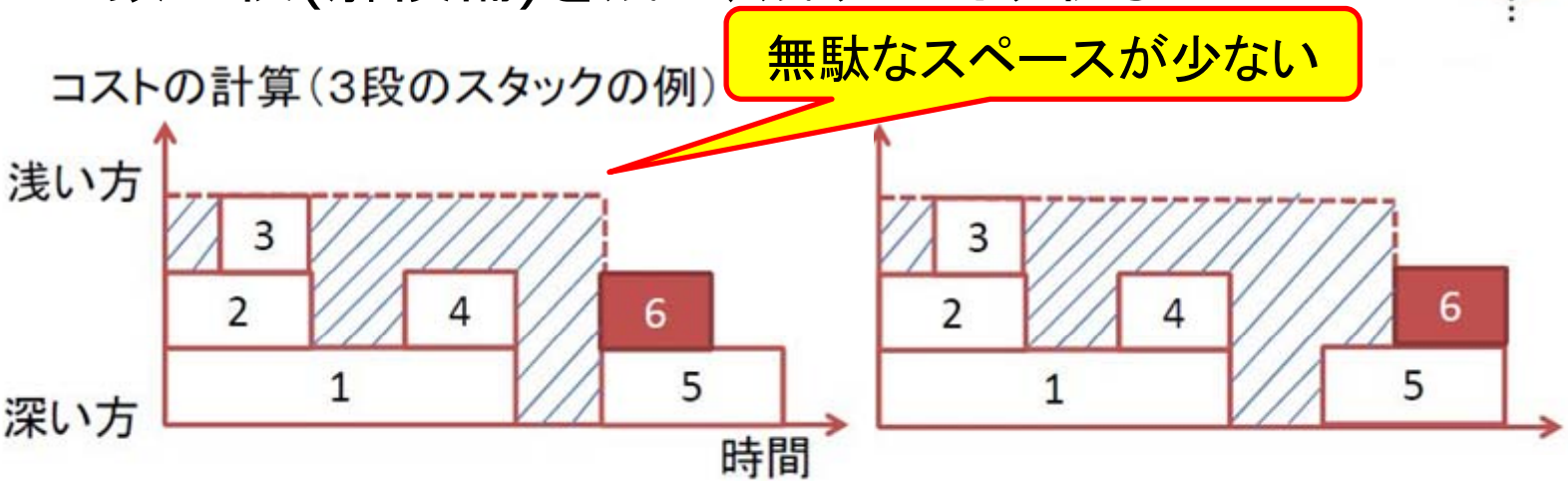
③ スタックの利用効率が高くなるようブロックを選択

深いスタックから割り当てる分枝限定法1

- スタックに置くブロックのパターンは、
  右図のように決定木で表すことができる

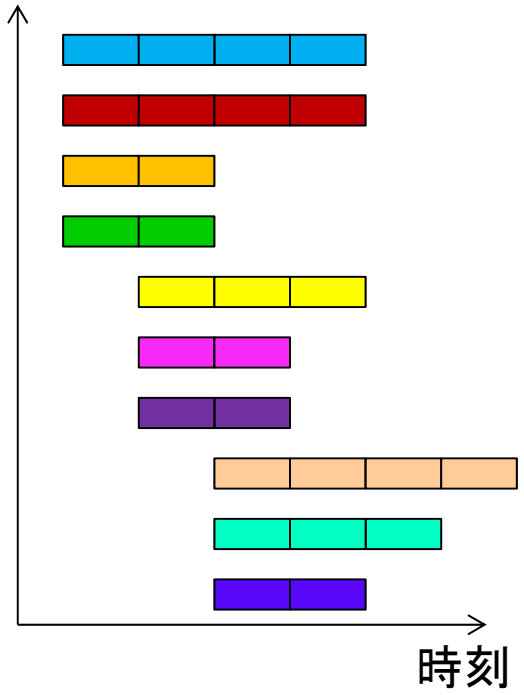


- コストを定義し、コストの低い順に決められた
  数の枝(解候補)を残し、残りは刈り取る

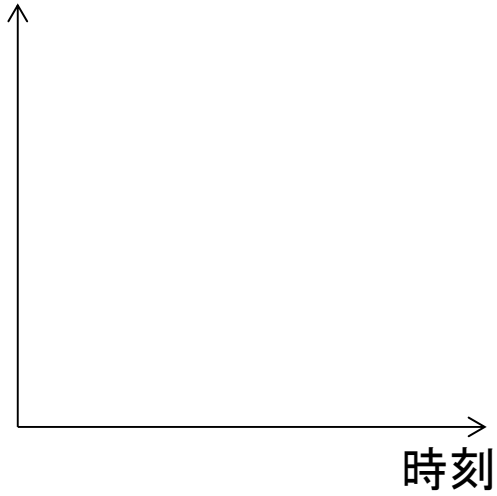


ブロック割り当てアルゴリズム

従来研究



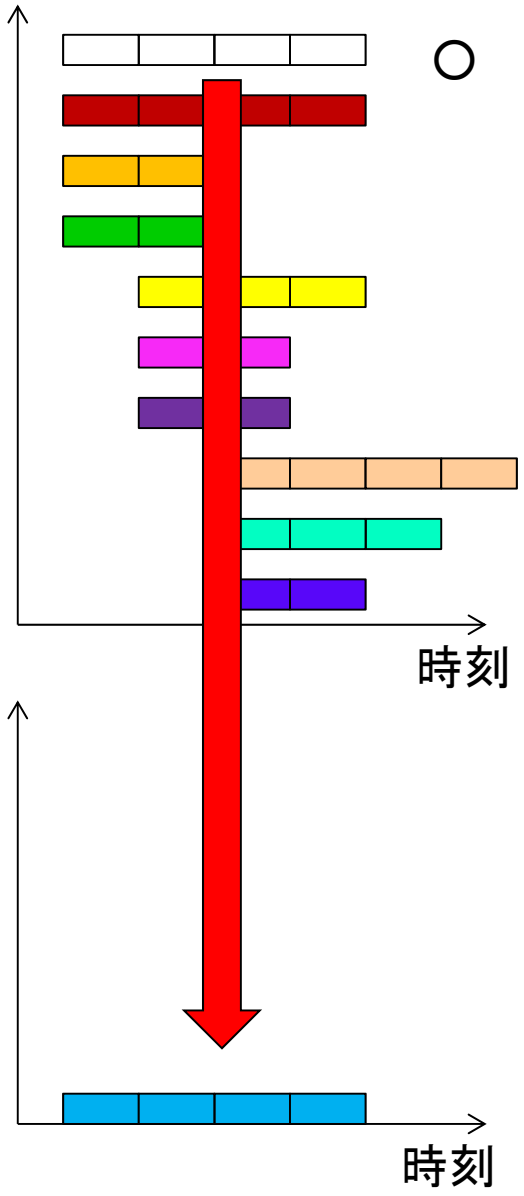
蔵置開始日と期間でソートされたブロック



3段のスタックへの割り当て例

ブロック割り当てアルゴリズム

従来研究

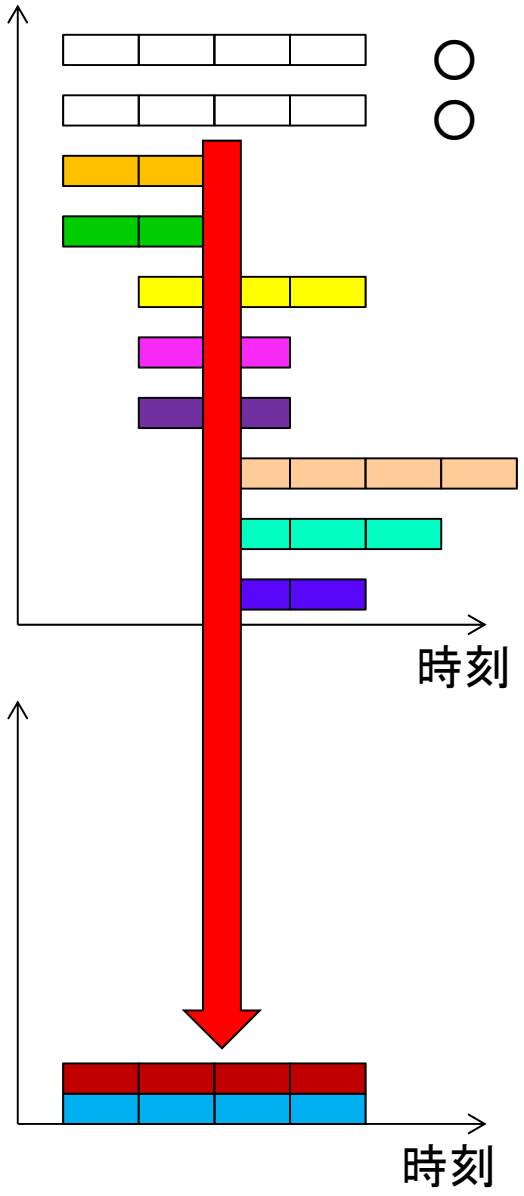


○ 蔵置開始日と期間でソートされたブロック

3段のスタックへの割り当て例

ブロック割り当てアルゴリズム

従来研究

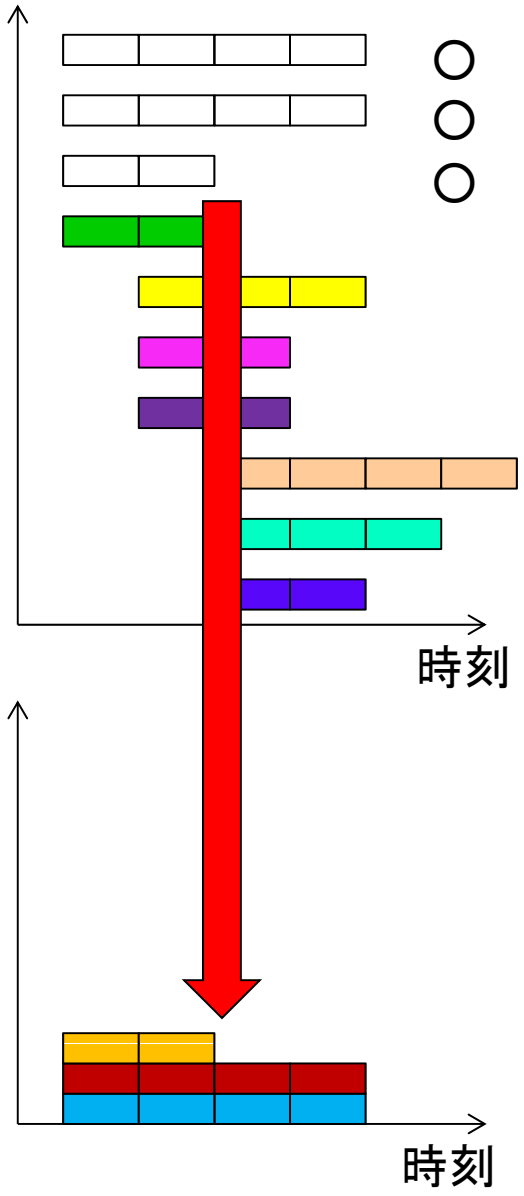


蔵置開始日と期間でソートされたブロック

3段のスタックへの割り当て例

ブロック割り当てアルゴリズム

従来研究

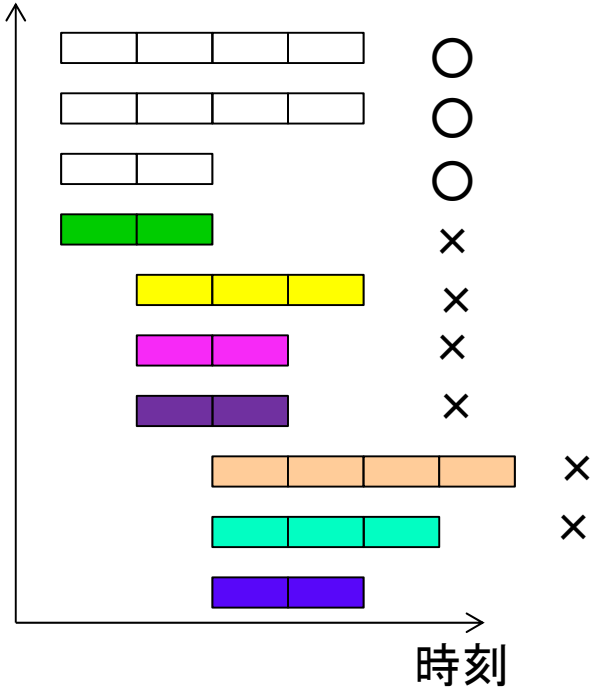


蔵置開始日と期間でソートされたブロック

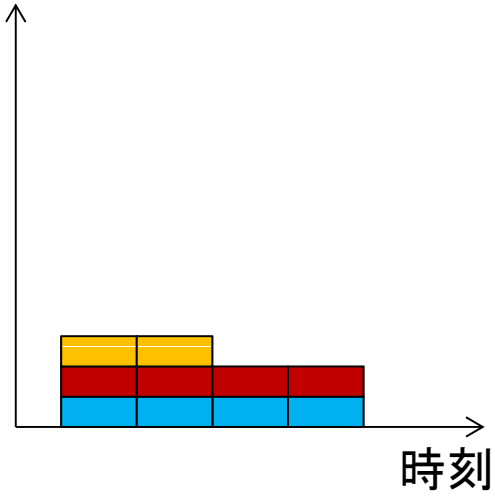
3段のスタックへの割り当て例

ブロック割り当てアルゴリズム

従来研究



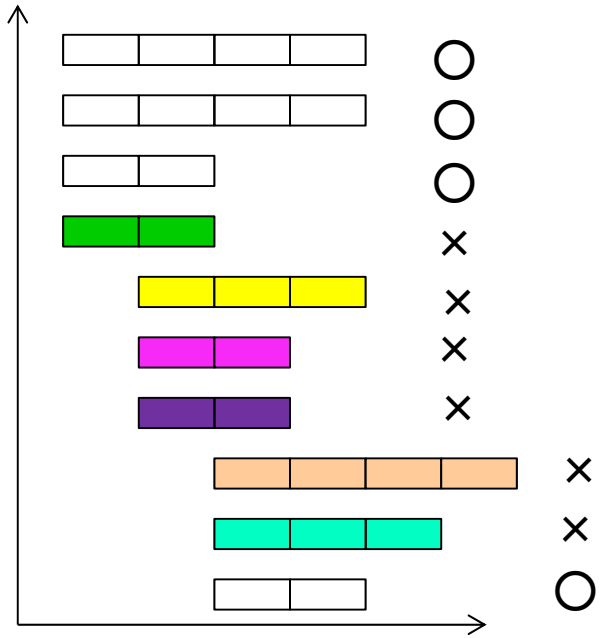
蔵置開始日と期間でソートされたブロック



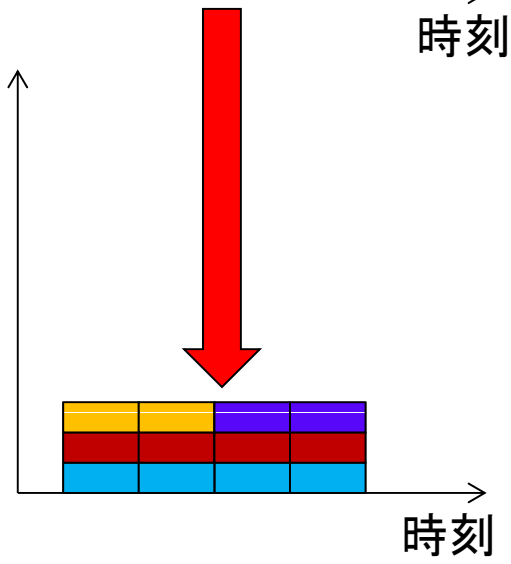
3段のスタックへの割り当て例

ブロック割り当てアルゴリズム

従来研究



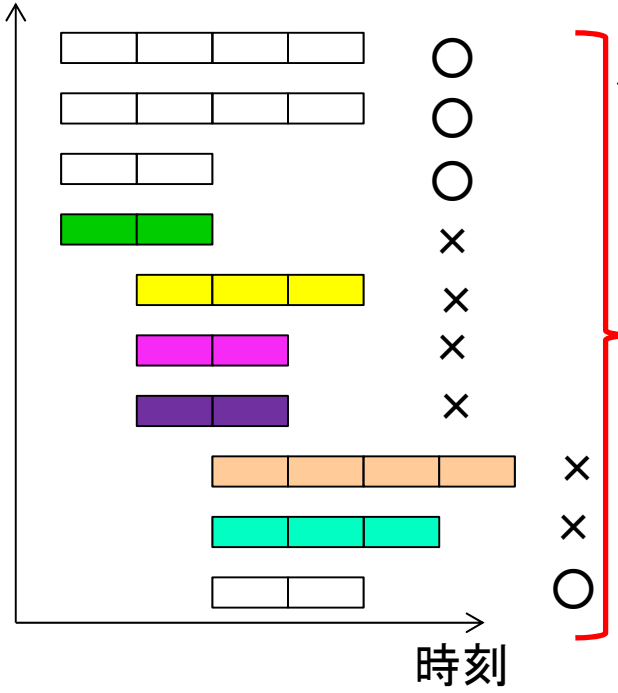
蔵置開始日と期間でソートされたブロック



3段のスタックへの割り当て例

ブロック割り当てアルゴリズム

従来研究



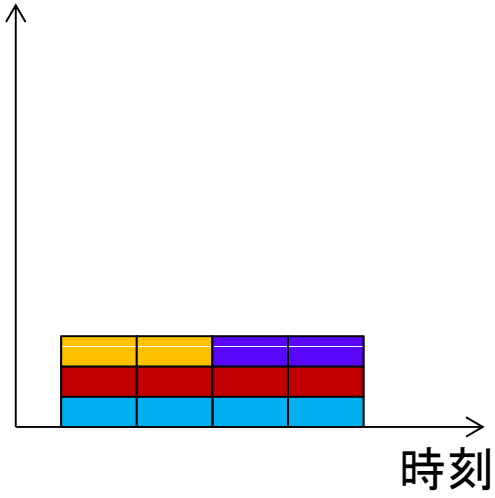
蔵置開始日と期間でソートされたブロック

問題の定式化

注目するスタックに
未蔵置ブロックを割り当てるかどうか

2の(ブロックの個数)乗の
組合せ最適化　2^{3079}

分枝限定法で解く

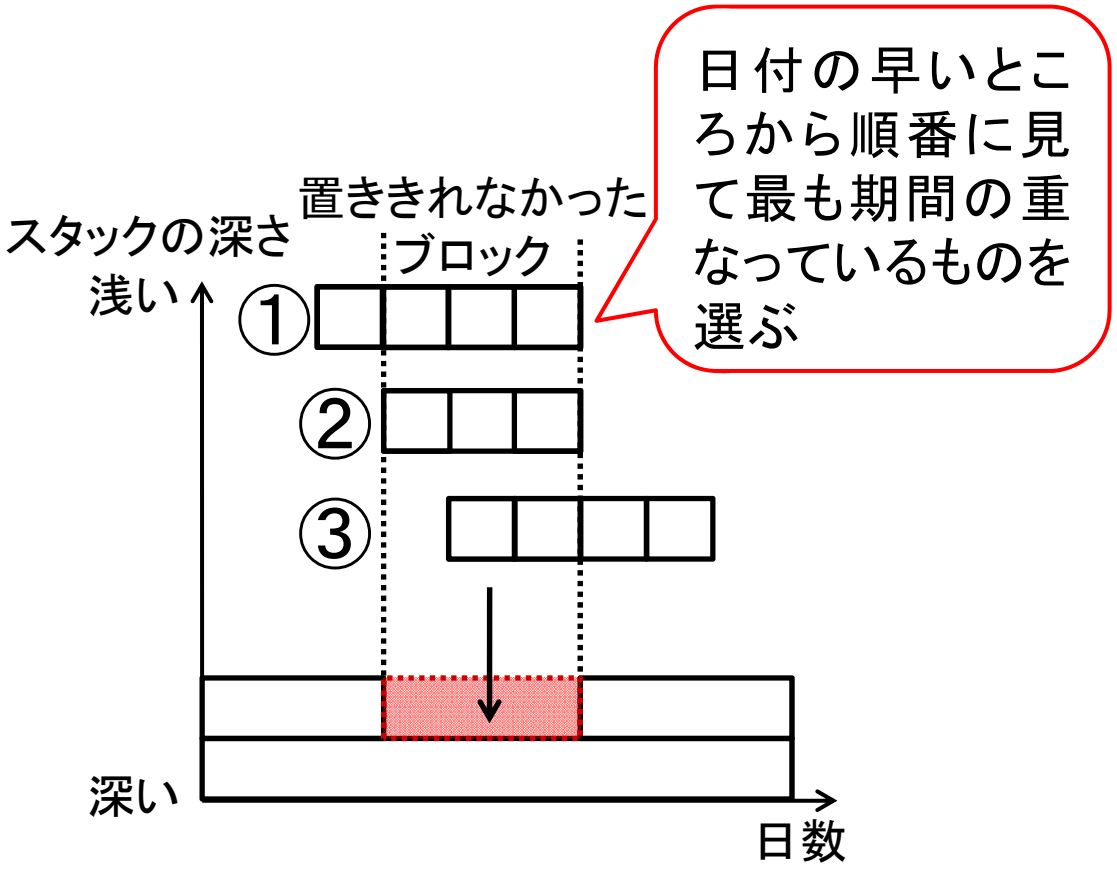


3段のスタックへの割り当て例

未蔵置ブロックの分割配置

従来研究

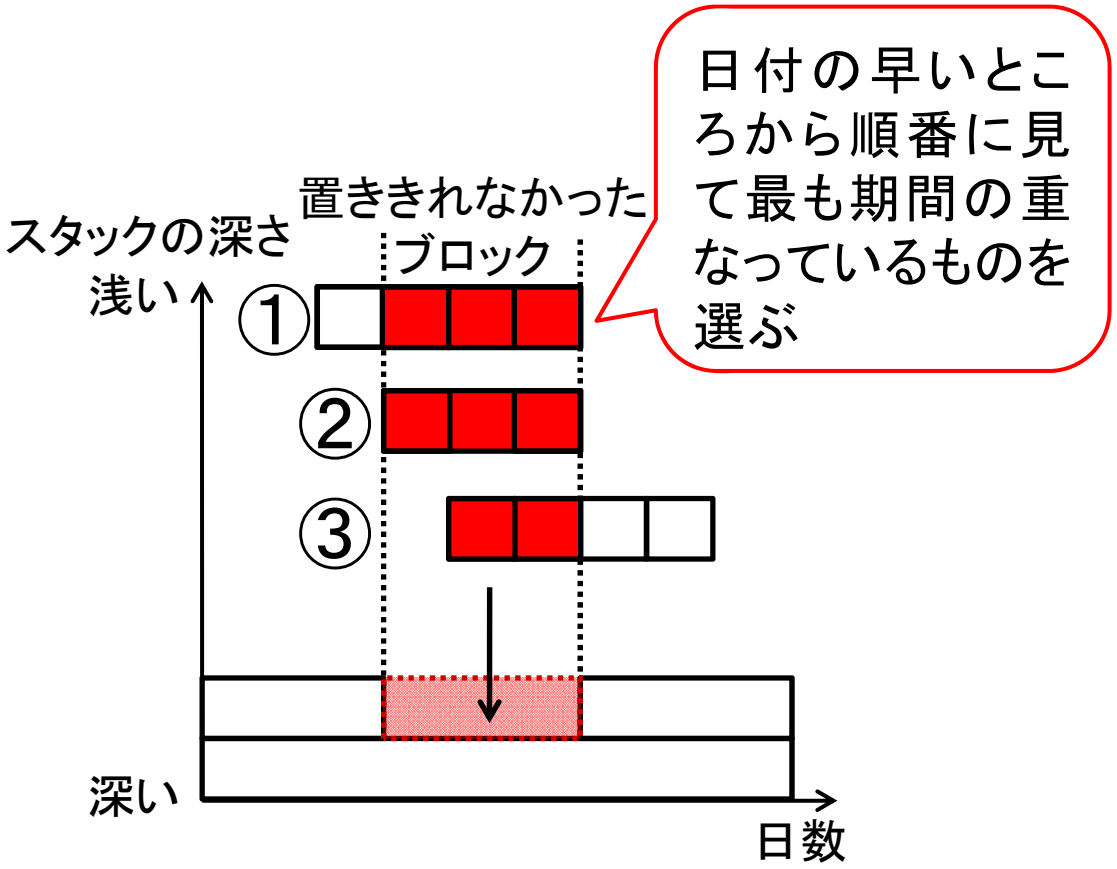
- 分枝限定法で配置できなかったブロックの蔵置期間を分割して蔵置する
  - 赤枠の空いたスペースにどのブロックを埋めるか



未蔵置ブロックの分割配置

従来研究

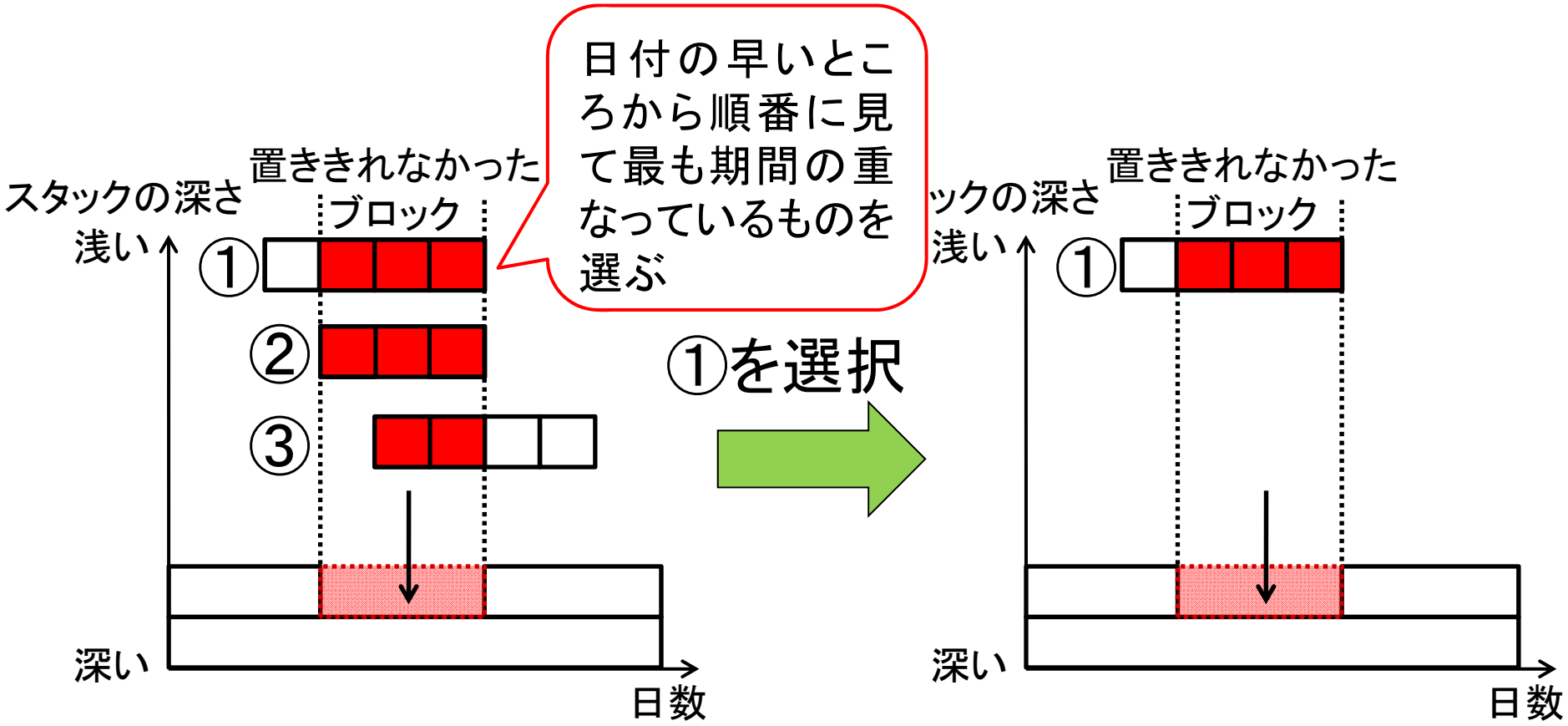
- 分枝限定法で配置できなかったブロックの蔵置期間を分割して蔵置する
  - 赤枠の空いたスペースにどのブロックを埋めるか



未蔵置ブロックの分割配置

従来研究

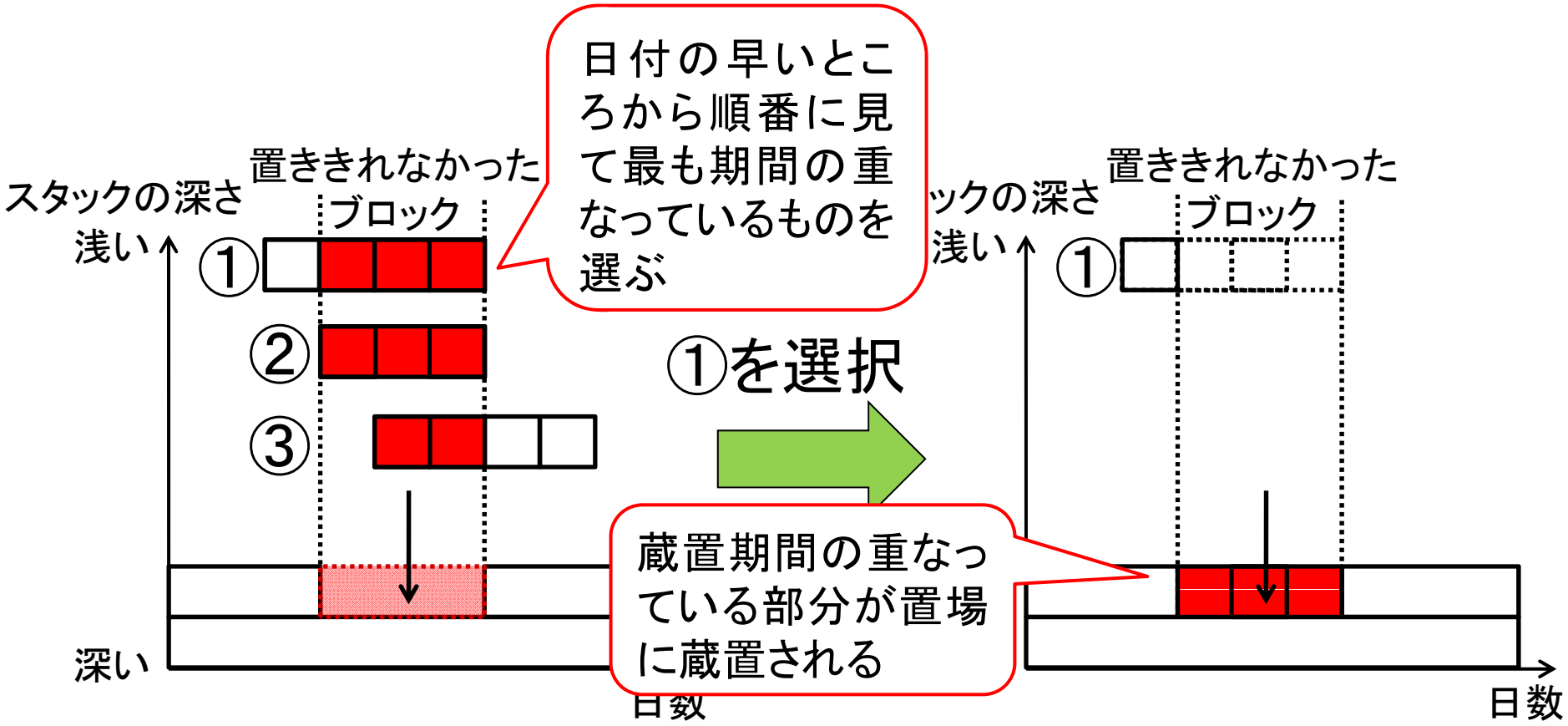
- 分枝限定法で配置できなかったブロックの蔵置期間を分割して蔵置する
  - 赤枠の空いたスペースにどのブロックを埋めるか



未蔵置ブロックの分割配置

従来研究

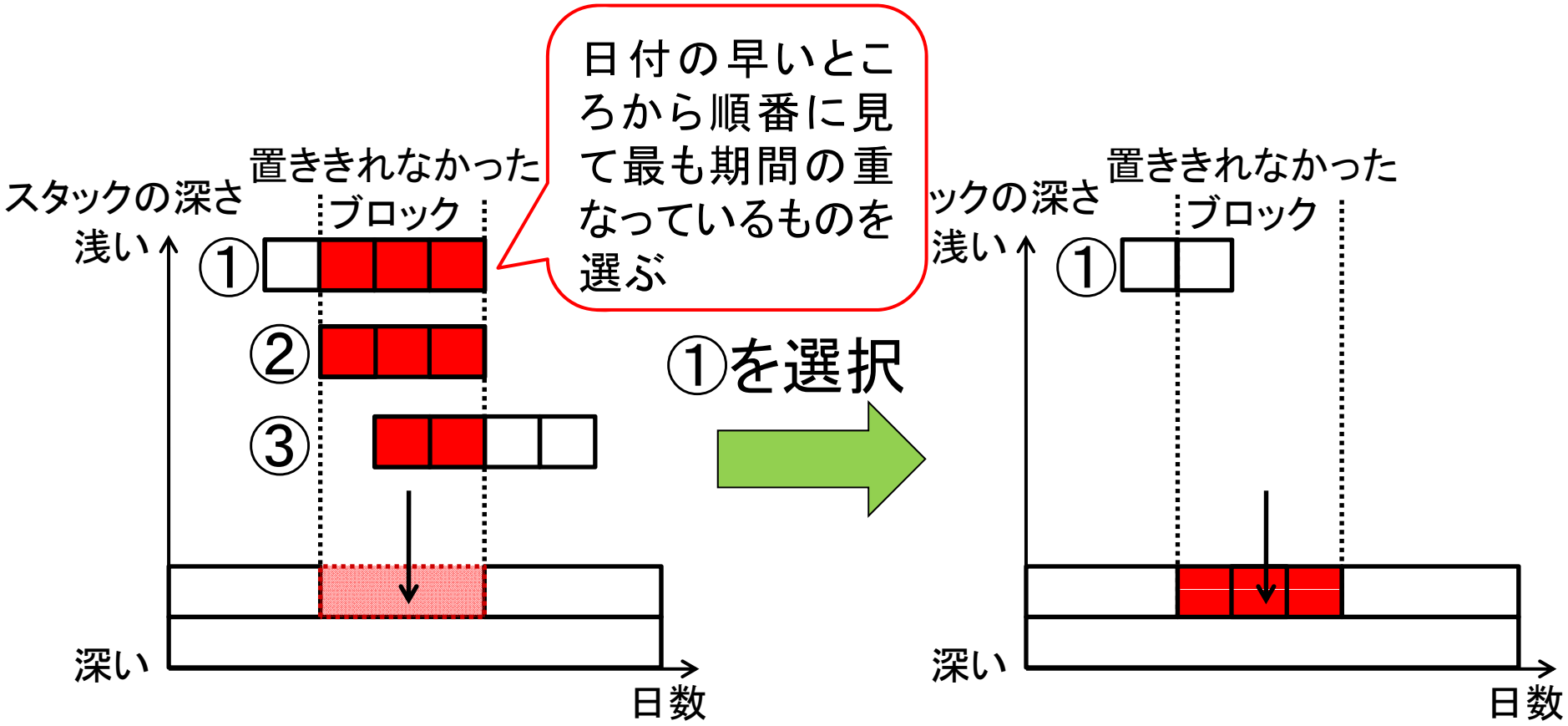
- 分枝限定法で配置できなかったブロックの蔵置期間を分割して蔵置する
  - 赤枠の空いたスペースにどのブロックを埋めるか



未蔵置ブロックの分割配置

従来研究

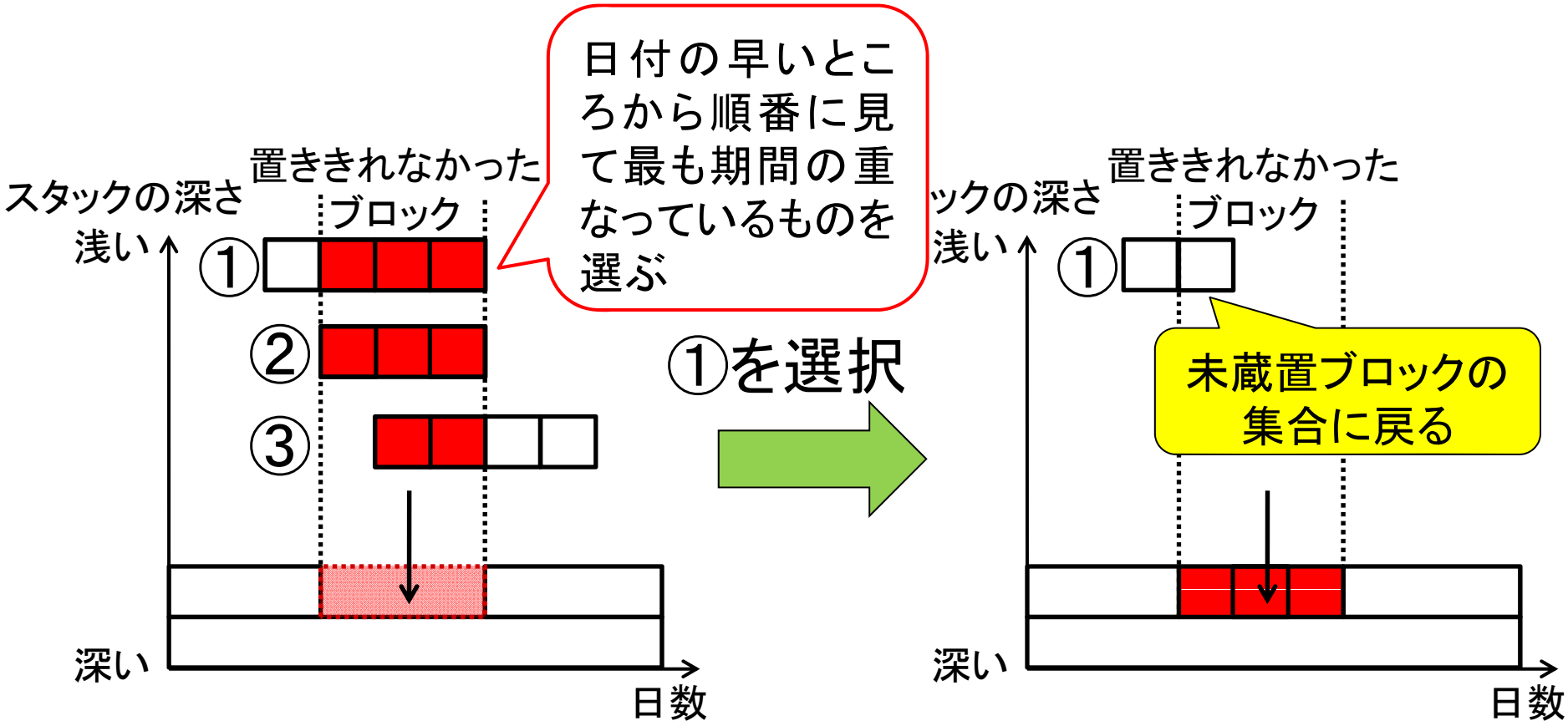
- 分枝限定法で配置できなかったブロックの蔵置期間を分割して蔵置する
  - 赤枠の空いたスペースにどのブロックを埋めるか



未蔵置ブロックの分割配置

従来研究

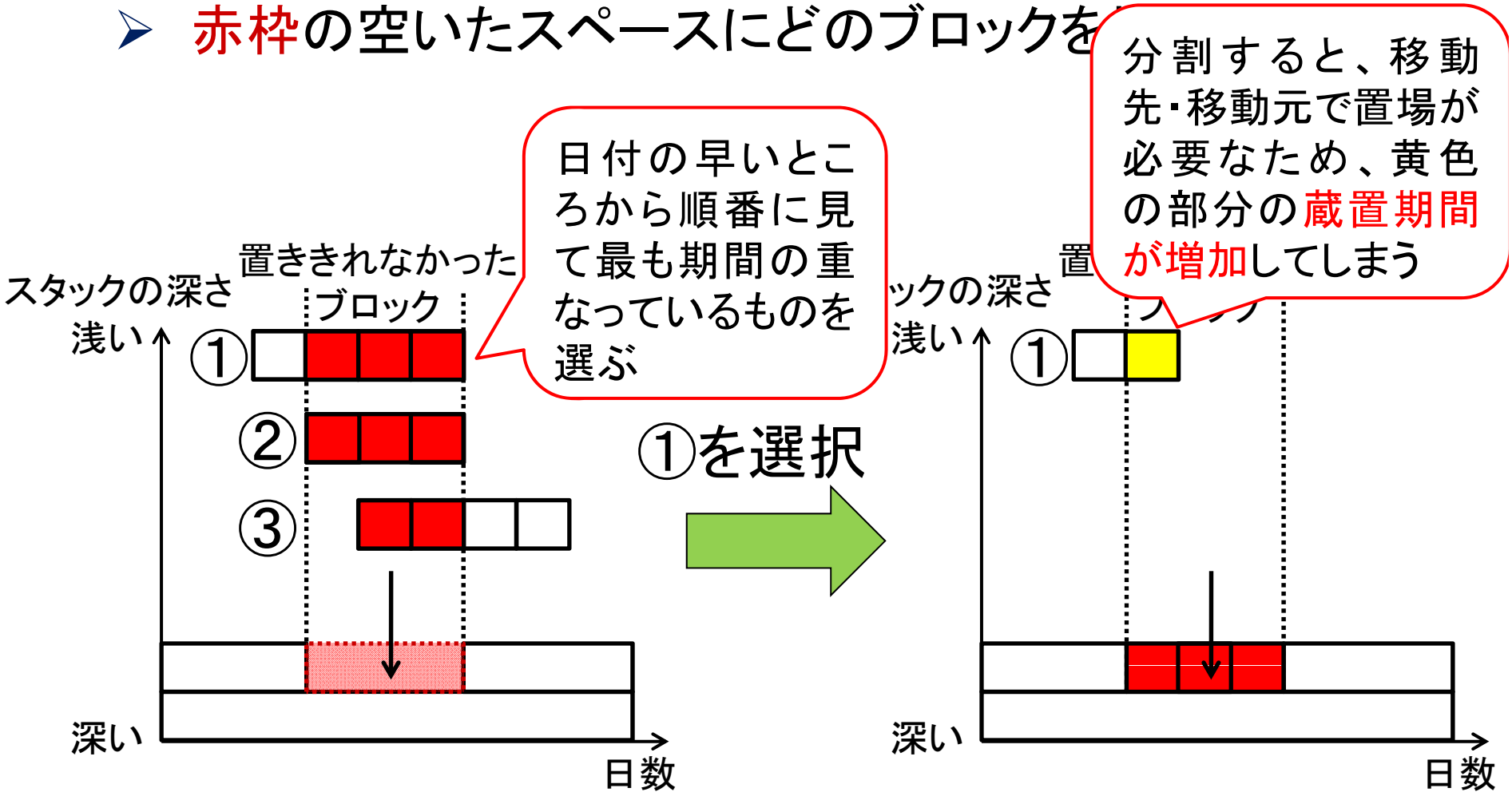
- 分枝限定法で配置できなかったブロックの蔵置期間を分割して蔵置する
  - 赤枠の空いたスペースにどのブロックを埋めるか



未蔵置ブロックの分割配置

従来研究

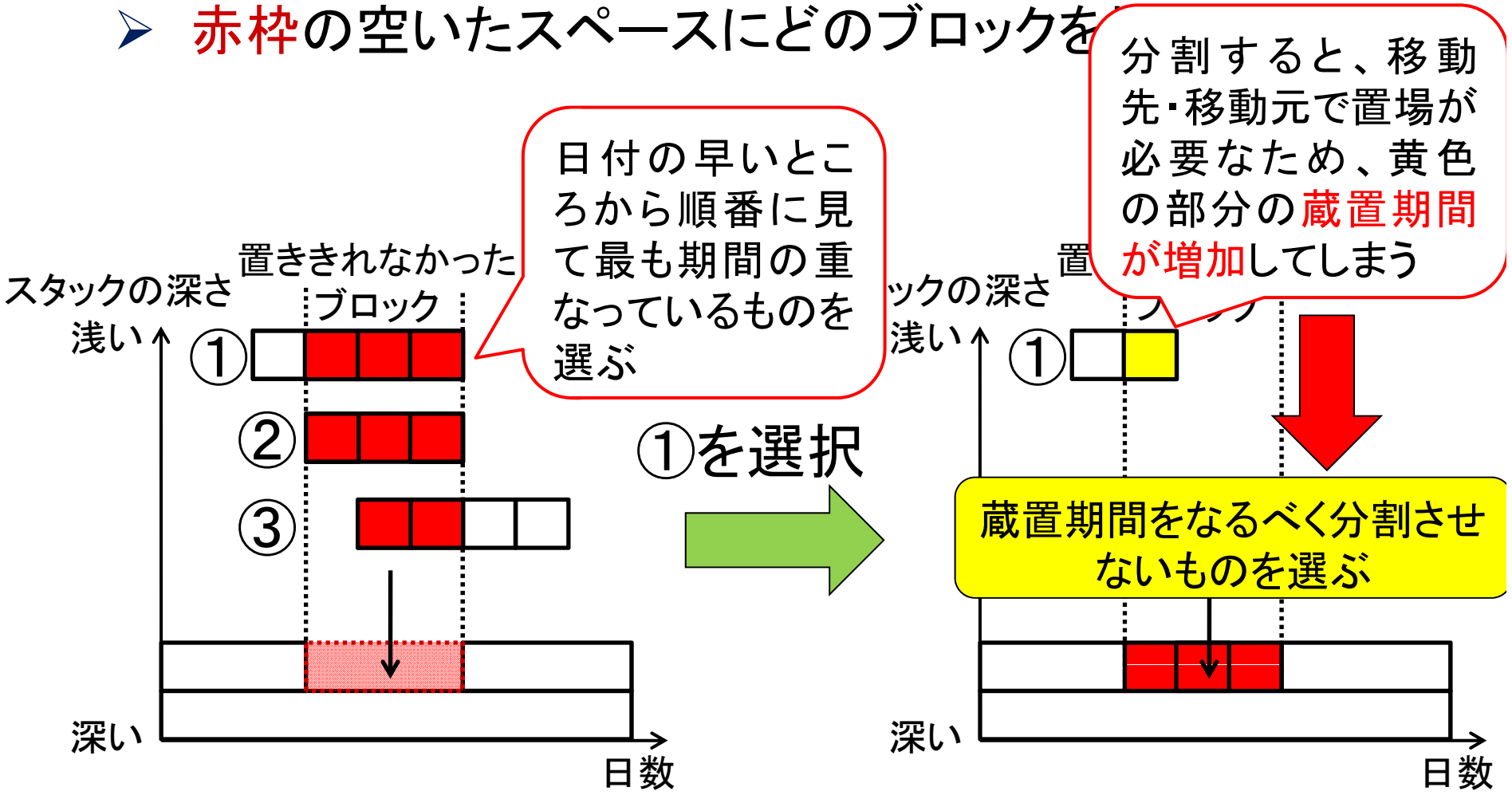
- 分枝限定法で配置できなかったブロックの蔵置期間を分割して蔵置する
  - 赤枠の空いたスペースにどのブロックを



未蔵置ブロックの分割配置

本研究

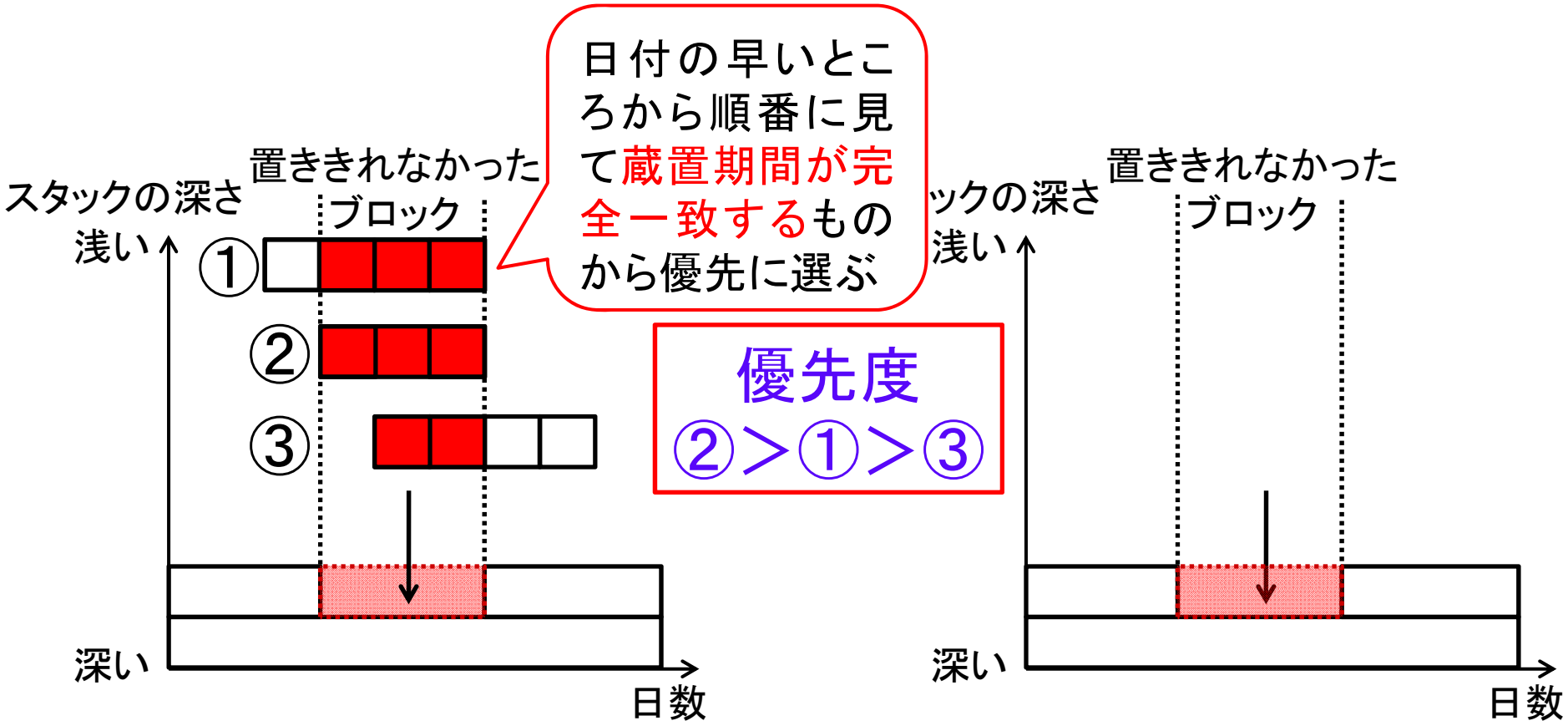
- 分枝限定法で配置できなかったブロックの蔵置期間を分割して蔵置する
  - 赤枠の空いたスペースにどのブロックを



未蔵置ブロックの分割配置

本研究

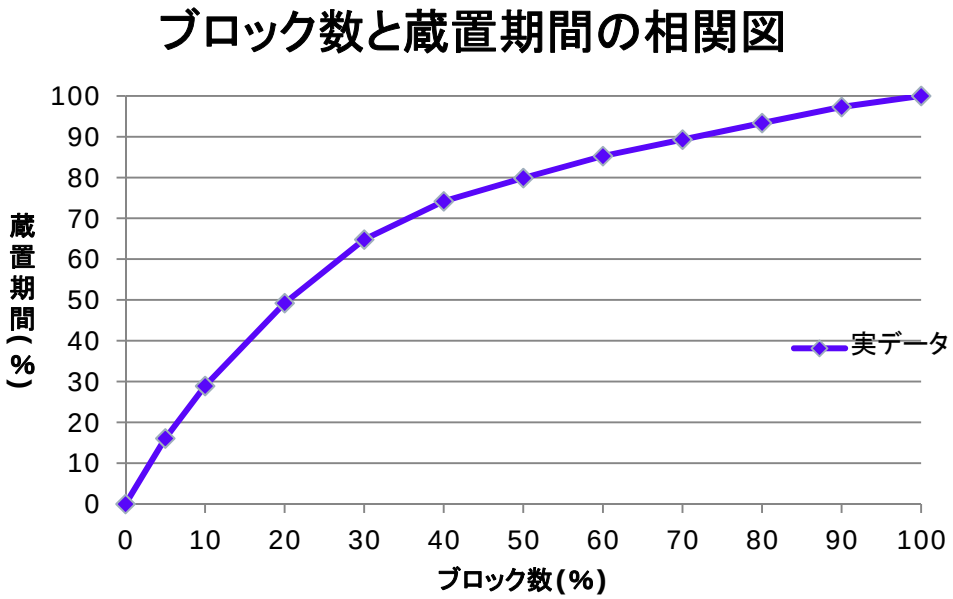
- 分枝限定法で配置できなかったブロックの蔵置期間を分割して蔵置する
  - 赤枠の空いたスペースにどのブロックを埋めるか



スケジュール方法の改善の検討

本研究

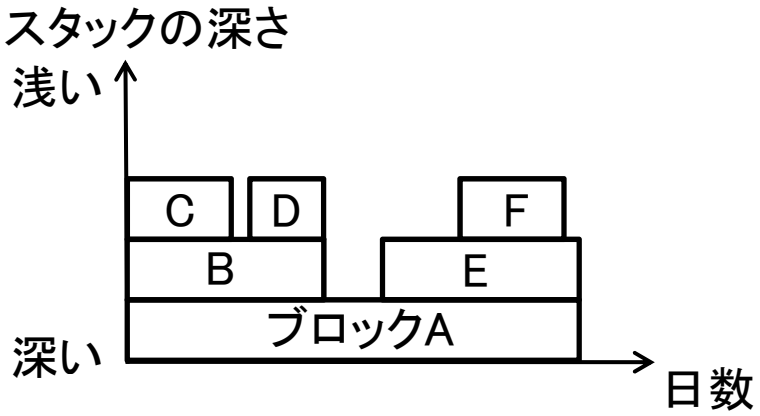
- 蔵置期間の長い順にブロックをソートし、ブロック数と全体の蔵置期間に占める蔵置期間の割合について調べた
- ブロック数20%で全体の蔵置期間の約50%を占めた
- 蔵置期間の長いブロックがスケジュール結果に影響を及ぼす



スケジュール方法の改善の検討

本研究

- このストックヤードの特徴であるスタック構造より蔵置期間の長いブロックがスタックの深い所に配置される必要がある
- 蔵置期間を分割して配置するとのべ蔵置期間が増加してしまう
- 蔵置期間の長いブロックがスケジュール結果に影響を及ぼす



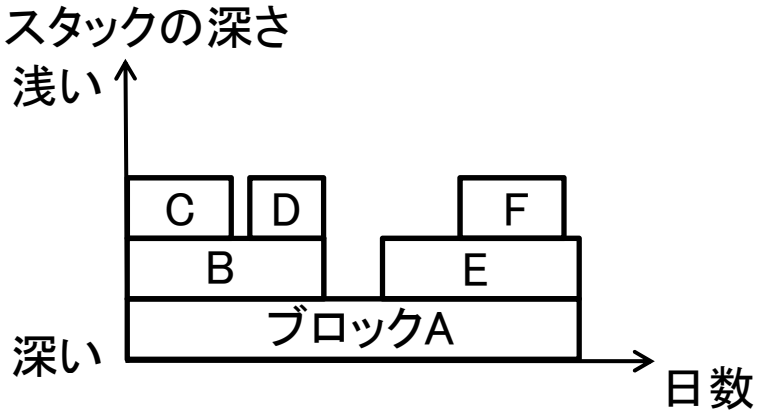
【理想スタック蔵置ルールに従う置き方】

スケジュール方法の改善の検討

本研究

- このストックヤードの特徴であるスタック構造より蔵置期間の長いブロックがスタックの深い所に配置される必要がある
- 蔵置期間を分割して配置するとのべ蔵置期間が増加してしまう
- 蔵置期間の長いブロックがスケジュール結果に影響を及ぼす

スケジュールの性能を向上させるには



【理想スタック蔵置ルールに従う置き方】

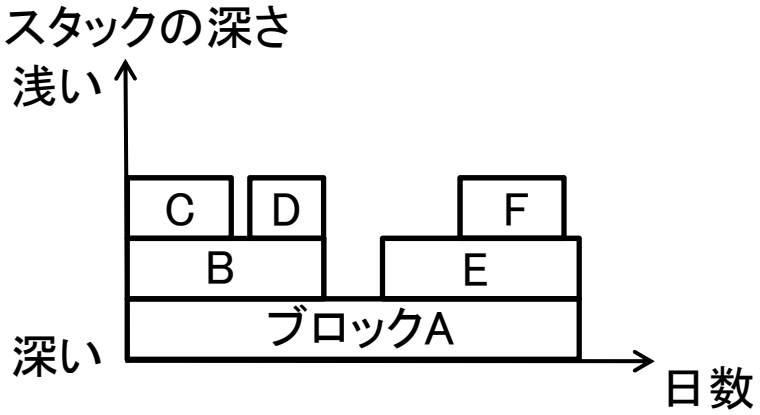
スケジュール方法の改善の検討

本研究

- このストックヤードの特徴であるスタック構造より蔵置期間の長いブロックがスタックの深い所に配置される必要がある
- 蔵置期間を分割して配置するとのべ蔵置期間が増加してしまう
- 蔵置期間の長いブロックがスケジュール結果に影響を及ぼす

スケジュールの性能を向上させるには

蔵置期間の長いブロック
を優先的にスケジュール

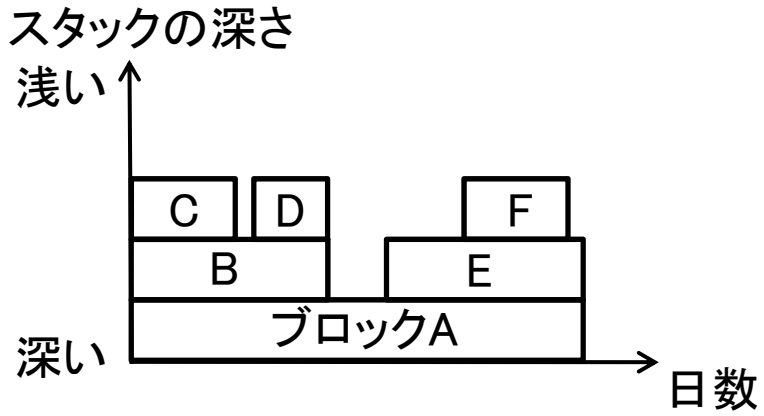


【理想スタック蔵置ルールに従う置き方】

蔵置期間の長いブロックを初期配置

本研究

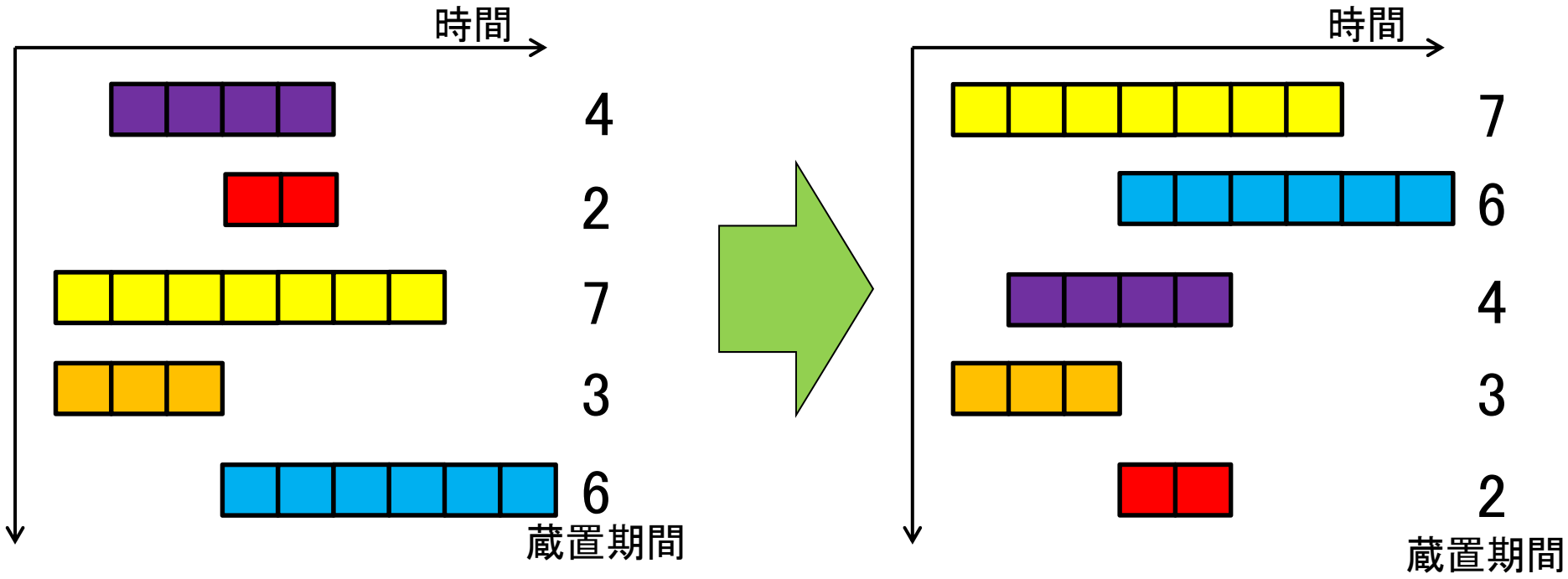
- 蔵置期間の長いブロックの任意の個数を分枝限定法でスケジュールする前に置き場に配置する
- この時、最も深いスタック(6段)から詰めていき、置き切れない場合、次に深いスタック(5段)へと移行する



【理想スタック蔵置ルールに従う置き方】

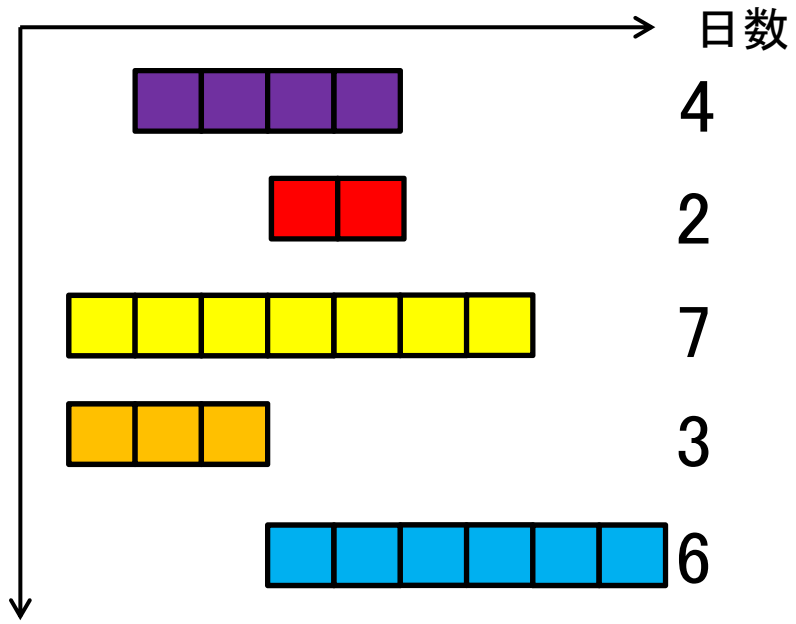
蔵置期間の長いブロックを初期配置

本研究



蔵置期間の長いブロックを初期配置

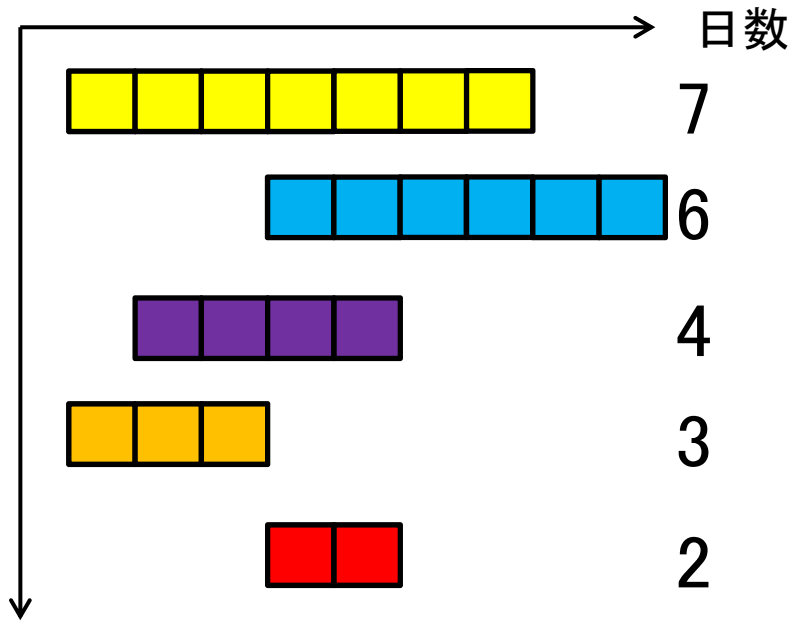
本研究



1．蔵置期間の長い順にブロックをソート

蔵置期間の長いブロックを初期配置

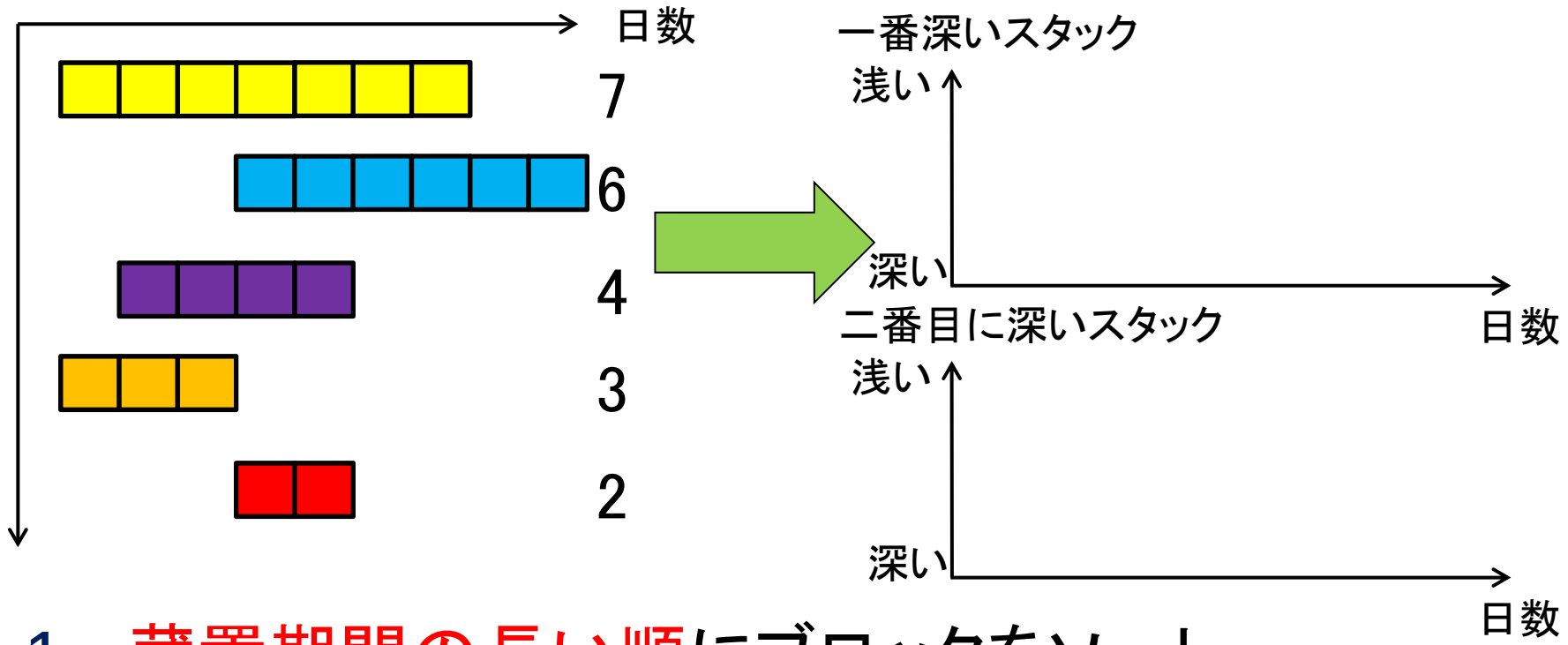
本研究



1．蔵置期間の長い順にブロックをソート

蔵置期間の長いブロックを初期配置

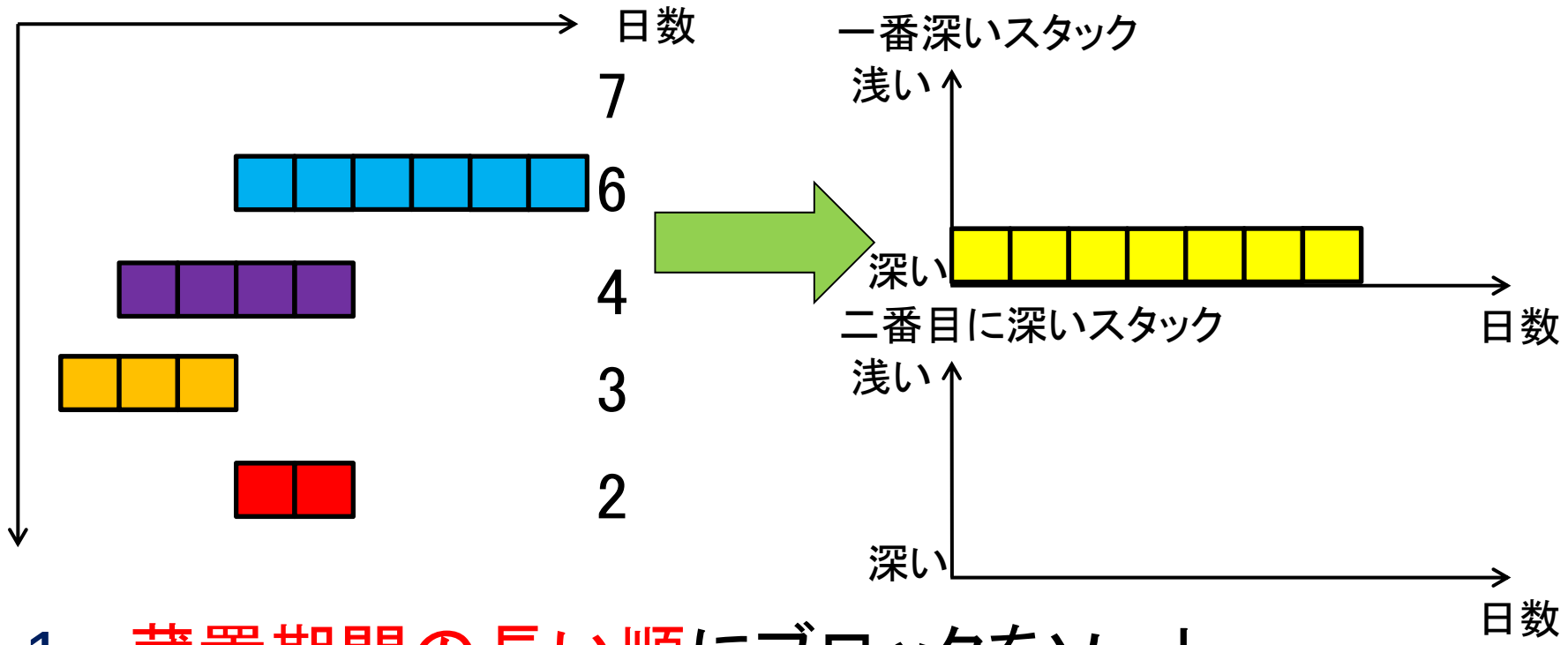
本研究



1. 蔵置期間の長い順にブロックをソート
2. ソートした順番に深いスタックから配置していく

蔵置期間の長いブロックを初期配置

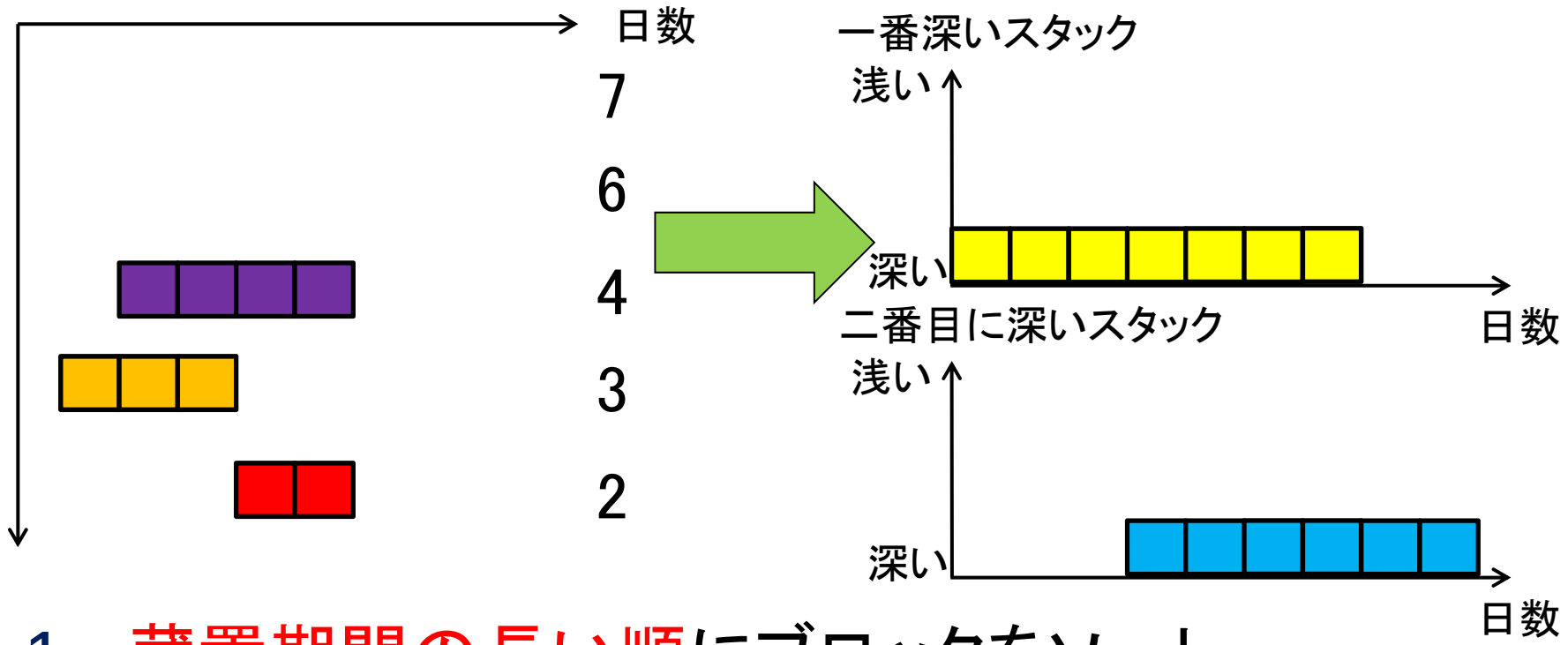
本研究



1. 蔵置期間の長い順にブロックをソート
2. ソートした順番に深いスタックから配置していく

蔵置期間の長いブロックを初期配置

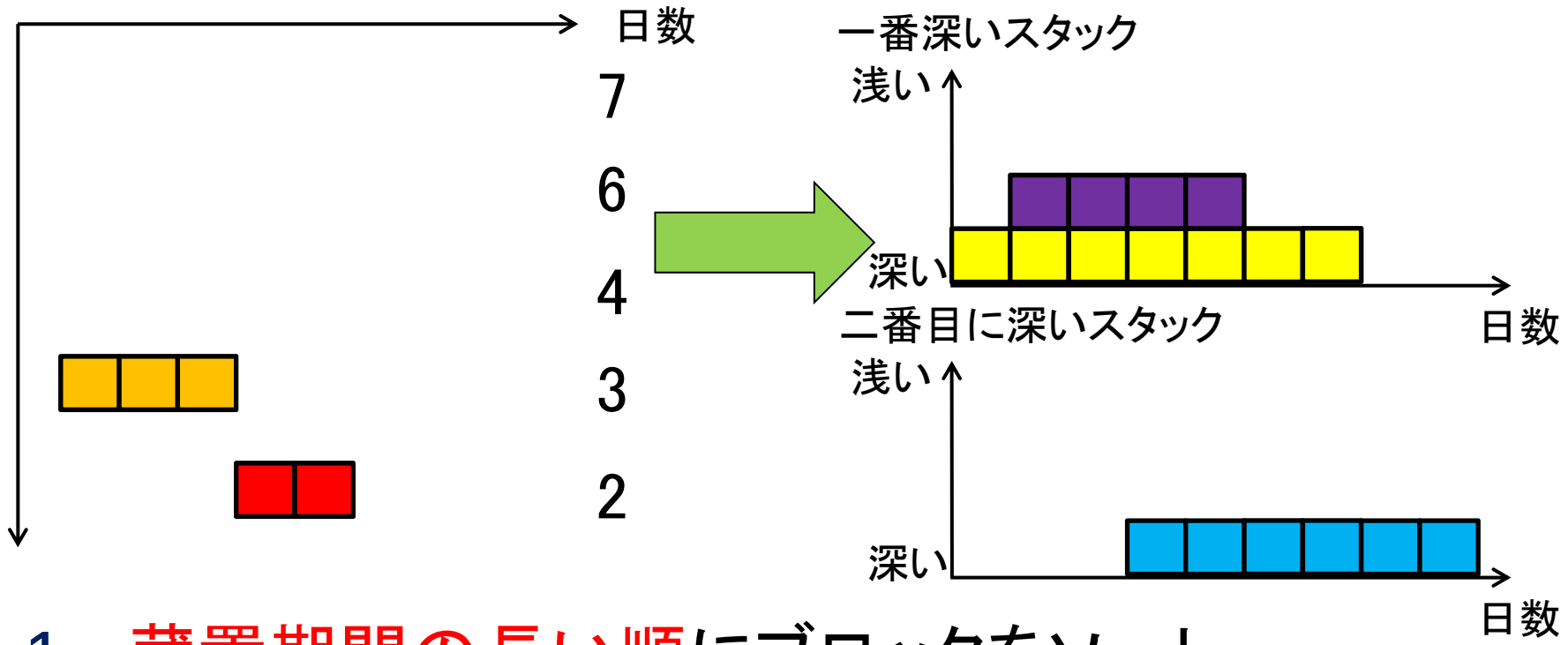
本研究



1. 蔵置期間の長い順にブロックをソート
2. ソートした順番に深いスタックから配置していく

蔵置期間の長いブロックを初期配置

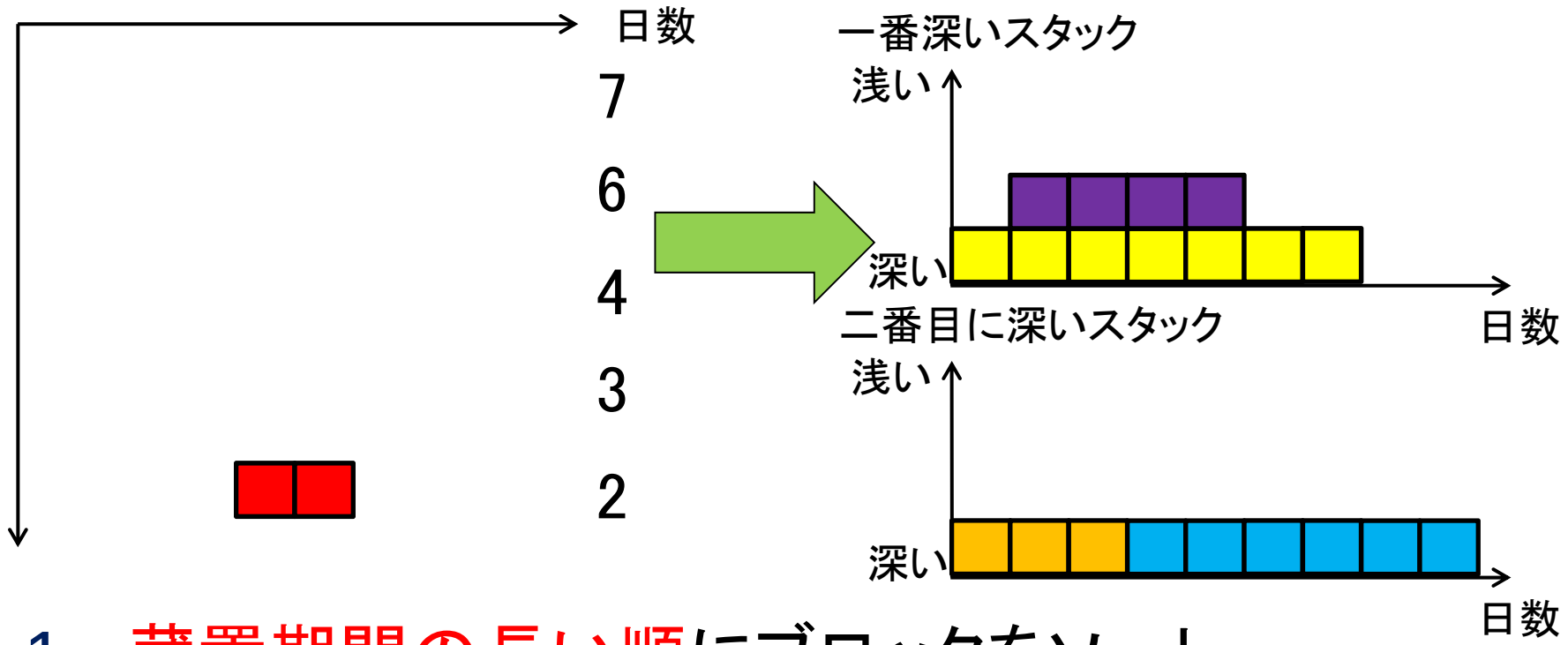
本研究



1. 蔵置期間の長い順にブロックをソート
2. ソートした順番に深いスタックから配置していく

蔵置期間の長いブロックを初期配置

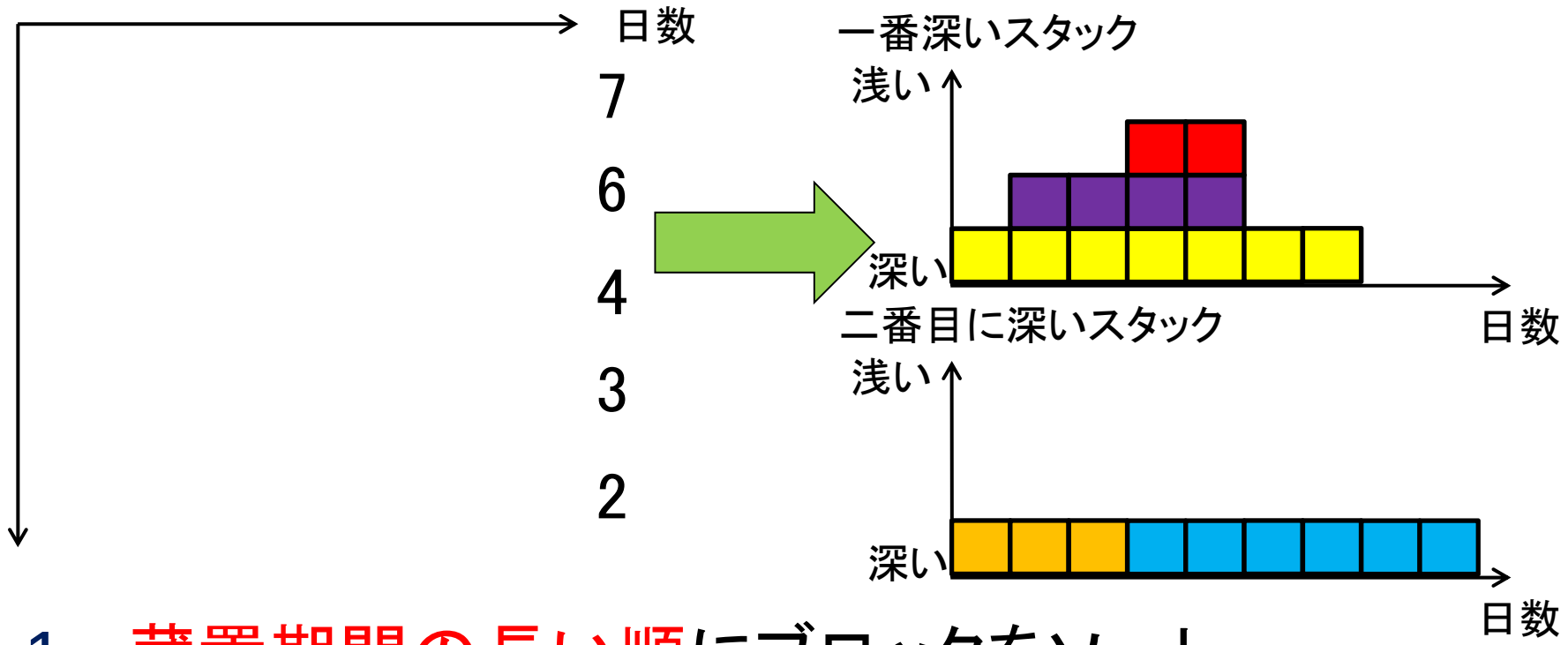
本研究



1. 蔵置期間の長い順にブロックをソート
2. ソートした順番に深いスタックから配置していく

蔵置期間の長いブロックを初期配置

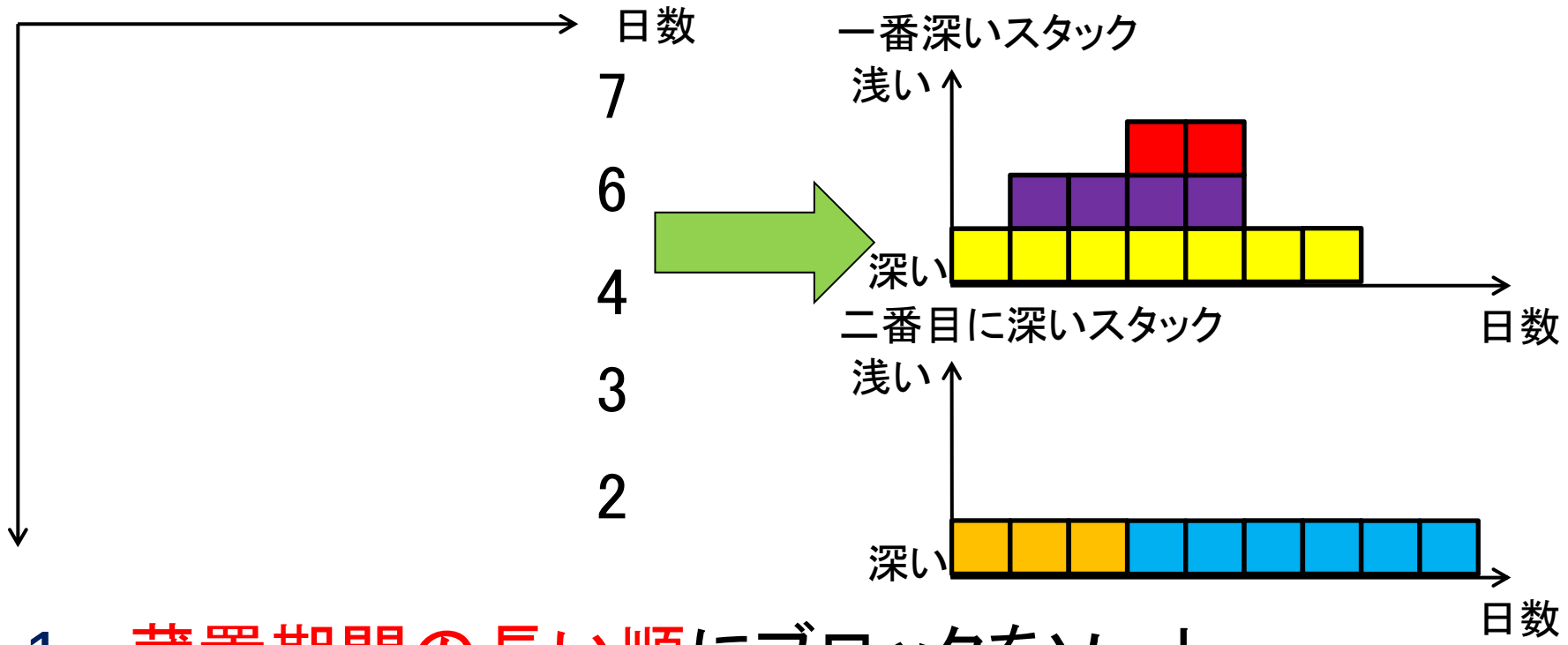
本研究



1. 蔵置期間の長い順にブロックをソート
2. ソートした順番に深いスタックから配置していく

蔵置期間の長いブロックを初期配置

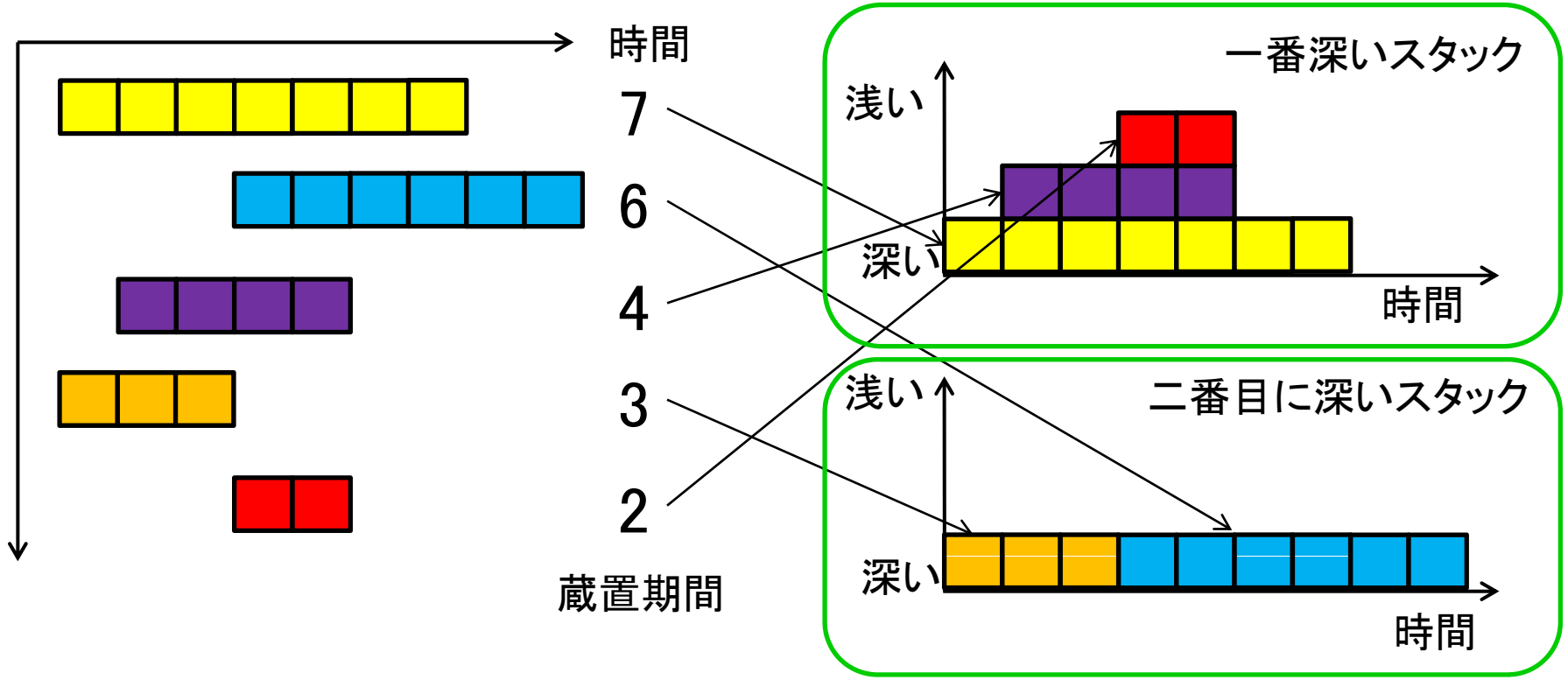
本研究



1. 蔵置期間の長い順にブロックをソート
2. ソートした順番に深いスタックから配置していく
3. 蔵置期間の長いブロックを配置後、分枝限定法で配置していく

蔵置期間の長いブロックを初期配置

本研究



本発表の概要

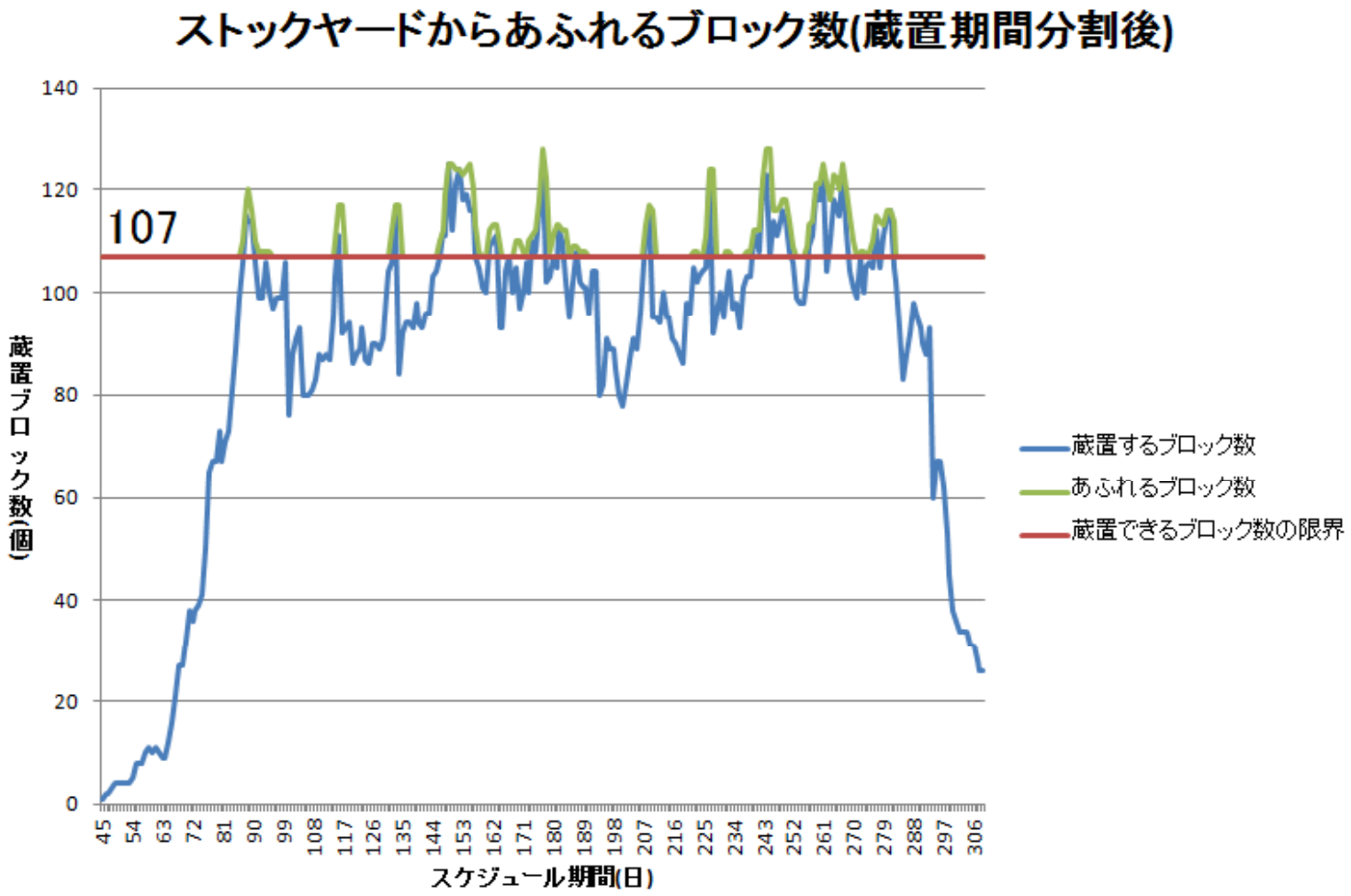
1. 背景と目的
   - 造船所のストックヤードの現状と問題
2. アプローチ
   - 造船所の置場情報とブロック蔵置ルールの設定
   - ブロック割り当てアルゴリズム
3. 結果と考察
   - 従来手法との比較
   - サンプルデータの生成と検討
4. 実用化への取り組み
   - リスケジューリングと属性情報の追加
5. まとめと今後の課題

スケジュール結果

エクセルで表示

全体のブロック数の蔵置期間の長いブロックの上位５%を初期配置し、深いスタックから配置していく分枝限定法1を用いてスケジューリングしたもの

スケジュール結果の検証



ブロックがストックヤードからあふれる日は、置場に蔵置されるブロック数の多い日と一致している

スケジュール結果の検証

分枝限定法1(枝300)を適用

	従来研究	分割蔵置改良後	初期配置5%	初期配置100%
ストックヤードの空きスペース	6855			
のべ未蔵置ブロック数	174			
のべ未蔵置ブロック期間	1312			
未蔵置ブロックの分割蔵置後				
ストックヤードの空きスペース	6111			
のべ未蔵置ブロック数	254			
のべ未蔵置ブロック期間	869			

スケジュール結果の検証

分枝限定法1(枝300)を適用

	従来研究	分割蔵置改良後	初期配置5%	初期配置100%
ストックヤードの空きスペース	6855	6855		
のべ未蔵置ブロック数	174	174		
のべ未蔵置ブロック期間	1312	1312		
未蔵置ブロックの分割蔵置後				
ストックヤードの空きスペース	6111	6113		
のべ未蔵置ブロック数	254	231		
のべ未蔵置ブロック期間	869	846		

2.6%のべ未蔵置ブロック期間が減少

ストックヤードの空きスペースは増加し、のべ未蔵置ブロック数とのべ未蔵置ブロック期間が減少

無駄なブロックの分割が減った

スケジュール結果の検証

蔵置期間の長い順にソートしたブロック数の上位5%（全蔵置期間の16%を占める）

分枝限定法1(枝300)を適用

	従来研究	分割蔵置改良後	初期配置5%	初期配置100%
ストックヤードの空きスペース	6855	6855	6593	
のべ未蔵置ブロック数	174	174	163	
のべ未蔵置ブロック期間	1312	1312	1050	
未蔵置ブロックの分割蔵置後				
ストックヤードの空きスペース	6111	6113	6153	
のべ未蔵置ブロック数	254	231	198	
のべ未蔵置ブロック期間	869	846	796	

20%のべ未蔵置ブロック期間が減少

5.9%のべ未蔵置ブロック期間が減少

未蔵置ブロックの分割蔵置前、分割蔵置後ともに、のべ未蔵置ブロック数とのべ未蔵置ブロック期間も減少した

→ 取り扱いの困難な蔵置期間の長いブロックを初期配置するのは有効な手段

スケジュール結果の検証

分枝限定法1(枝300)を適用

	従来研究	分割蔵置改良後	初期配置5%	初期配置100%
ストックヤードの空きスペース	6855	6855	6593	6766
のべ未蔵置ブロック数	174	174	163	386
のべ未蔵置ブロック期間	1312	1312	1050	1223
未蔵置ブロックの分割蔵置後				
ストックヤードの空きスペース	6111	6113	6153	
のべ未蔵置ブロック数	254	231	198	
のべ未蔵置ブロック期間	869	846	796	990

6.8%のべ未蔵置ブロック期間が減少

分枝限定法を用いず、蔵置期間の長いブロックから詰めていくとのべ未蔵置ブロック期間は減少する

→ 蔵置期間の長いブロックから順に処理するという手法はシンプルだが非常に有効

スケジュール結果の検証

分枝限定法1(枝300)を適用

	従来研究	分割蔵置改良後	初期配置5%	初期配置100%
ストックヤードの空きスペース	6855	6855	6593	6766
のべ未蔵置ブロック数	174	174	163	
のべ未蔵置ブロック期間	1312	1312	1050	
未蔵置ブロックの分割蔵置後				
ストックヤードの空きスペース	6111	6113	6153	6356
のべ未蔵置ブロック数	254	231	198	361
のべ未蔵置ブロック期間	869	846	796	990

17%のべ未蔵置ブロック期間が増加

分枝限定法を用いず、蔵置期間の長いブロックから詰めていくと分割蔵置後ののべ未蔵置ブロック期間は悪くなる

→ コストを定義していないため、ブロックがぎっしり詰めて配置されておらず、デッドスペースが発生している

スケジュール結果の検証

分枝限定法1(枝300)を適用

	従来研究	分割蔵置改良後	初期配置5%	初期配置100%
ストックヤードの空きスペース	6855	6855	6593	6766
のべ未蔵置ブロック数	174	174	163	386
のべ未蔵置ブロック期間	1312	1312	1050	1223
未蔵置ブロックの分割蔵置後				
ストックヤードの空きスペース	6111	6113	6153	
のべ未蔵置ブロック数	254	231	198	
のべ未蔵置ブロック期間	869	846	796	

8.4%のべ未蔵置ブロック期間が減少

分割蔵置の改良、蔵置期間の長いブロックの上位5%を初期配置することで、従来研究より置場からあふれるブロックを8．4%減らすことができた

仮想データの生成

本研究

- 造船所から提供されたブロックデータだけではスケジューラーの性能の検証が不十分
- そこで造船所のブロックデータを元に

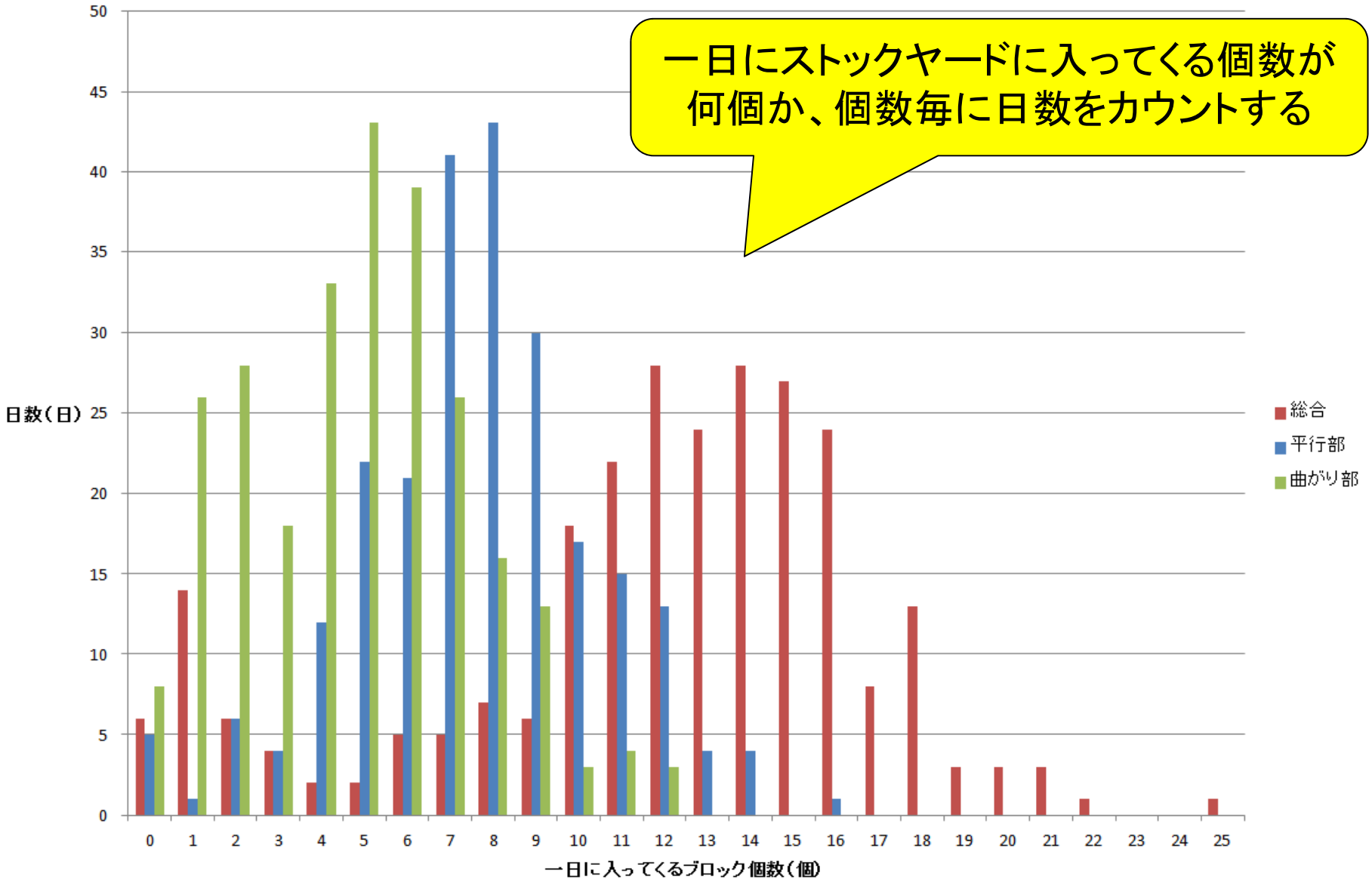
  ①「一日にストックヤードに入ってくる個数」

  ②「各ブロックの蔵置期間」

  の2つのヒストグラムを作成
- このヒストグラムを元に確率分布に従って、仮想データを生成する

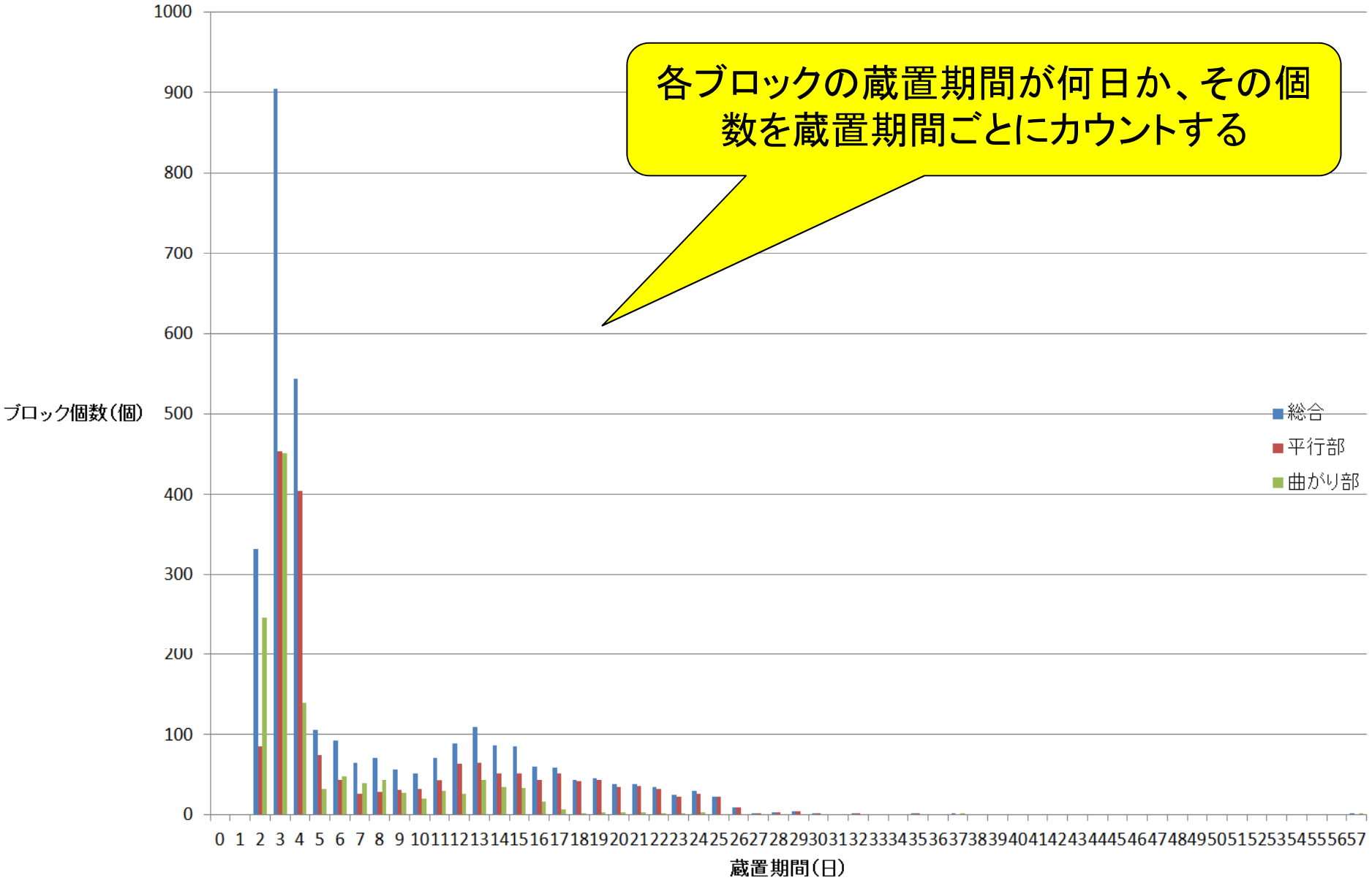
仮想データの生成

本研究



仮想データの生成

本研究



仮想データの生成

本研究

1. 「一日にストックヤードに入ってくる個数」のヒストグラムから、その日にブロックを何個生成するか決定

生成した仮想データ

1	0	2	製造ナンバー:1	MAGARIBU↓
2	0	8	製造ナンバー:2	MAGARIBU↓
3	0	1	製造ナンバー:3	MAGARIBU↓
4	0	4	製造ナンバー:4	MAGARIBU↓
5	0	20	製造ナンバー:5	MAGARIBU↓
6	0	6	製造ナンバー:6	MAGARIBU↓
7	0	16	製造ナンバー:7	MAGARIBU↓
8	0	31	製造ナンバー:8	MAGARIBU↓
9	0	3	製造ナンバー:9	MAGARIBU↓
10	1	7	製造ナンバー:10	MAGARIBU↓
11	1	14	製造ナンバー:11	MAGARIBU↓
12	1	3	製造ナンバー:12	MAGARIBU↓
13	1	18	製造ナンバー:13	MAGARIBU↓
14	1	4	製造ナンバー:14	MAGARIBU↓
15	1	4	製造ナンバー:15	MAGARIBU↓
16	1	3	製造ナンバー:16	MAGARIBU↓
17	1	18	製造ナンバー:17	MAGARIBU↓
18	1	17	製造ナンバー:18	MAGARIBU↓
19	1	13	製造ナンバー:19	MAGARIBU↓
20	1	3	製造ナンバー:20	MAGARIBU↓
21	2	14	製造ナンバー:21	MAGARIBU↓
22	2	15	製造ナンバー:22	MAGARIBU↓
23	2	5	製造ナンバー:23	MAGARIBU↓
24	2	13	製造ナンバー:24	MAGARIBU↓
25	2	4	製造ナンバー:25	MAGARIBU↓
26	2	6	製造ナンバー:26	MAGARIBU↓
27	2	24	製造ナンバー:27	MAGARIBU↓
28	2	4	製造ナンバー:28	MAGARIBU↓
29	3	14	製造ナンバー:29	MAGARIBU↓
30	3	6	製造ナンバー:30	MAGARIBU↓
31	3	6	製造ナンバー:31	MAGARIBU↓
32	3	7	製造ナンバー:32	MAGARIBU↓
33	3	16	製造ナンバー:33	MAGARIBU↓
34	3	9	製造ナンバー:34	MAGARIBU↓
35	3	15	製造ナンバー:35	MAGARIBU↓
36	3	13	製造ナンバー:36	MAGARIBU↓
37	3	6	製造ナンバー:37	MAGARIBU↓
38	4	17	製造ナンバー:38	MAGARIBU↓
39	4	25	製造ナンバー:39	MAGARIBU↓
40	4	21	製造ナンバー:40	MAGARIBU↓

仮想データの生成

本研究

1. 「一日にストックヤードに入ってくる個数」のヒストグラムから、その日にブロックを何個生成するか決定
2. 「各ブロックの蔵置期間」のヒストグラムからそのブロックの蔵置期間を決定

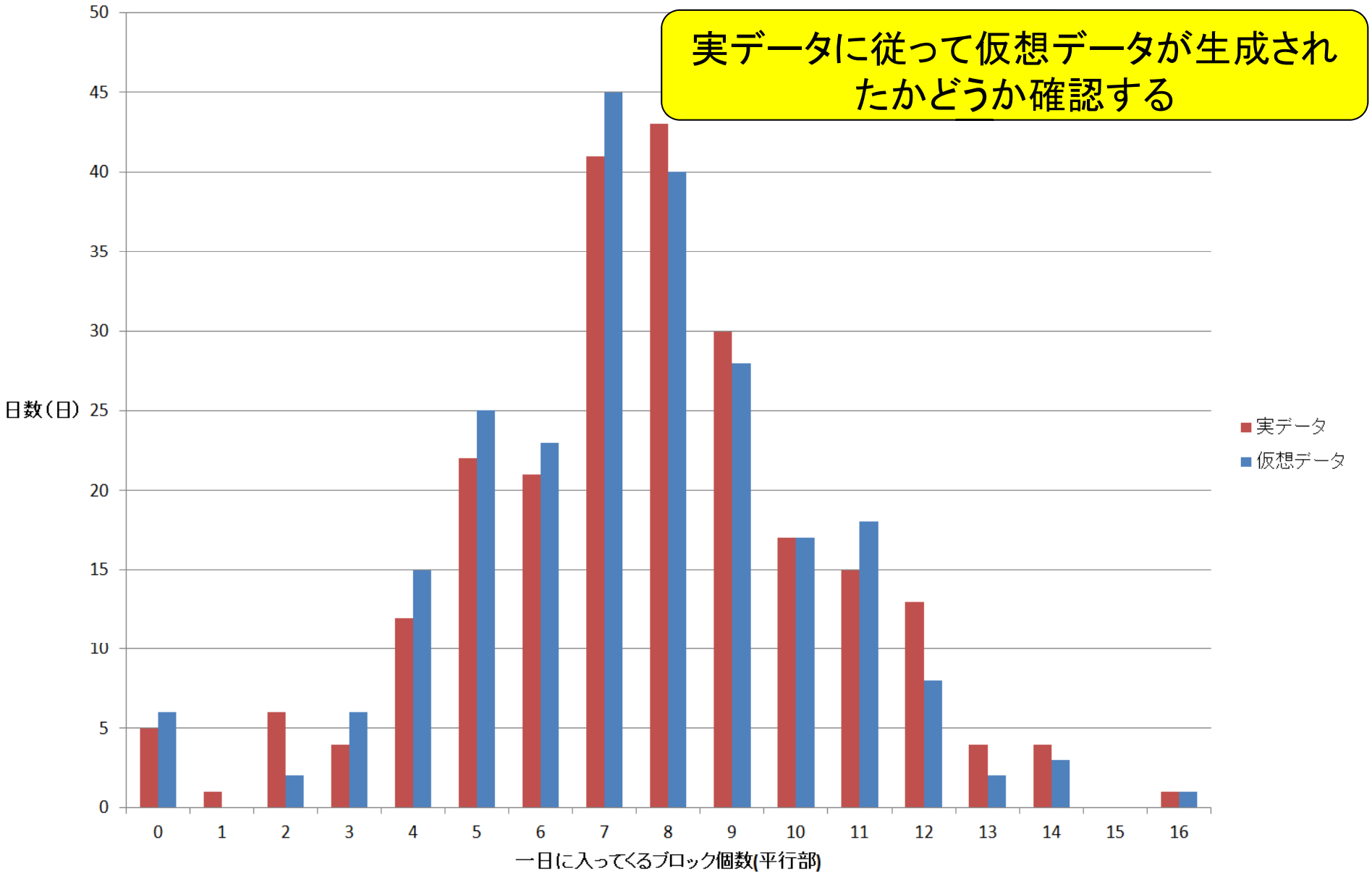
生成した仮想データ

1	0	2	製造ナンバー:1	MAGARIBU
2	0	8	製造ナンバー:2	MAGARIBU
3	0	1	製造ナンバー:3	MAGARIBU
4	0	4	製造ナンバー:4	MAGARIBU
5	0	20	製造ナンバー:5	MAGARIBU
6	0	6	製造ナンバー:6	MAGARIBU
7	0	16	製造ナンバー:7	MAGARIBU
8	0	31	製造ナンバー:8	MAGARIBU
9	0	3	製造ナンバー:9	MAGARIBU
10	1	7	製造ナンバー:10	MAGARIBU
11	1	14	製造ナンバー:11	MAGARIBU
12	1	3	製造ナンバー:12	MAGARIBU
13	1	18	製造ナンバー:13	MAGARIBU
14	1	4	製造ナンバー:14	MAGARIBU
15	1	4	製造ナンバー:15	MAGARIBU
16	1	3	製造ナンバー:16	MAGARIBU
17	1	18	製造ナンバー:17	MAGARIBU
18	1	17	製造ナンバー:18	MAGARIBU
19	1	13	製造ナンバー:19	MAGARIBU
20	1	3	製造ナンバー:20	MAGARIBU
21	2	14	製造ナンバー:21	MAGARIBU
22	2	15	製造ナンバー:22	MAGARIBU
23	2	5	製造ナンバー:23	MAGARIBU
24	2	13	製造ナンバー:24	MAGARIBU
25	2	4	製造ナンバー:25	MAGARIBU
26	2	6	製造ナンバー:26	MAGARIBU
27	2	24	製造ナンバー:27	MAGARIBU
28	2	4	製造ナンバー:28	MAGARIBU
29	3	14	製造ナンバー:29	MAGARIBU
30	3	6	製造ナンバー:30	MAGARIBU
31	3	6	製造ナンバー:31	MAGARIBU
32	3	7	製造ナンバー:32	MAGARIBU
33	3	16	製造ナンバー:33	MAGARIBU
34	3	9	製造ナンバー:34	MAGARIBU
35	3	15	製造ナンバー:35	MAGARIBU
36	3	13	製造ナンバー:36	MAGARIBU
37	3	6	製造ナンバー:37	MAGARIBU
38	4	17	製造ナンバー:38	MAGARIBU
39	4	25	製造ナンバー:39	MAGARIBU
40	4	21	製造ナンバー:40	MAGARIBU

仮想データの生成確認

本研究

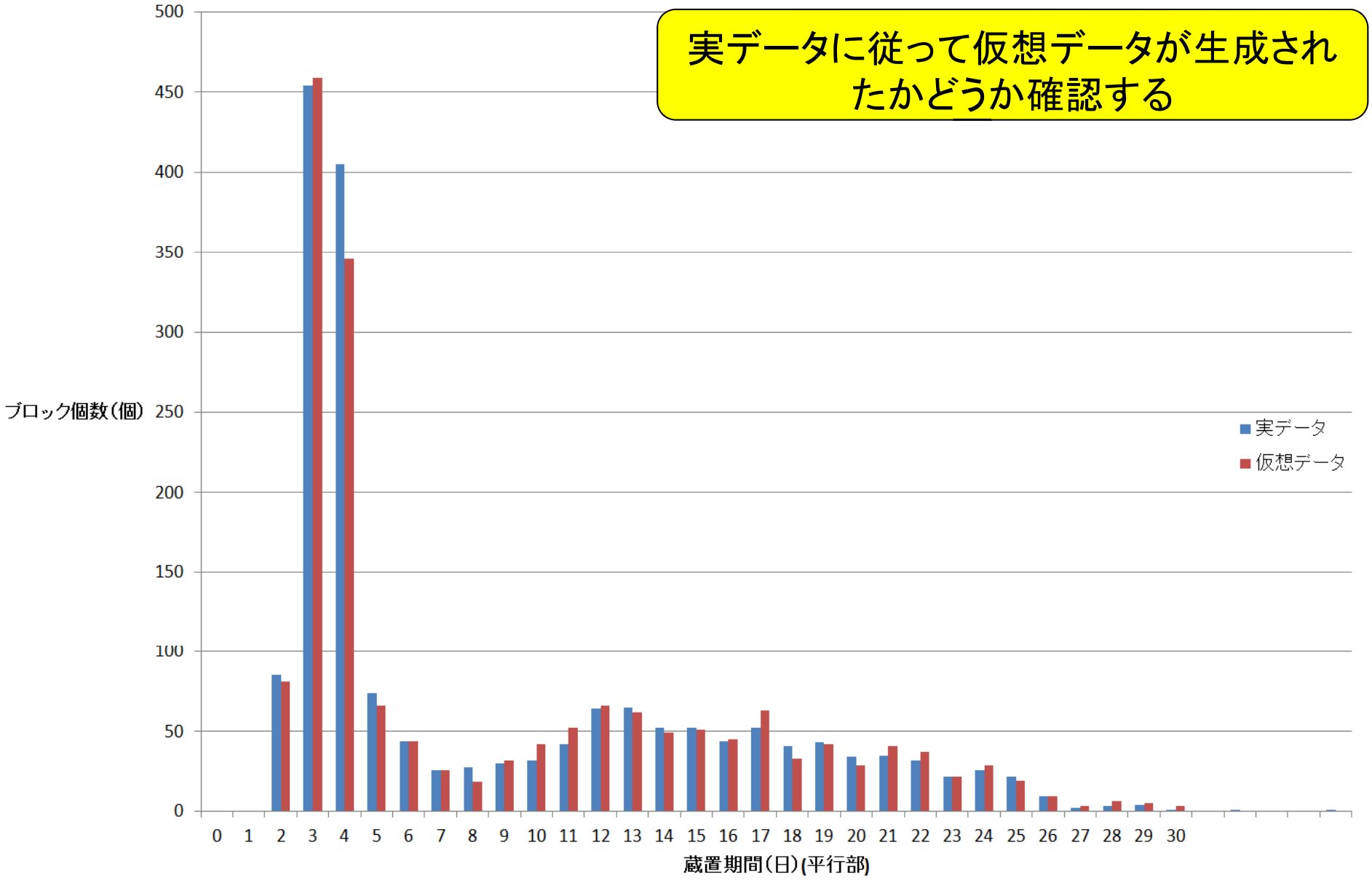
実データに従って仮想データが生成されたかどうか確認する



仮想データの生成確認

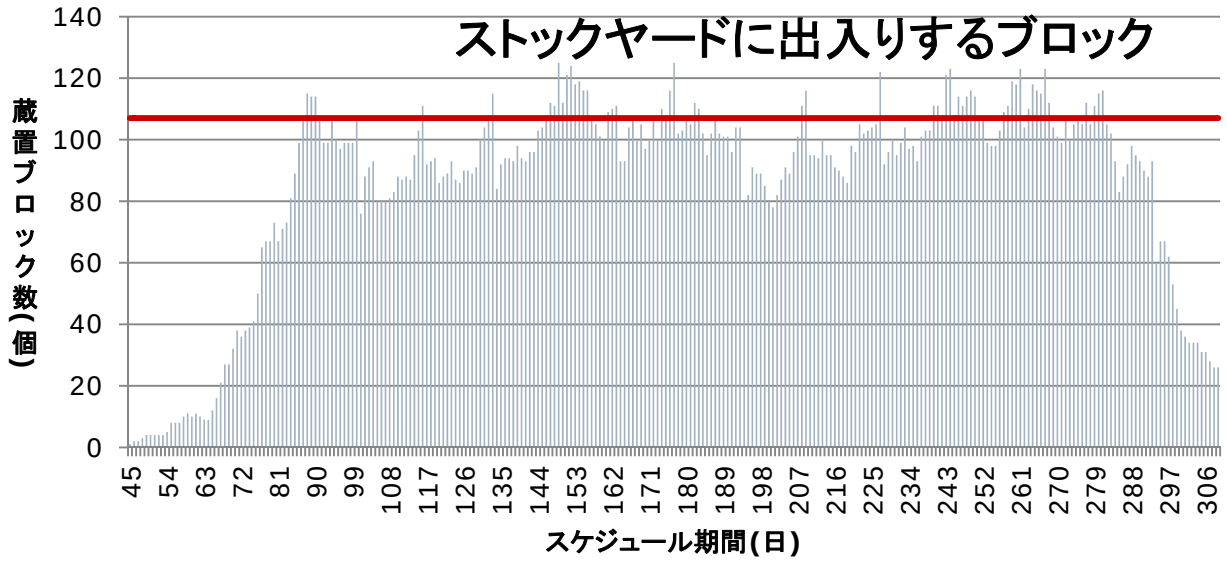
本研究

実データに従って仮想データが生成されたかどうか確認する

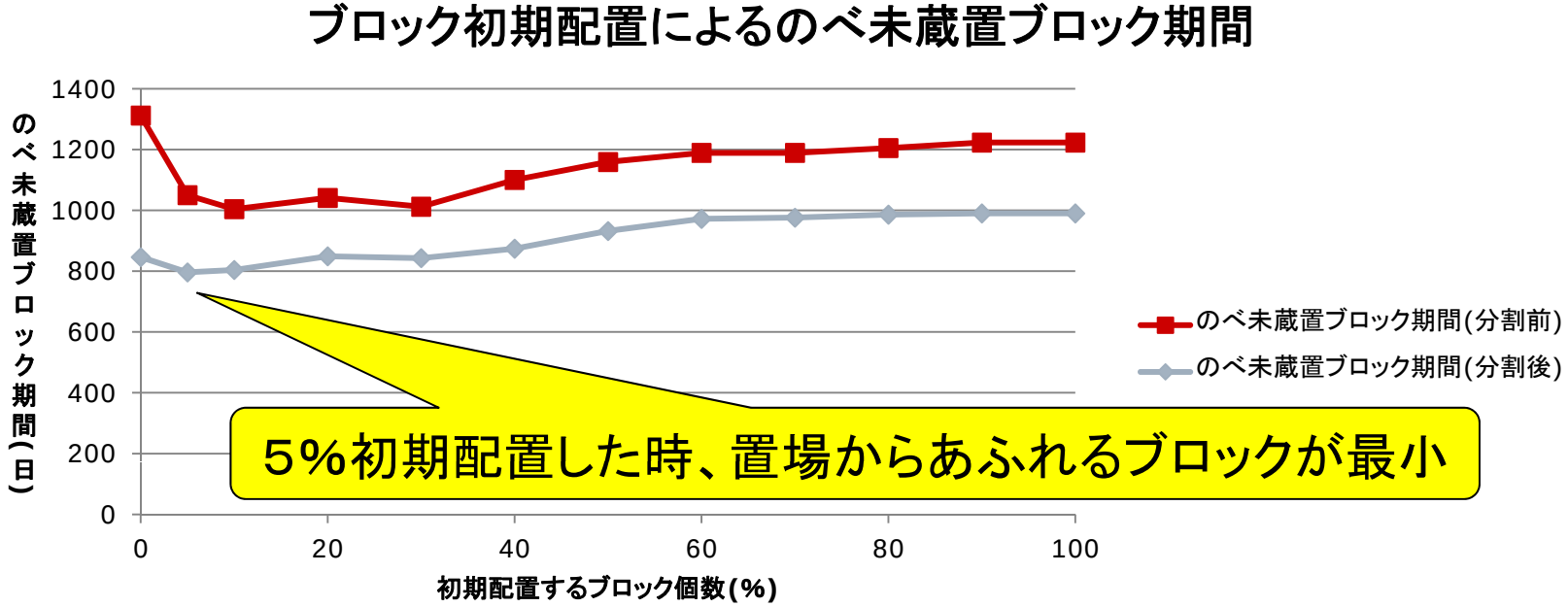


スケジュール結果(実データ)

本研究



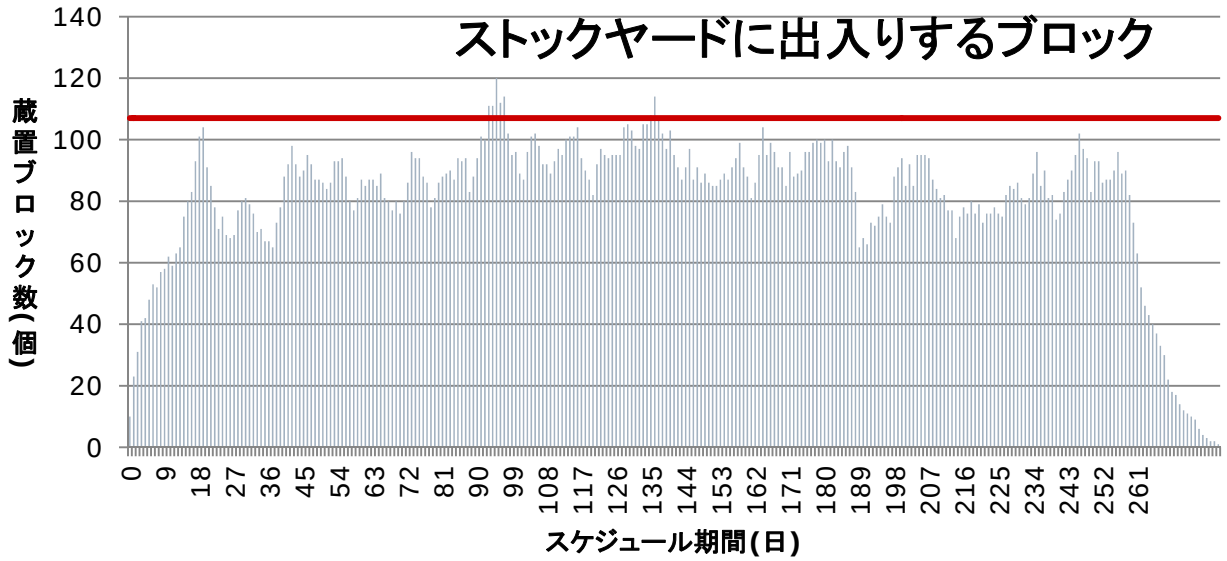
ブロック数:3079
蔵置限界を超えた
ブロック数:412



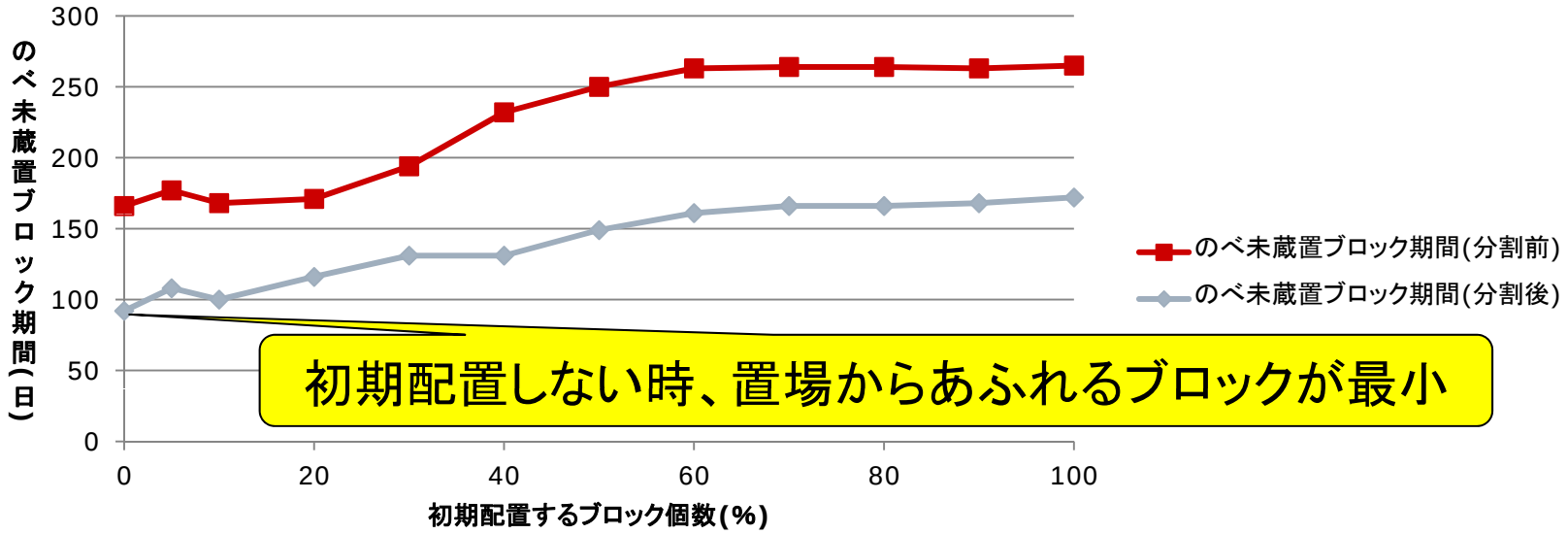
5%初期配置した時、置場からあふれるブロックが最小

スケジュール結果(仮想データ1)

本研究



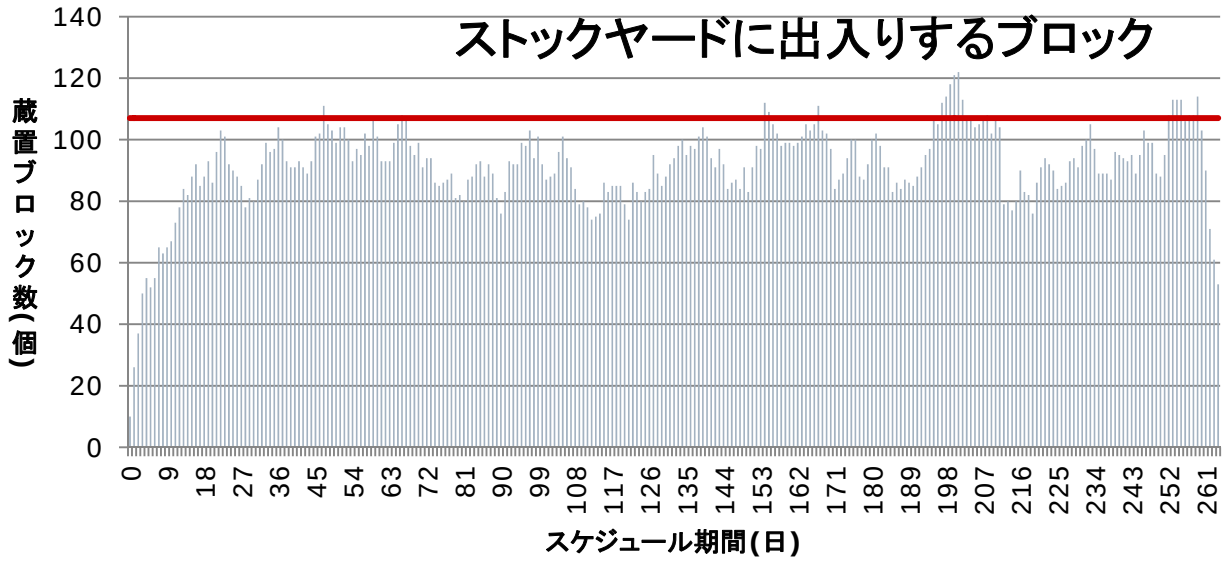
ブロック数:3079
蔵置限界を超えた
ブロック数:41



初期配置しない時、置場からあふれるブロックが最小

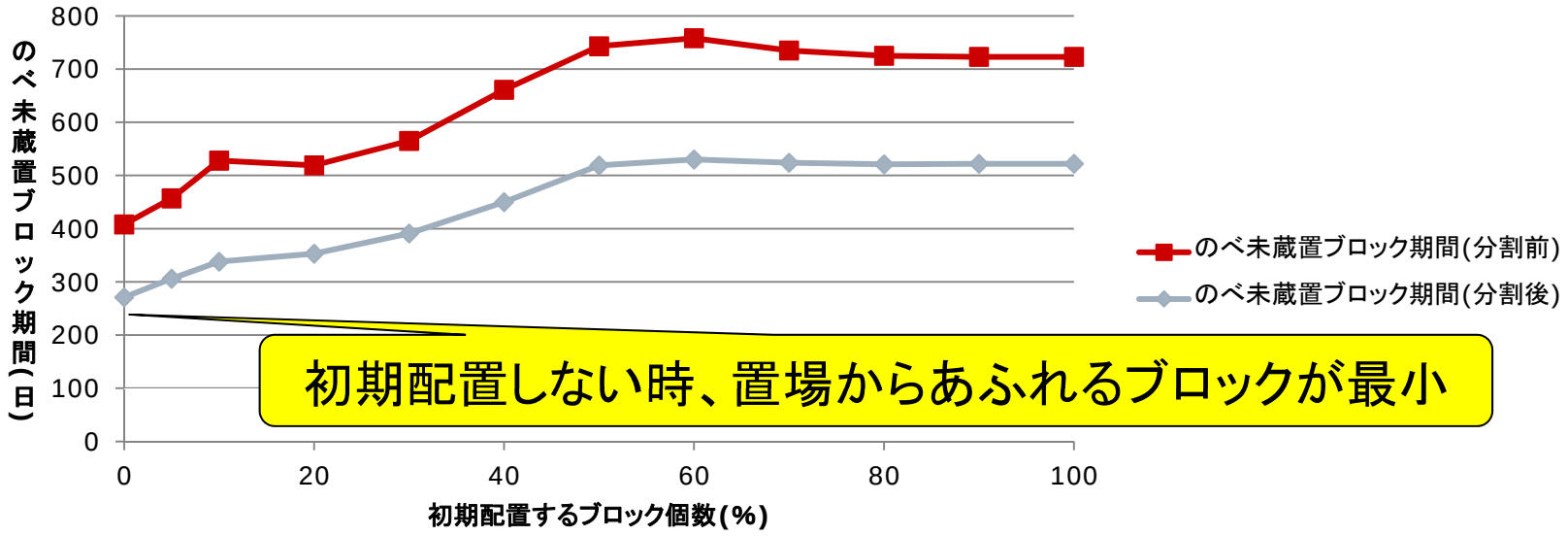
スケジュール結果(仮想データ2)

本研究



ブロック数:3244
蔵置限界を超えた
ブロック数:103

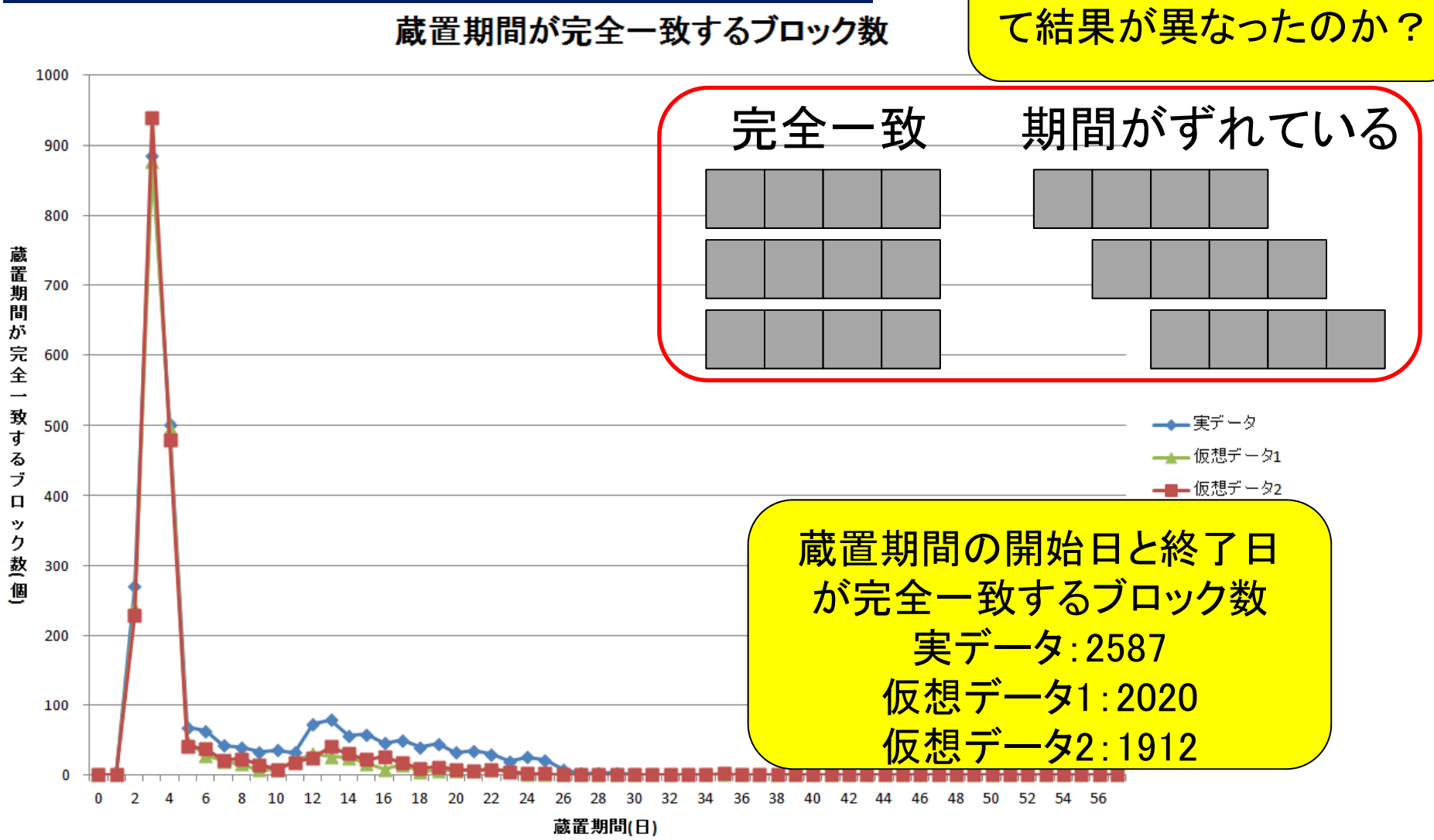
蔵置するブロック数
蔵置できる限界ブロック数



初期配置しない時、置場からあふれるブロックが最小

スケジュール結果の考察

なぜ、実データと仮想データで初期配置によって結果が異なったのか？

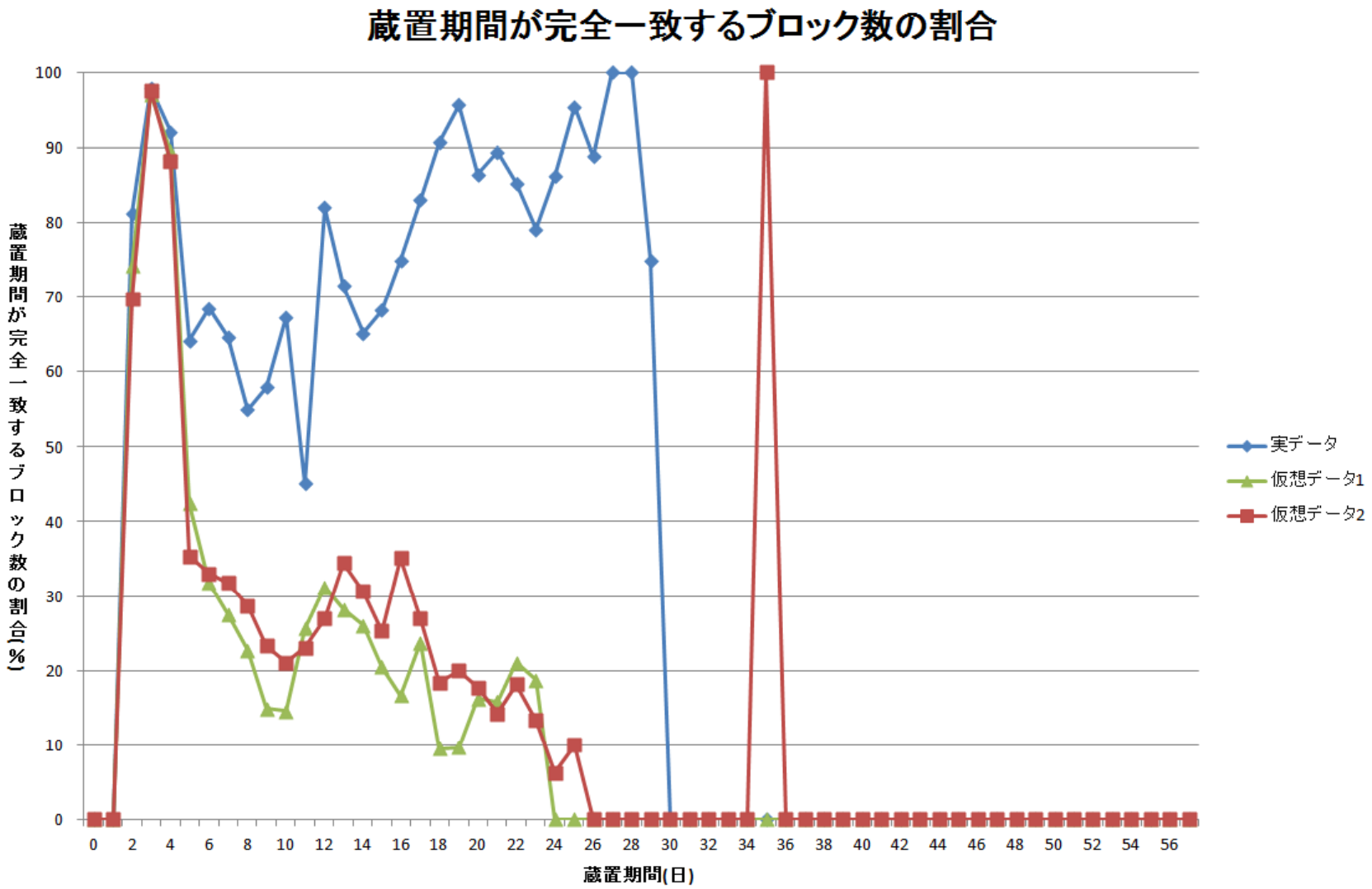


蔵置期間の開始日と終了日が完全一致するブロック数
実データ:2587
仮想データ1:2020
仮想データ2:1912

造船所の実データは仮想データより
蔵置期間の一致するブロックが多い

スケジュール結果の考察

本研究



実データは仮想データより蔵置期間が完全一致するブロックの割合が、蔵置期間が長くなっても高い割合を占めている

スケジュール結果の考察

本研究

	実データ	仮想データ1	仮想データ2
ブロック数	3079	3079	3244
蔵置限界を超えたブロック数	412	41	103

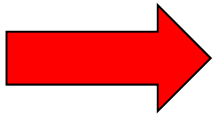
① 造船所の実データは蔵置限界を超えるブロックが仮想データより多い

スケジュール結果の考察

本研究

	実データ	仮想データ1	仮想データ2
ブロック数	3079	3079	3244
蔵置限界を超えたブロック数	412	41	103

① 造船所の実データは蔵置限界を超えるブロックが仮想データより多い



ブロックの蔵置日が集中している

スケジュール結果の考察

本研究

	実データ	仮想データ1	仮想データ2
ブロック数	3079	3079	3244
蔵置限界を超えたブロック数	412	41	103

① 造船所の実データは蔵置限界を超えるブロックが仮想データより多い → ブロックの蔵置日が集中している

② 蔵置期間の完全一致するブロックの数やその割合が非常に高い

スケジュール結果の考察

本研究

	実データ	仮想データ1	仮想データ2
ブロック数	3079	3079	3244
蔵置限界を超えたブロック数	412	41	103

① 造船所の実データは蔵置限界を超えるブロックが仮想データより多い → ブロックの蔵置日が集中している

② 蔵置期間の完全一致するブロックの数やその割合が非常に高い

ストックヤードに蔵置されている期間にドック搬入日などを考慮した規則性があるように思われる

スケジュール結果の考察

本研究

	実データ	仮想データ1	仮想データ2
ブロック数	3079	3079	3244
蔵置限界を超えたブロック数	412	41	103

① 造船所の実データは蔵置限界を超えるブロックが仮想データより多い → ブロックの蔵置日が集中している

② 蔵置期間の完全一致するブロックの数やその割合が非常に高い

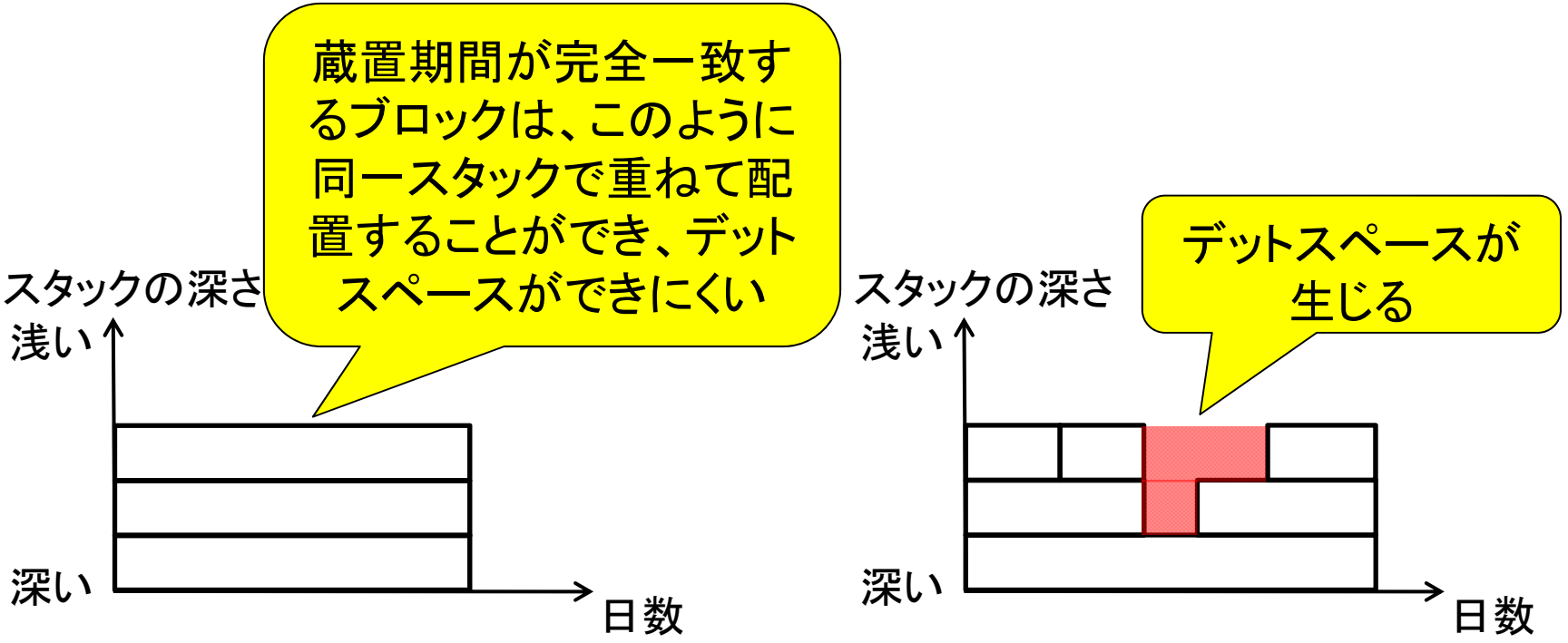
ストックヤードに蔵置されている期間にドック搬入日などを考慮した規則性があるように思われる

仮想データの生成はこの規則性を考慮する必要がある

スケジュール結果の考察

本研究

- ブロックを蔵置期間の長いものから初期配置する手法は、造船所の実データのように蔵置期間が完全一致する蔵置期間の長いブロックが多い場合において有効な手法であると推測される



本発表の概要

1. 背景と目的
   - ➢ 造船所のストックヤードの現状と問題
2. アプローチ
   - ➢ 造船所の置場情報とブロック蔵置ルールの設定
   - ➢ ブロック割り当てアルゴリズム
3. 結果と考察
   - ➢ 従来手法との比較
   - ➢ サンプルデータの生成と検討
4. 実用化への取り組み
   - ➢ リスケジューリングと属性情報の追加
5. まとめと今後の課題

リスケジューリング

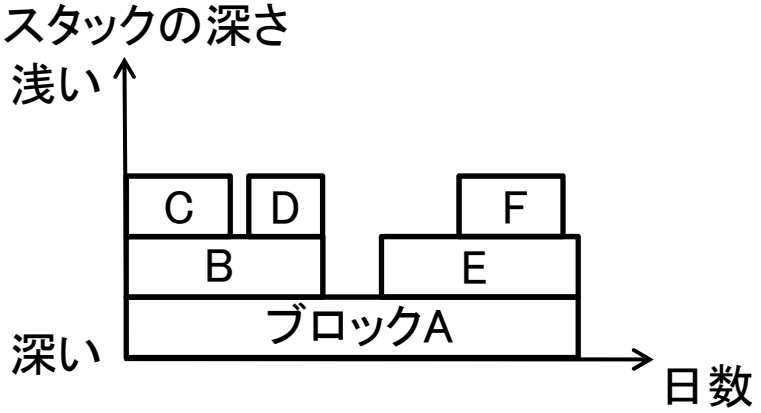
本研究

- 初期計画時にブロックの製造工程は決められているが、製造工程の遅れや計画変更に伴い、ストックヤードに入ってくるブロックの数や蔵置期間が変更される
- これに対応するために現場では、任意の日からスケジュールをやり直すリスケジューリング機能が求められる

リスケジューリング

本研究

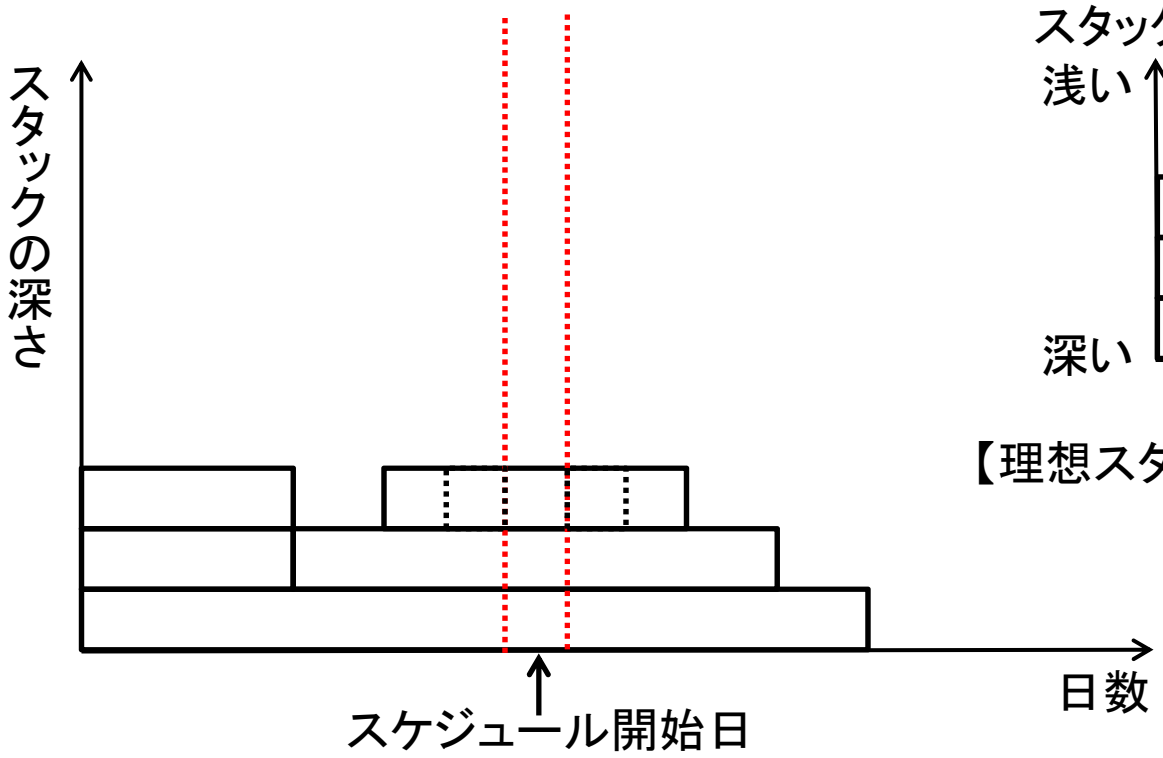
- リスケジューリングを行う際に、まずスケジュール開始日においてブロックが理想スタック蔵置ルールに従って置かれているかどうか確かめる
- 理想スタック蔵置ルールを満たしていないブロックについては蔵置期間を分割し、今後、無駄な荷繰りが発生しないようにする
- 全てのスタックで理想スタック蔵置ルールを満たした後、スケジュールする

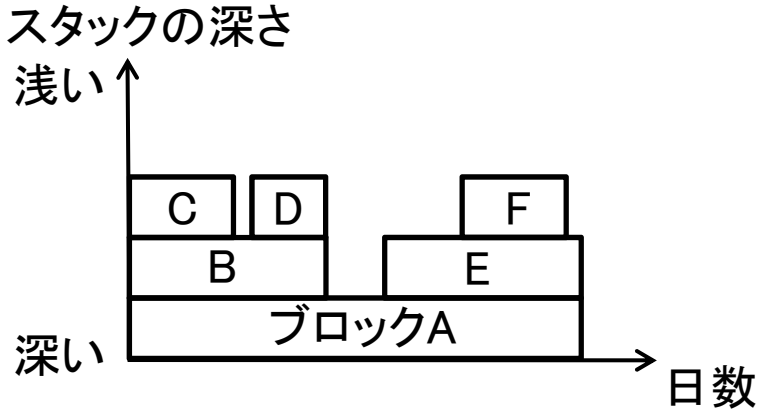


【理想スタック蔵置ルールに従う置き方】

リスケジューリング

本研究



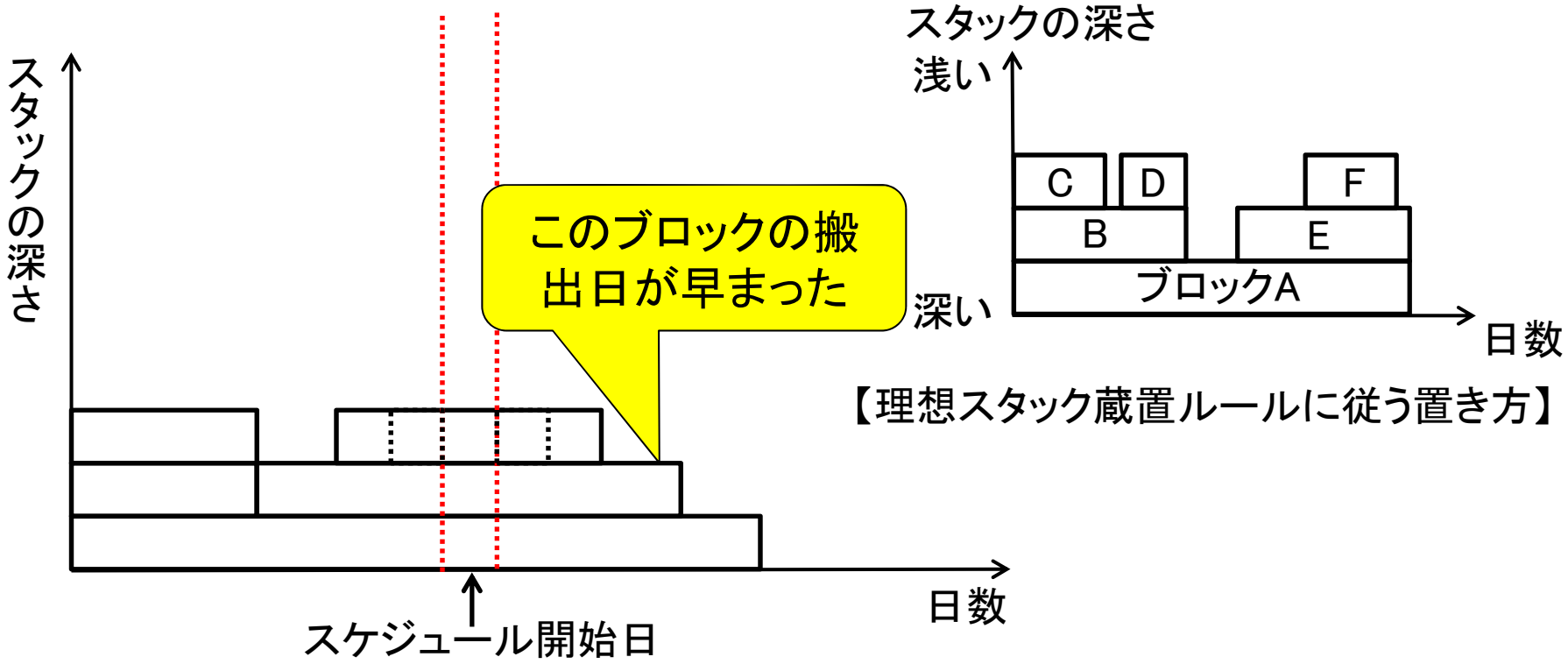


【理想スタック蔵置ルールに従う置き方】

- ストックヤードの現在蔵置されているブロックの情報を読み込む

リスケジューリング

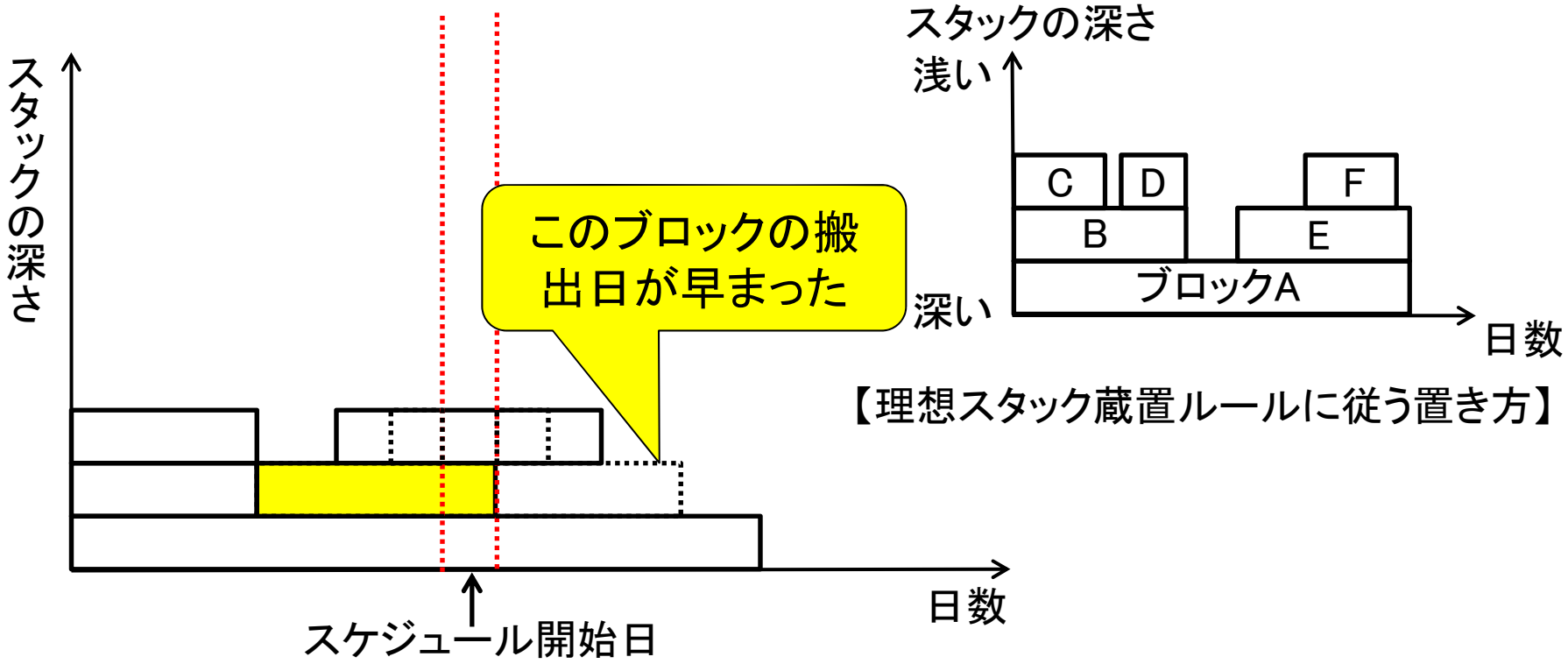
本研究



- 作業工程の遅れなどにより蔵置計画が変更されたブロックに着目

リスケジューリング

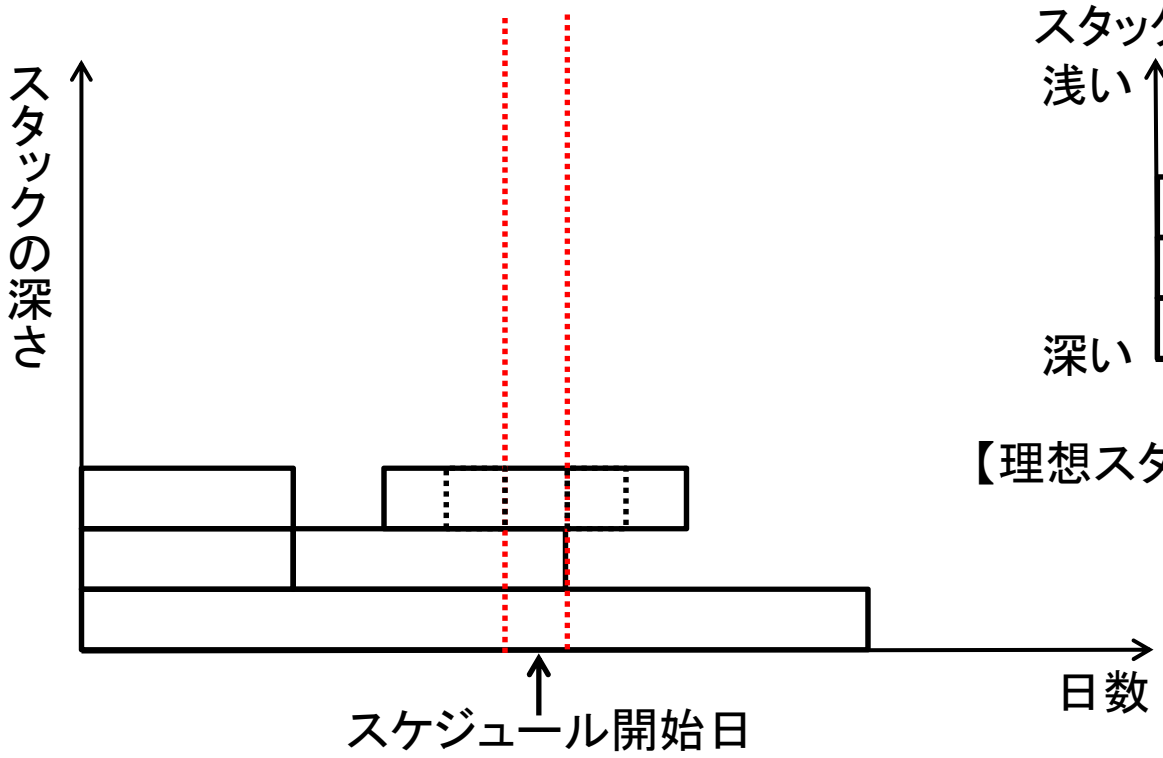
本研究

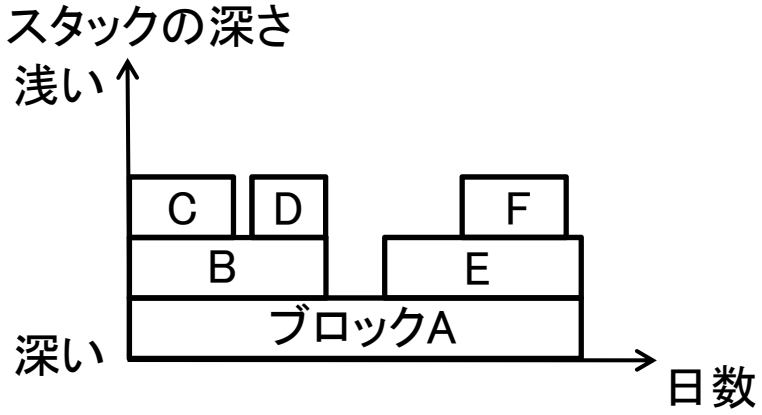


- 対象ブロックの蔵置期間を変更する

リスケジューリング

本研究



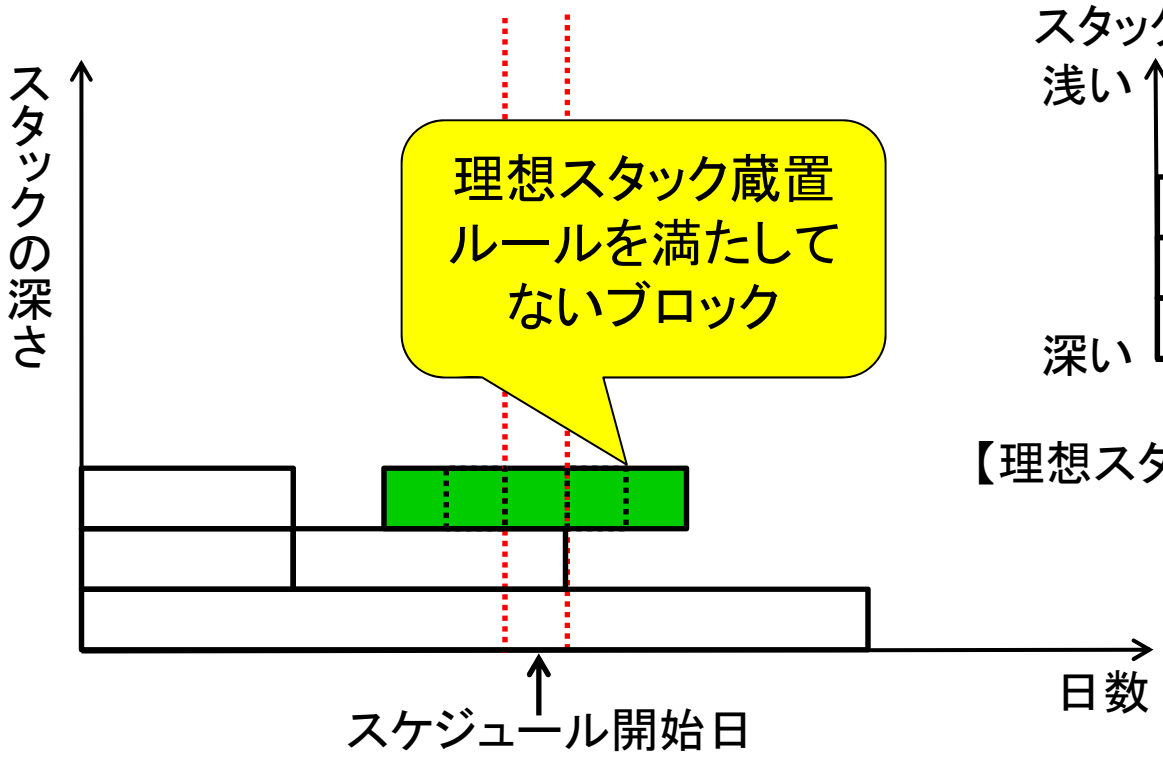


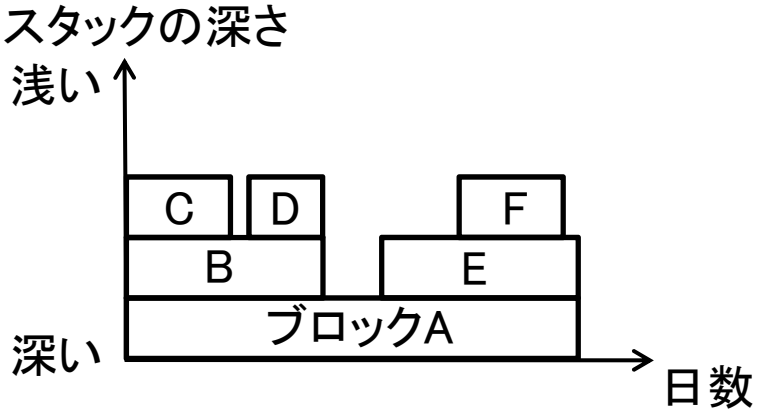
【理想スタック蔵置ルールに従う置き方】

- スケジューリング開始日の各スタックごとに理想スタック蔵置ルールを満たしていないブロックを探す

リスケジューリング

本研究



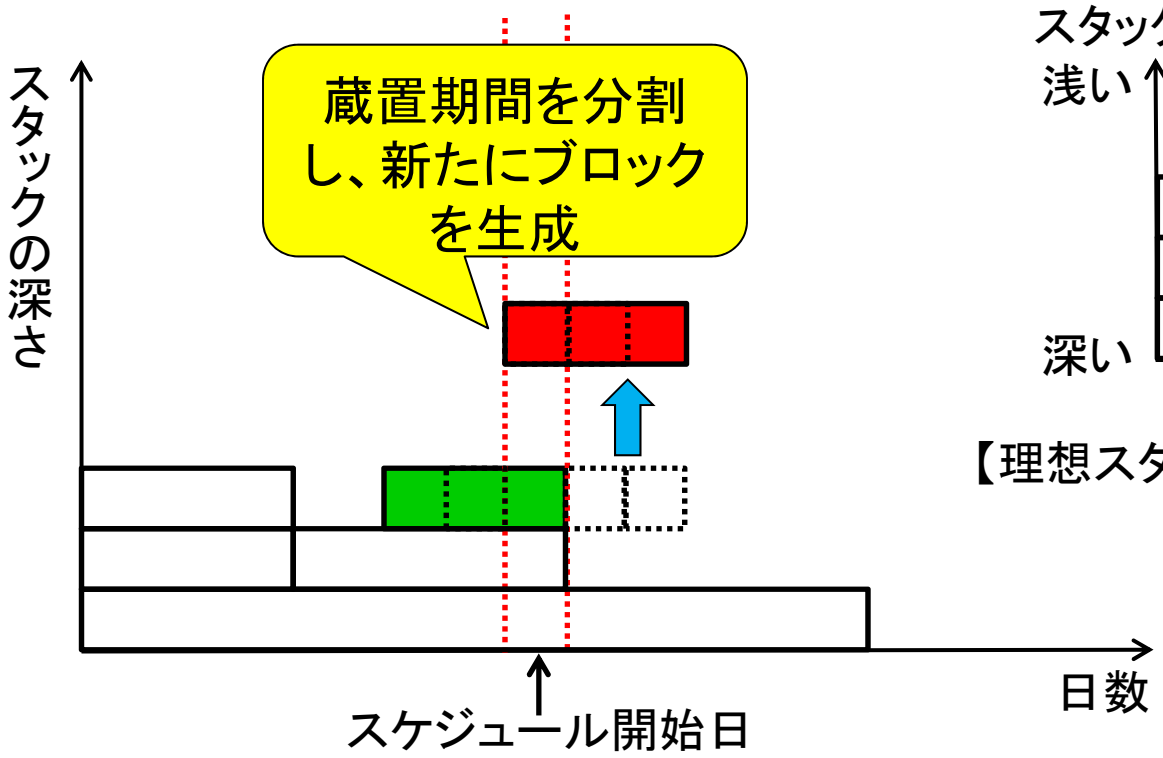


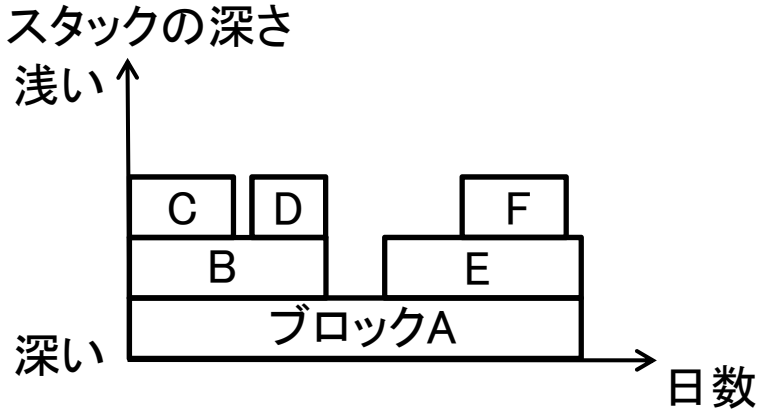
【理想スタック蔵置ルールに従う置き方】

- スケジューリング開始日の各スタックごとに理想スタック蔵置ルールを満たしていないブロックを探す

リスケジューリング

本研究



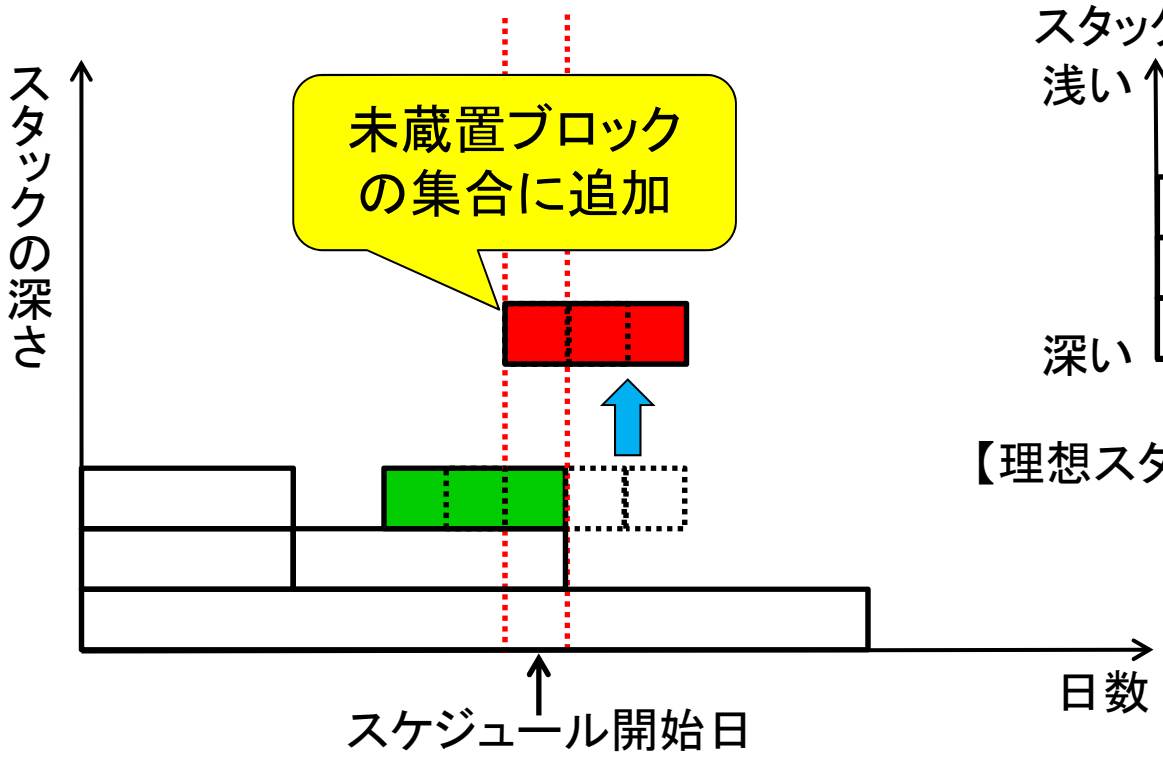


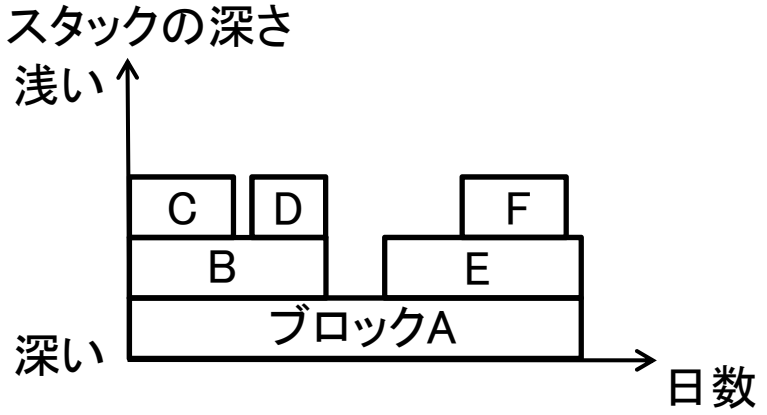
【理想スタック蔵置ルールに従う置き方】

- 理想スタック蔵置ルールを満たしていないブロックの蔵置期間を分割し、新たにブロックを生成

リスケジューリング

本研究



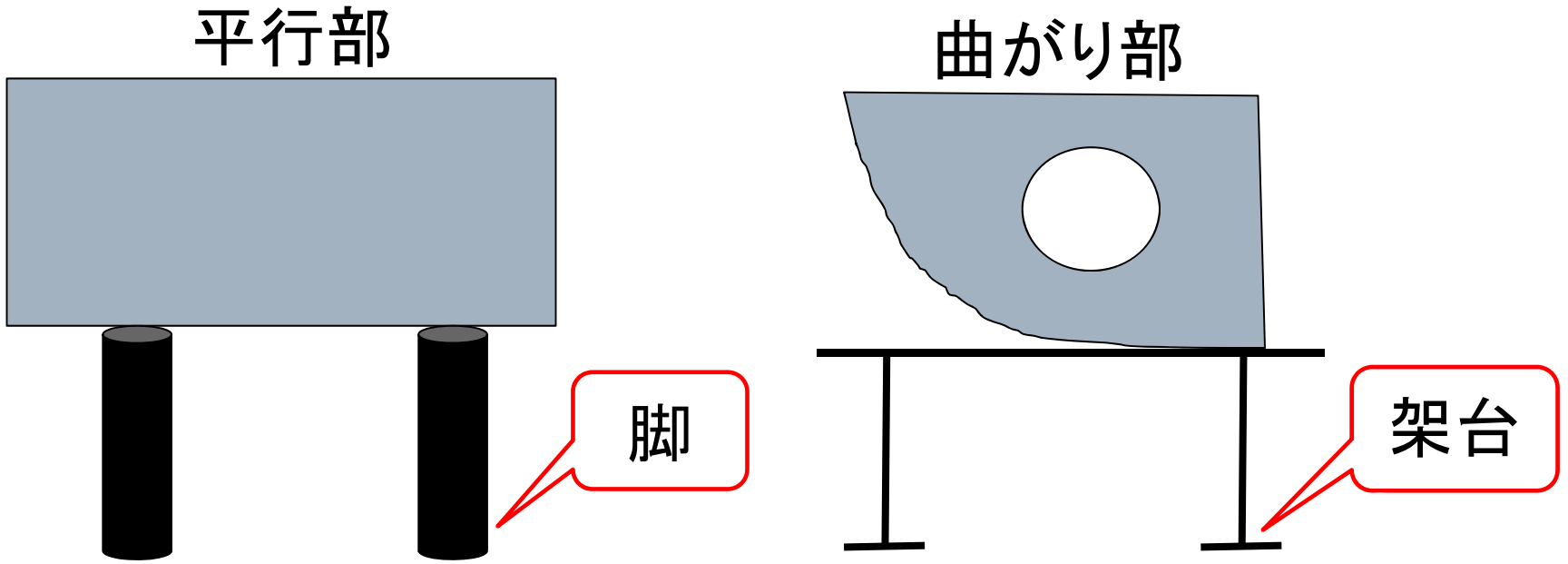


【理想スタック蔵置ルールに従う置き方】

- 生成したブロックを未蔵置ブロックの集合に追加し、この日からリスケジューリングを開始する

ブロックの属性情報の追加

本研究



- ブロックは平行部、曲がり部の２種類のブロックが存在する
- 置場も平行部、曲がり部それぞれ専用に別れている

→ ブロックを置くことのできるスタックが制限される

スケジュール結果の検証

本研究

分枝限定法1(枝300)を適用

	属性情報追加前 (初期配置5%)	属性情報追加後 (初期配置30%)
ストックヤードの空きスペース	6593	7091
のべ未蔵置ブロック数	163	354
のべ未蔵置ブロック期間	1050	1548
未蔵置ブロックの分割蔵置後		
ストックヤードの空きスペース	6153	6508
のべ未蔵置ブロック数	198	374
のべ未蔵置ブロック期間	796	1242

56%のべ未蔵置ブロック期間が増加

ブロックに平行部、曲がり部の区別をつけると結果が大幅に悪くなる

→ ブロックを置ける場所が限定されるとスケジュールが困難である

本発表の概要

1. 背景と目的
   - 造船所のストックヤードの現状と問題
2. アプローチ
   - 造船所の置場情報とブロック蔵置ルールの設定
   - ブロック割り当てアルゴリズム
3. 結果と考察
   - 従来手法との比較
   - サンプルデータの生成と検討
4. 実用化への取り組み
   - リスケジューリングと属性情報の追加
5. まとめと今後の課題

まとめと今後の課題

- 未蔵置ブロックの分割配置の改良、蔵置期間の長いブロックの初期配置を行った
- 置場からあふれるブロックが5%初期配置した時、従来研究より8.4%減少した
- リスケジューリングに対応した
- 置場とブロックの属性情報(平行部、曲がり部)を追加した
- 造船所の担当者の方から「実用レベルである」とコメントをいただいた

今後の課題

- 仮想データの実データに従った生成方法の検討

開発したスケジューラプログラム

・スケジューラプログラム
　Javaコンソールアプリ

・置き場（スタック）の情報とブロックの情報、
　およびスケジュール開始日のブロック蔵置状態を
　XMLファイルに記述　→　入力ファイル

・スケジュール結果および各種レポートをテキストファイルに出力