

【工学】

研究論文

RCPSP法によるストックヤードの工程計画

—ストック区画内のブロック蔵置場所の決定—

梶原 宏之*¹・木村 元*²・山田 拓史*³

Process Scheduling of a Shipbuilding Stockyard based on RCPSP Approach

--- How to determine the positions of blocks stored in stock areas ---

KAJIWARA Hiroyuki, KIMURA Hajime and YAMADA Hirofumi

Summary

First of all, the paper focuses the problem to construct a scheduling model of the stockyard in a shipyard in Japan, which should be verified by comparing the actual usages of facilities. Secondly it discusses the problem to determine the positions of blocks stored in stock areas. It is shown that these problems are formulated into RCPSP (Resource Constraint Project Scheduling Problem) and solved successfully using RCPSP-solver OptSeq developed by Prof. M. Kubo and Prof. K. Nonobe.

Keywords : (stockyard, process scheduling, block positioning, RCPSP, OptSeq)

1. 緒言

本研究で対象とする造船所では、建造ドック内に2隻を横に並べて1組とし、これを縦に2組並べ、この2組4隻を1サイクルとして建造する方式（ツインタンドム建造方式）を採用している。年間の建造隻数から建造ピッチが決まり、1サイクル4隻分の総組開始日が固定される。ドックでのブロック搭載は短期間に立ち上がるので、総組場に引き取られるブロックは予めストックしておかなければならず、約2隻分のストック区画が用意されている。また、当該造船所では、大組ブロックの組立棟はコンベアライン化されており、コンスタントな出棟が可能である。年間の建造隻数を増やすため建造ピッチを短縮する場合、総組場へのブロック引取りに対して、コンベアラインからのブロック出棟が適切に管理されていないと、ストック区画が足りなくなる恐れがある。

一方、建造隻数が増えるにつれて、ストックヤードの管理を預かる現場から、後工程（総組工程）で必要な順に出棟・ストックできていない、したがってブロックを運搬するキャリアの負荷（置換え、空走行、バッティングなど）が高く残業が多いなどの指摘がなされていた。実際にはブロックの種別に応じて、プラスト・塗装や艀装の作業の順番が変わり、またそれらの間に待機場所を確保する必要がある。これはブロック組立工程の計画問題にはなかった要素で、その待機期間はあらかじめ特定できず、計画後確定するという大変厄介な性質のものである。

このような計画問題は一般には、作業順番に関する制約（先行制約）と設備に関する種々の制約（資源制約）の下での資源制約付きスケジューリング問題、すなわちRCPSP (Resource Constraint Project Scheduling Problem)¹⁾として定式化できると考えられ、その特徴は大規模な組合せ

*¹ 長崎総合科学大学 新技術創成研究所 特命教授

*² 九州大学大学院 工学研究院 海洋システム工学部門

*³ ㈱大島造船所 情報システム部 技術システム課

2019年9月29日受付

2019年11月26日受理

最適化問題なることである。造船工程計画問題に対する伝統的なアプローチ²⁾は、PERT (Program Evaluation and Review Technique) を基礎とするので、先行制約については十分考慮できるが、資源制約については手作業による調整が必要となり、計画の手数の縮減（理想的には自動化）が期待されている。この目的に沿う従来研究として文献³⁾が発表されており、開発されたソルバーの有効性を示すために、ある造船所における定盤計画問題を検討している。このソルバーは OptSeq (LogOpt 社製) として商用化されている（文献⁴⁾参照）。

以上から、本研究の第1の目的は、組立出棟後から総組開始までの工程計画の検討を行い、ストックヤードにおけるリソースの使用状況の正確な把握を目指すことである。これによりストックヤードの工程モデルが得られたことになり、その効率的運用に資するばかりでなく、様々な応用が考えられる。その一つとして、本研究の第2の目的を、ブロックをあるストック区画に蔵置する場合の具体的な位置決めについて検討することとする。以上2つの目的を達成するために、RCPSP への定式化とそのソルバー OptSeq による求解からなるアプローチ（以下、RCPSP 法とよぶ）を採用する。

本論文の構成は次の通りである。第2章において問題設定のための準備を行い、第3章において RCPSP 法によるストックヤードの工程計画を述べ、第4章においてストック区画蔵置におけるブロック位置決めについて検討を行う。また、第5章において研究成果のまとめと今後の課題について述べる。

2. 問題設定のための準備⁵⁾

2.1 スtockヤードの設備とブロック・フロー

当該造船所のストックヤードには、次の4種類の設備が設置されている。

- ① ブラスト棟 (n_B 棟)
- ② 塗装棟 (n_P 棟)
- ③ 艦装棟 (n_F 棟)
- ④ スtock区画 (n_S 個)

ストックヤードにおける全設備の総数 $n_B+n_P+n_F+n_S$ は 157 個である。これらを組立棟 (8 棟) から出棟してくるブロックが総組引取までどのように通過すべきかを実績を踏まえて検討したところ、Table 1 に示す5種類のブロック・フロー1、2、3、4、5を抽出できた。

これから、出棟後から総組引取までの主な作業（アクティビティ）はブラストと塗装で、その前後に艦装が入り、また待機はタイミングによって4種類あることがわ

Table 1 Block Flows.

フ ロ ー	出 棟	待 機 1	艦 装 1	艦 装 2	待 機 2	ブ ラ ス ト	塗 装 3	待 機 3	艦 装 3	待 機 4	総 組 引 取
1	●	○	-	-	-	●	●	-	-	○	●
2	●	○	●	-	-	-	-	-	-	○	●
3	●	○	●	●	○	●	●	-	-	○	●
4	●	○	-	-	-	●	●	○	●	○	●
5	●	○	-	-	-	-	-	-	-	-	●

かる。ブラスト、塗装、艦装の作業期間は見積ることができるが、どれくらいの待機期間が必要となるかの見積りは、設備の空くタイミングが不明であることから事前にはできないことに留意する。

2.2 リソース選択肢と割付けたブロック数

本研究では、2組4隻を1サイクルとして、その前後サイクルを含めた3サイクル 12 隻の計画を行う。12 隻分のブロック総数は 642 個である。ブロックサイズの度数分布を Fig.1 に示す。ここでは、各ブロックを据置いたときの面積（長さ×幅）を、平均的なブロックの据置面積 320m²で規格化し、整数となるように 10 倍した値（以下、これをブロックサイズとよぶ）の頻度を求めている。

一方、全設備 157 個について搬入可能面積を、平均的なブロックの面積 320m²で規格化し、整数となるように 10 倍した値による設備サイズの度数分布を Fig.2 に示す。

Fig.1 と Fig.2 を比較すると、平均的なサイズ 10 のブロックを収容できる設備の数が断トツとなっているが（当該階級に○印を付す）、中にはかなり大きなサイズの設備もある。実際には、ストック区画については複数をまとめて運用し、2区画に3個のブロックを入れることもある。よって、たとえばストック区画の許容数量は1（各ストック区画には1度に1個のブロックしかストックしない）とするのではなく、収容面積による管理が必要となる。艦装棟についても同様である。このような事情を勘案し、全設備 157 個を統合整理した設備（以下、リソースとよぶ）の総数は 83 個となった。

一般に、各ブロックの種別とサイズ・重量によって、使用できるリソースが限られてくる。そこで、各ブロックについて、どの個別のリソースを使えるか検討し、リソース選択肢を定めた。その数を次に示す。

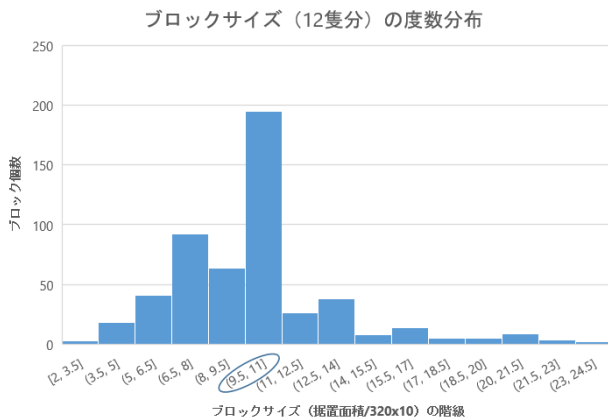


Fig.1 Histogram of block sizes.

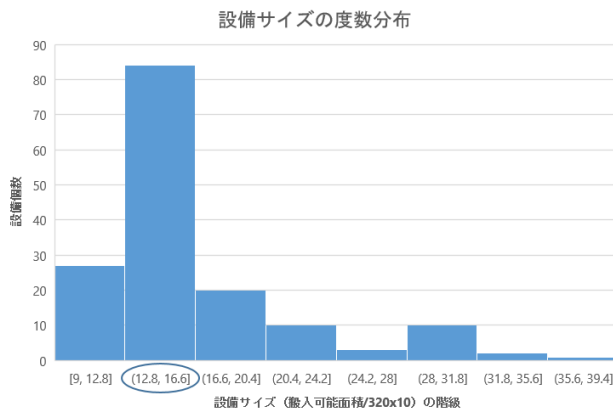


Fig.2 Histogram of facility sizes.

- ① ブラスト棟選択肢： 3 通り (B01~B03)
- ② 塗装棟選択肢： 10 通り (P01~P10)
- ③ 舩装棟選択肢： 5 通り (F01~F05)

④ ストック区画選択肢：11 通り (S01~S11)

ここで、ブラスト棟、塗装棟、舩装棟の選択肢は、主にブロックサイズから決定した。各選択肢の要素は 83 個のリソースから決められる。一方、ストック区画の選択肢については、待機後の次の作業の近くに置くという方針から決定した。このルールが遵守されると、キャリアによるブロックの運搬作業が効率化されるためである。

2.3 アクティビティの定義

Table 1 に示した 5 種類のブロック・フローについて、ブロック・フロー 1 では、

組立出棟⇒待機 1 ⇒ブラスト⇒塗装⇒待機 4 ⇒総組引取の順番で作業が実施されるので、次の情報をもつようにアクティビティの定義を行う。

- 1) 出棟：出棟日、後続作業
- 2) ブラスト：作業期間、ブラスト棟選択肢、後続作業
- 3) 塗装：作業期間、塗装棟選択肢、後続作業、先行作業
- 4) 引取：引取日

ここで、ブラスト前 (1 と 2 の間) と、引取前 (3 と 4 の間に) に待機の仮想アクティビティをプログラム内で自動的に定義する。したがって、全部で 6 個のアクティビティがスケジューリングされることになる。また、ブラストと塗装の間には待機を入れないので、プログラム内でその指示を行う。

他のブロック・フロー 2, 3, 4, 5 でも同様にしてアクティビティの定義を行う。それぞれ待機まで含めて 5 個、8 個、8 個、3 個のアクティビティがスケジューリングされる。

ID	アクティビティ名	後作業	期間	開始日	作業場所	待機場所	時数 1	時数 2	長さ	幅	高さ	
1	ship#_blk# C FLOW1 assembly	-2	14	35	A03	S03	[(2, 4):6, (4, 6):17, (6, 8):22, (8, 10):11]	[(6, 10):10, (10, 12):20, (12, 14):30]	28.6	18.6	4.3	FLOW1
2	ship#_blk# C FLOW1 blast	3	1		B01				28.6	18.6	4.3	FLOW1
3	ship#_blk# C FLOW1 painting	-4	10		P02	S03			28.6	18.6	4.3	FLOW1
4	ship#_blk# C FLOW1 preerection	0	1	78					28.6	18.6	4.3	FLOW1
13	ship#_blk# C FLOW5 assembly	-14	24	69	A07	S03	[(0, 2):12, (2, 4):18, (4, 6):24, (6, 8):6]	[(6, 10):12, (10, 12):12, (12, 14):17, (14, 16):17]	9.5	7.9	4.8	FLOW5
14	ship#_blk# C FLOW5 preerection	0	1	80					9.5	7.9	4.8	FLOW5
15	ship#_blk# C FLOW3 assembly	-16	16	41	A03	S06	[(2, 4):9, (4, 6):26, (6, 8):34, (8, 10):17]	[(6, 10):18, (10, 12):35, (12, 14):53, (14, 16):55]	32.1	19.1	3.7	FLOW3
16	ship#_blk# C FLOW3 fitting	-17	14		F02	S06			32.1	19.1	3.7	FLOW3
17	ship#_blk# C FLOW3 blast	18	1		B01				32.1	19.1	3.7	FLOW3
18	ship#_blk# C FLOW3 painting	-19	6		P04	S06			32.1	19.1	3.7	FLOW3
19	ship#_blk# C FLOW3 preerection	0	1	82					32.1	19.1	3.7	FLOW3
24	ship#_blk# C FLOW4 assembly	-25	18	19	A03	S06	[(2, 4):14, (4, 6):42, (6, 8):56, (8, 10):28]	[(6, 10):32, (10, 12):64, (12, 14):96, (14, 16):96]	26.4	21.6	6.2	FLOW4
25	ship#_blk# C FLOW4 blast	26	1		B02				26.4	21.6	6.2	FLOW4
26	ship#_blk# C FLOW4 painting	-27	12		P06	S06			26.4	21.6	6.2	FLOW4
27	ship#_blk# C FLOW4 fitting	-28	28		F01	S06			26.4	21.6	6.2	FLOW4
28	ship#_blk# C FLOW4 preerection	0	1	94					26.4	21.6	6.2	FLOW4
34	ship#_blk# C FLOW2 assembly	-35	14	49	A04	S08	[(0, 2):15, (2, 4):23, (4, 6):30, (6, 8):8]	[(6, 8):20, (8, 10):50, (10, 12):80, (12, 14):40]	32.9	21.6	3.9	FLOW2
35	ship#_blk# C FLOW2 fitting	-36	14		F04	S05			32.9	21.6	3.9	FLOW2
36	ship#_blk# C FLOW2 preerection	0	1	96					32.9	21.6	3.9	FLOW2
2525	ship#_blk# C FLOW1 assembly	-2526	8	303	A06	S11	[(0, 2):14, (2, 4):7, (4, 6):14]	[(2, 4):21, (4, 6):28, (6, 8):21]	22.3	14	3.2	FLOW1
2526	ship#_blk# C FLOW1 blast	2527	1		B01				22.3	14	3.2	FLOW1
2527	ship#_blk# C FLOW1 painting	-2528	6		P02	S11			22.3	14	3.2	FLOW1
2528	ship#_blk# C FLOW1 preerection	0	1	318					22.3	14	3.2	FLOW1

Fig.3 A part of data set prepared in the research.

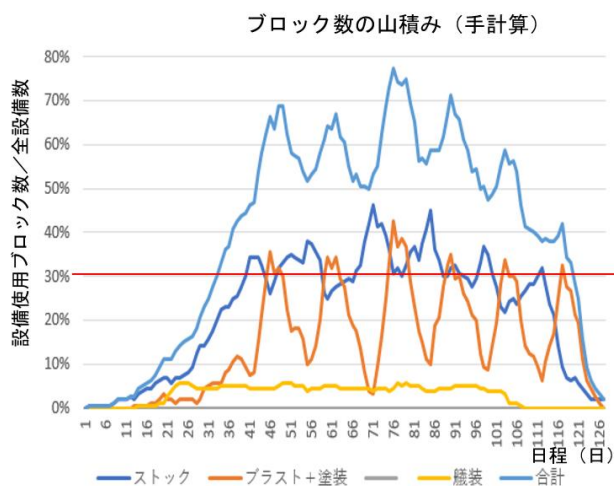


Fig.4 Used amounts of 4 facilities via hand calculation.

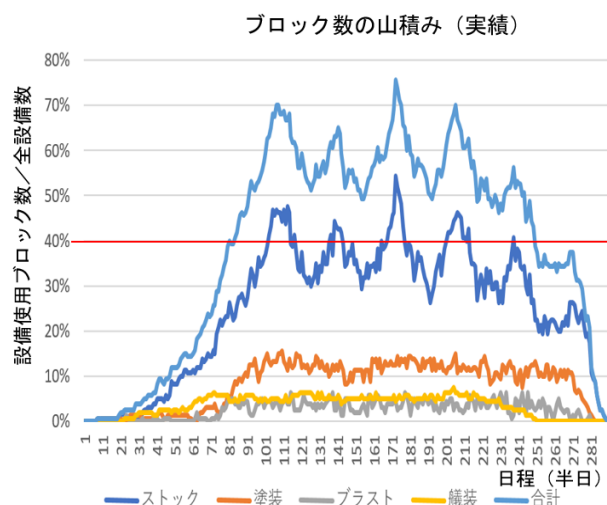


Fig.5 Actual uses of 4 facilities with a planning unit period of half day.

本研究で扱う 12 隻分のブロック 642 個に対してすべてのアクティビティを定義すると 2528 個、待機を表す仮想各ティビティまで含めるとアクティビティの総数は 3815 個となった。そのデータセットの一部を Fig.3 に示す。ここで、単位計画期間は半日としている。

2.4 ストック量の事前見積と実績データ

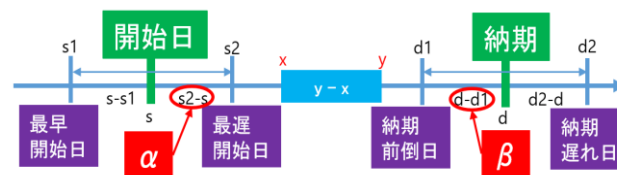
ストックヤードの工程計画の評価指標として、各設備を使用するブロック数の山積みがある。現行の工程計画に基づいて、各設備の使用数の山積みを行った結果を Fig.4 に示す。一方、各設備へのブロックの出入りを記録して山積みを行った結果を Fig.5 に示す。両者は設備総数 157 個に占める割合として表している。両者にはストックされるブロック数のレベルについて明らかな相違がみられ、現状の計画では、待機のタイミングと期間までは考慮できないので、正確な見積もりができていないことがわかる。

3. スtockヤードの工程計画⁵⁾

3.1 RCPSP 法による工程計画

本研究では、目的関数を納期遅れ総和にとって、これを最小化するようにアクティビティの開始日を決定するスケジューリング問題を RCPSP として定式化し、ソルバー OptSeq を用いて解いた。各アクティビティの先行制約はブロックフローから定まり、資源制約はリソース選択肢から規定され、基本的に面積管理に相当する。ただし、Fig.6 に示すような、開始遅れ（出棟遅れ）と納期前倒

（総組引取の前倒し）を許容するパラメータ α と β を設定した。 $\alpha=6$ と $\beta=0$ のとき、最大開始遅れ 6、開始遅れ総和 152、最大納期前倒 0 で解が求まった（単位計画期間は半日、計算時間 108.63 秒）。その結果は、日程計画（3815 個のアクティビティのガントチャート）とリソース使用状況（83 個のリソースの使用頻度など）の 2 種類に分かれて出力される。以下、これらについて説明する。

Fig.6 Moving start date backward with α and due date forward with β .

3.2 RCPSP 法による日程計画

日程計画の一例を、ブロック・フローごとに Fig.7 に示す。ここで、■の並びは作業の実施を、□の並びは待機を表す。

Fig.7 の 1 行目から 6 行目はブロック・フロー 1 の一例である。1 行目は当該ブロックの出棟日、6 行目が引取日を表す。どちらも指定日に固定された上で、それらの間にプラストと塗装のアクティビティが連続してスケジューリングされている（他のブロック・フローでも同様）。この場合プラスト前の待機 1 は不要となっているが、引取前の待機 4 の期間は自動的に決定されている。

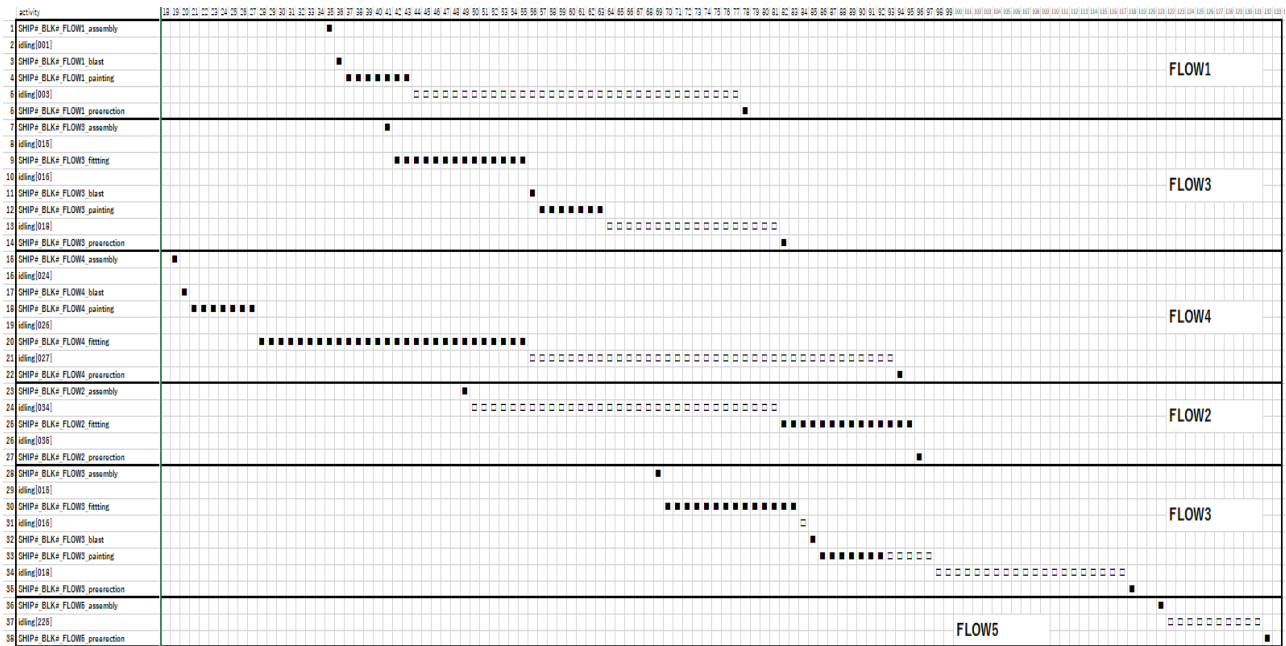


Fig.7 A part of RCPSP-based scheduling result.

Fig.7 の 23 行目から 27 行目はブロック・フロー 2 の一例である。指定された出棟日と引取日の間に、艀装 1 のアクティビティがスケジューリングされている。艀装 1 の前に待機 1 が置き場待機として確保され、艀装後は直ちに引取られている。

Fig.7 の 7 行目から 14 行目はブロック・フロー 3 の一例である。指定された出棟日と引取日の間に、艀装 1 とブラストと塗装のアクティビティがスケジューリングされている。引取前の待機 4 が置き場待機として確保されている。また Fig.7 の 28 行目から 35 行目もブロック・フロー 3 の一例である。ここでの特徴は待機 4 の一部が塗装終了後も塗装棟を使って行われていることである。本研究ではこれをその場待機と呼び、待機場所を確保する置き場待機と区別しており、このような設定ができることがソルバー OptSeq を用いる利点でもある。

Fig.7 の 15 行目から 22 行目はブロック・フロー 4 の一例である。指定された出棟日と引取日の間に、ブラストと塗装と艀装 3 のアクティビティがスケジューリングされている。艀装 3 の前後に置き場待機が確保されている。

Fig.7 の 36 行目から 38 行目はブロック・フロー 5 の一例である。指定された出棟日と引取日の間は、特に作業はなく置き場待機だけが確保されている。

3.3 RCPSP 法によるリソース使用状況

まず各リソースを使用するブロック数の山積みを行っ

た結果を Fig.8 に示す。ただし、ブロック数への換算は、各リソースを使用するブロックの占有面積を平均的なブロックの据置面積 320m² で規格化して行っている。各リソースの使用率の割合は、Fig.5 とほぼ同等のものが得られており、実績をかなり反映できているといえる。このことは、RCPSP 法によるストックヤードの工程計画が実用に資する可能性を示唆している。

次に、全リソース 83 個の使用状況を、艀装棟、ブラスト棟、塗装棟、ストック区画について、Fig.9 と Fig.10 に示す。比較的偏りなく使用されているといえるが、中には使用頻度の少ないリソースも見受けられる。

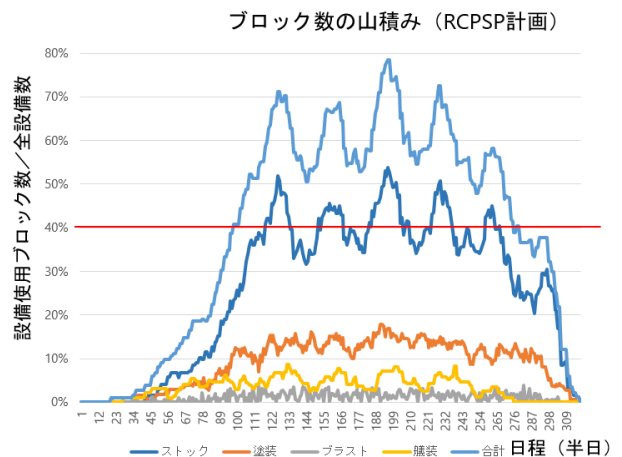


Fig.8 Used amounts of resources in RCPSP-based scheduling.

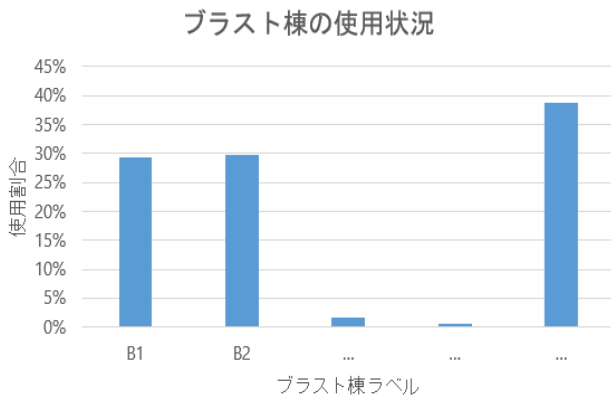


Fig.9a Resource uses of blast facilities in RCPSP-based scheduling.

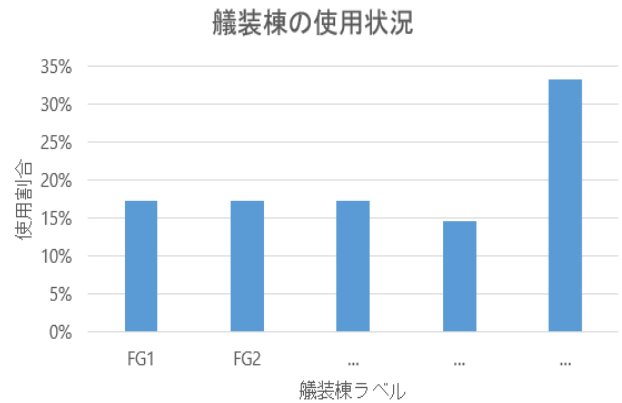


Fig.10a Resource uses of fitting facilities in RCPSP-based scheduling.

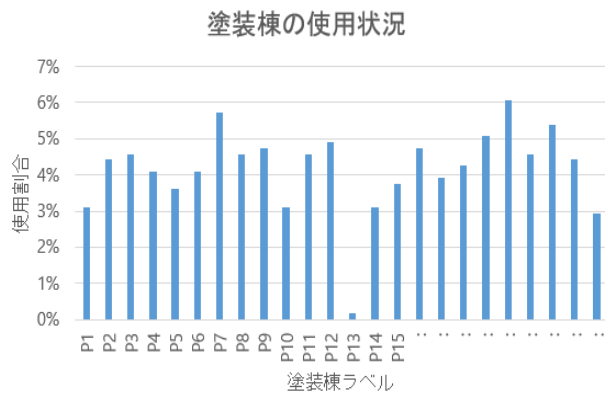


Fig.9b Resource uses of painting facilities in RCPSP-based scheduling.

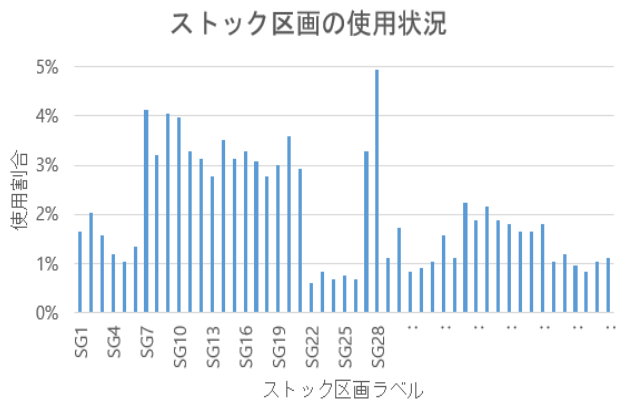


Fig.10b Resource uses of stock areas in RCPSP-based scheduling.

4. ストック区画蔵置の位置決め

前章までで、当該造船所のリソースの面積管理による日程計画が固まった。ただ、ストック区画へのブロック蔵置については、ブロックをどのストック区画におくかまではわかるが、どの位置におくかまではわからない。ただし、ここでいうストック区画は 2.1 節で示したリソースとしての区画である。その面積サイズ (平均的なブロックの面積 320m^2 で規格化し、整数となるように 10 倍した値) によるヒストグラムは Fig.11 となった。これによると大部分のストック区画は 1~2 個のブロック (標準サイズ 10) を収容できるが、中には 7~8 個のブロックを収容できるものもある。

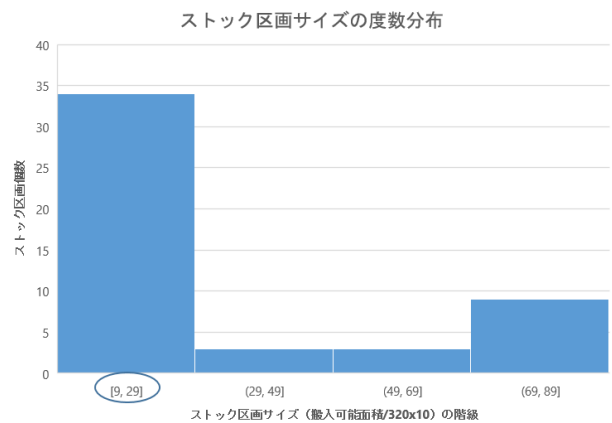


Fig.11 Histogram of stock area sizes.

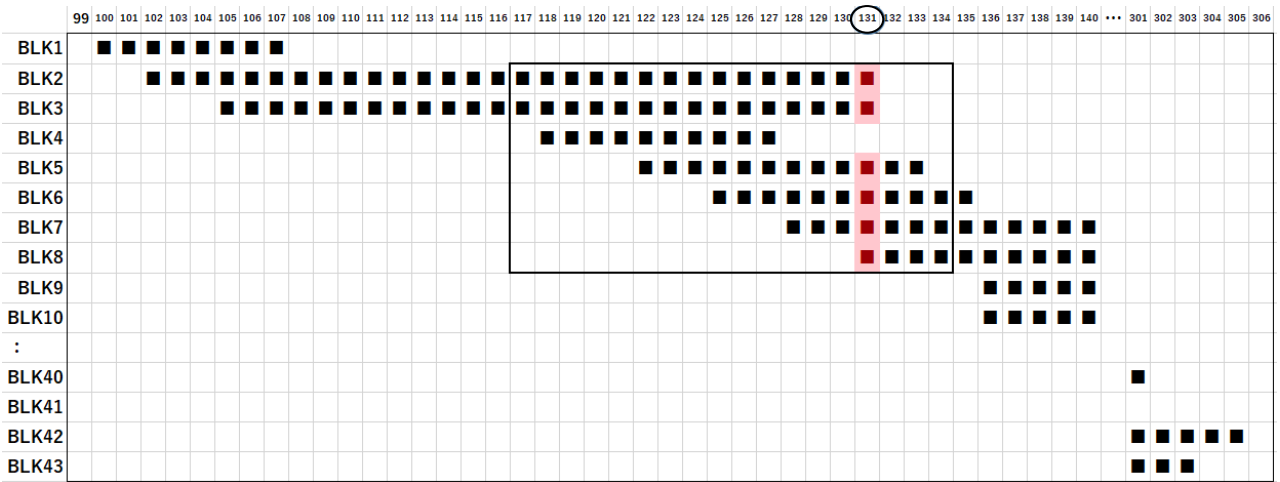


Fig.12 Scheduling result on a stock place.

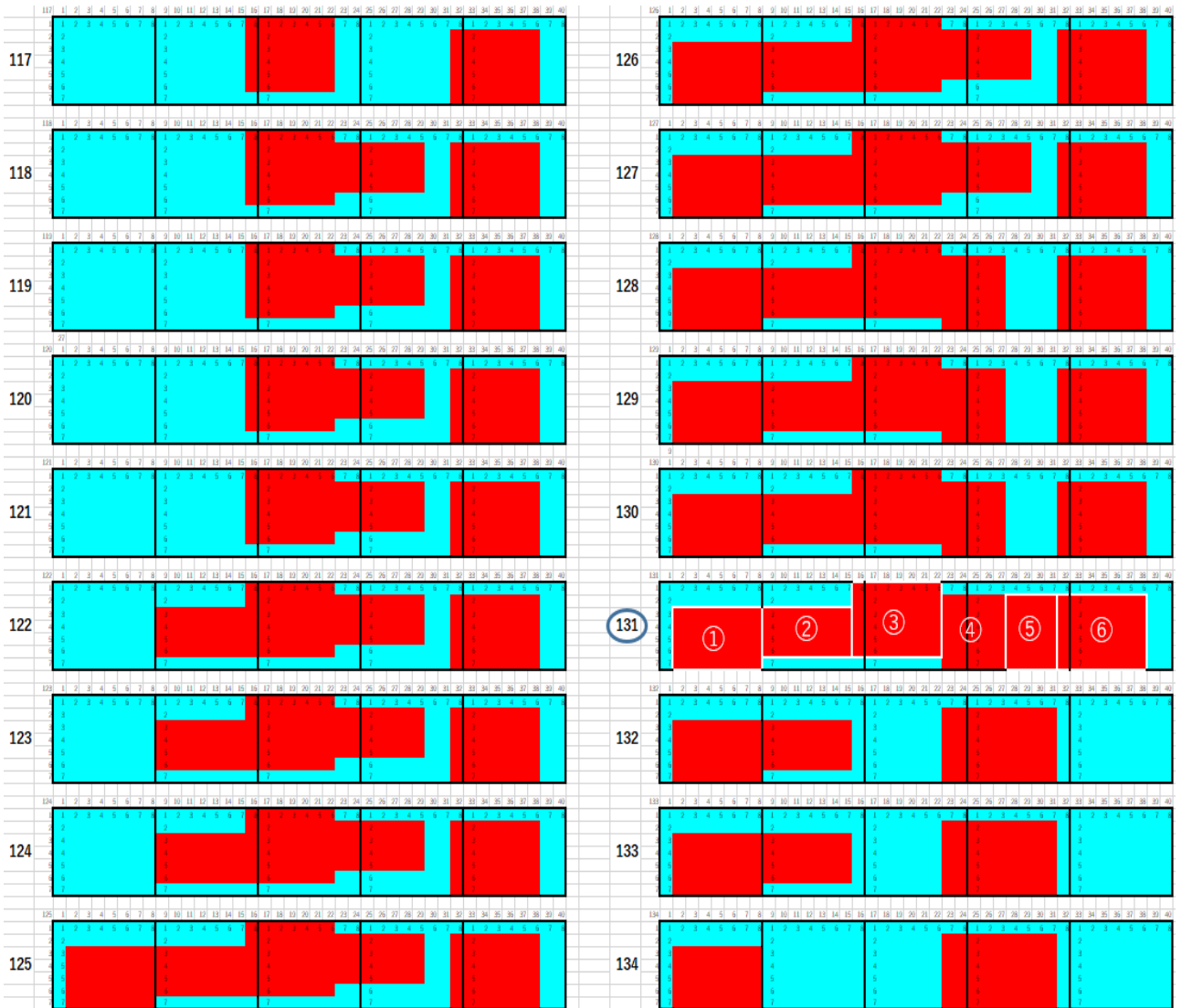


Fig.13 Block positioning corresponding to Fig.12.

Fig.12 は、本研究で求めた、あるストック区画(114m×20m)における 43 種類の待機アクティビティの日程計画である。開始日の早い順にソーティングしており、前章までの記法における待機を□ではなく、■で表している。計画期間 206 半日のうち、最大6個蔵置している日数は 23 半日、5 個蔵置している日数は 36 半日である。

さて、ストック区画への割付けをどう行うかという問題について考える。この問題は一般の造船所における定盤計画問題と類似しており、第 1 著者は文献⁶⁾において RCPSP 法による定盤計画へのアプローチを示している。これを Fig.12 の日程計画に適用した結果を Fig.13 に示す。ただし、ブロック蔵置の向きは同日に入れるブロック長さの和がストック区画の長さを超えないように決定している。Fig.12 で最初に最大6個蔵置しているのは 131 半日目であるが、Fig.13 の 131 半日目を見ると確かに6個のブロックが蔵置されている。この配置状況を見越し以前のブロックの位置決めを行うことは人手では容易ではなく、本手法の有用性を示すものと思われる。

5. 結 言

本研究の第 1 の目的であるストックヤードにおけるリソースの使用状況の正確な把握については、その実績 Fig.5 を再現する RCPSP 法による計画における Fig.8 を得た。これはストックヤードの工程モデルが得られたことを意味し、塗装棟などへの搬入、ストック区画への蔵置、キャリアによる運搬など、これまで現場に頼ってきた作業指示の原案を作成でき、ストックヤードの効率的運用に役立つと思われる、その予実管理においては実績と比較する計画を入手できることになる。また、文献⁶⁾では上流の組立工程の時数平準化問題の検討を行っている。さらに、仮線表の実施可能性、設備増強の妥当性などの検討にも役立つものと思われる。

本研究の第 2 の目的であるストック区画でのブロック位置決めについては、文献⁶⁾の手法を適用することができた。これは面積管理による日程計画でストック区画での待機期間と待機場所を決め、これをベースに位置管理を行うアプローチの有用性を示唆している。一般に面積管理でうまくいっても位置管理がうまくいくとは限らず、さらなら事例研究が必要であることは述べるまでもない。

当該造船所ではストック区画へのブロックの搬入出はクレーンではなくキャリアによる。そこでストック区画

のストック構造に留意しなければならない。これは可能な限り出入りに近い場所には相対的に早く移動するブロックを、奥にある場所には相対的に遅く移動するブロックを蔵置するのが望ましいことを意味する。したがって、同時期に2個以上蔵置する計画のストック区画については、このストック構造を考慮したブロックの位置決めを考える必要がある。

このような制約を最初から RCPSP として定式化することは容易ではない。これは通常の定盤計画のようにストック区画の置き方をすべて資源制約として定義することはできるが、ストック期間の長いものを奥に蔵置しようとしてもストック期間自体が予め分からないので先行関係を定義できないと考えられるからである。

文献⁷⁾ではストック開始日とストック期間がわかる場合のストック構造を考慮したストック区画への蔵置の問題を扱っているが、ブロックをどのストック区画におくかを一度リセットする必要があり、本研究との整合性について検討中である。

参考文献

- (1) Christian Artigues, Sophie Demassey, Emmanuel Néron: Resource-Constrained Project Scheduling: Models, Algorithms, Extensions and Applications, Wiley, 2008
- (2) 山崎真喜: 造船の計画管理, 成山堂書店, 1995
- (3) 野々部宏司, 茨木俊英: 資源制約付きスケジューリング問題の定式化と近似解法, 数理解析研究所講究録, vol.1120, pp.88-97, 1999
- (4) 久保幹雄, 小林和博, 斉藤努, 並木誠, 橋本英樹: Python 言語によるビジネスアナリティクス, 近代科学社, 2016
- (5) 梶原宏之, 木村元, 石川一郎, 山田拓郎, 嵩下雄介, 中島省吾: スtockヤード工程の RCPSP 計画に関する研究, 日本船舶海洋工学会講演論文集 (投稿中)
- (6) 梶原宏之: 造船工程計画のための新技術 RCPSP 法について, 長崎総合科学大学新技術創成研究所報第 14 巻, pp.39-44, 2019
- (7) 木村元, 酒井英輔: 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, 日本船舶海洋工学会講演論文集, vol.14, pp.431-434, 2012