

Production Scheduling in Factories using a Mathematical Optimization Approach

Ikuto Nakatsukasa*, Yasuaki Ito[†], Koji Nakano[†], and Victor Parque[†]

*School of Informatics and Data Science, Hiroshima University

Kagamiyama 1-4-1, Higashi-Hiroshima, 739-8527, JAPAN

[†]Graduate School of Advanced Science and Engineering, Hiroshima University

Kagamiyama 1-4-1, Higashi-Hiroshima, 739-8527, JAPAN

Abstract—The scheduling of production in a factory is a complex process that is typically carried out over an extended period by personnel possessing the necessary skills and experience. This process requires careful consideration of various factors, including daily production variations, machine utilization rates, and the management of multiple product lines. While manual scheduling may be effective in certain situations, it can lead to inefficiencies, such as lost time and increased reliance on specific personnel, which may restrict the flexibility of factory operations. In this study, we developed a mathematical optimization model to automate the scheduling process. The model allocates tasks to available time slots while accounting for constraints such as production capacity and deadlines. The results demonstrate that the model generates schedules comparable to those created by experienced workers, significantly reducing the time required to complete tasks.

Index Terms—mathematical optimization, production scheduling, factory operation, task allocation

I. はじめに

工場における生産スケジュールは、日々の生産状況や、複数の製品を扱うといった要因のため、一般的なスケジューリングより複雑なものとなっている。特に複数の設備や機材を使用するような作業の場合、製品に応じた適切な設備等の割り当てや、処理条件の設定といった、複雑な制約を考慮する必要がある。したがって生産スケジュールは、一般に知識や経験が豊富な担当者が長く時間をかけて作成することが多い。これは、作成に長時間を費やすことによる時間的ロスや、担当者による属人化の進行、より効率の良いスケジュールの見落としなど、多くの問題がある。

本研究では、実際に複雑な生産スケジュールを必要とする工場において、数理最適化を用いた生産スケジュールの作成、最適化を行うモデルを作成する。

II. 数理最適化

数理最適化 (Mathematical Optimization) とは、いくつかの制約関数のもとで、ある目的関数を最大化または最小化するような変数を解く手法のことである。制約や目的関数の設定次第で、非常に多くの場面で活用が可能で、効率的な資源の割り当て、コスト削減、スケジュール作成などで貢献している。属人化の拡大を招き、時間的コストのかかる作業であるスケジューリングを自動化することで、多くのメリットを得ることができる。実際の例として、運搬車の移動時間及び移動費用が与えられた時の最小費用のスケジュールを求める運搬車スケジューリング問題を数理最適化で解くことで、必要な車両数の削減に成功している [1]。他に、混合サービス

モードのドックエリアにおけるクロスドッキングのトラックスケジューリング問題に対し、到着から出発までの時間を最小化することを実現している [2]。最近では、修理に必要なリソースや時間、設備の割り当てを考慮した上で車両の待機時間を削減するスケジューリングを解いていた例がある [3]。その他にも、配送量や配送拠点、車両の割り当てを考慮した物流の最適化 [4]、大規模な望遠鏡ネットワークのスケジューリング [5] など、多くの分野で利用されている。

III. スケジューリング問題

スケジューリング問題は、人員や時間、設備といった限られたリソースを効率よくタスクに割り振る問題である。一般的なスケジューリング (シフト等) から、生産スケジュール、物流、プロジェクト管理に至るまで、多くの場面で応用可能な問題である。その内容や特徴に応じて幾つかのタイプに分類され、例えば以下のものがある。

- ジョブショップスケジューリング問題 (Job Shop Scheduling Problem)
それぞれ異なる順序 (設備の通過順等) のあるタスクに対するスケジューリング問題。
- フロースショップスケジューリング問題 (Flow Shop Scheduling Problem)
同じ順序の作業を行うタスクに対するスケジューリング問題。
- オープンショップスケジューリング問題 (Open Shop Scheduling Problem)
順序に制約がないタスクに対するスケジューリング問題。

IV. 扱う問題

今回扱うのは、加工業を行う工場におけるスケジューリングで、できるだけ短い時間で全ての注文の処理を完了するスケジュールを作成することを目的とする。

各注文 i には次の情報が含まれている。

- 製品 I : 処理する製品の種類
- 総タイムスロット数 T_i : 処理の完了に必要なタイムスロット数
- 優先度 P_i : 自然数で、値が大きい注文ほどできるだけ早く処理されるようにする

また、各製品 I には次の情報が含まれている。

- 処理条件 C_I : 処理時に、設備に設定するパラメータ

- 処理可能設備 F_I : 処理可能な設備の番号 設備は α, β の2機とする $F = \{\alpha, \beta\} F_I \in F$
- グループ G_I : 類似品ごとに分けられたグループの番号

さらに, 制約を設定する上で何も配置されないタイムスロットが発生する場合がある. これは, 配置可能な注文がない場合挿入されるもので, 以降空タイムスロットと呼ぶ.

これらを踏まえて, 次の条件を全てを満足するようなスケジュールを求める.

- 1) 同じ注文 I は, 同じ設備 F に, 全て連続して配置すること.
- 2) 全ての注文 i がいずれかのタイムスロットに, 総タイムスロット数 T_i 分配置されること.
- 3) 連続する2つのタイムスロット間の処理条件 C が大きく離れるときは, 設備 F に処理条件 C を反映させるための時間が必要なため, その前後の処理条件 C の値ごとに決まった個数の空タイムスロットを挟むこと. 必要な空タイムスロット数は表 I に示す.

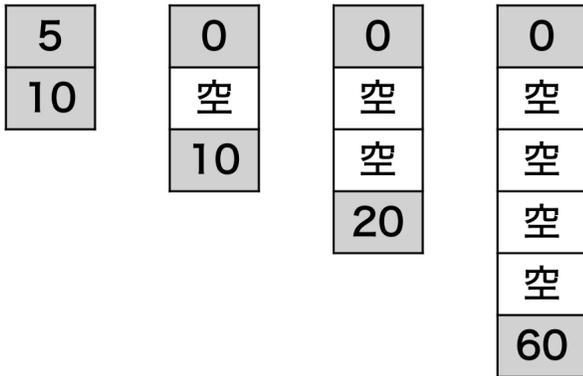


Fig. 1. 必要な空タイムスロットのイメージ (数字は処理条件)

TABLE I
 必要な空タイムスロット数

		次の処理条件													
		0	5	10	15	20	25	30	35	40	45	50	55	60	
直前の処理条件	0	0	1	1	1	2	2	2	2	4	4	4	4	4	
	5	1	0	0	0	1	1	1	2	2	2	4	4	4	
	10	1	0	0	1	2	2	2	2	2	2	2	2	2	
	15	1	0	0	0	1	1	2	2	2	2	2	2	2	
	20	1	1	1	0	0	0	1	1	2	2	2	2	2	
	25	1	1	1	1	0	0	0	1	2	2	2	2	2	
	30	2	1	1	1	1	0	0	0	1	2	2	2	2	
	35	2	1	1	1	1	0	0	0	1	1	2	2	2	
	40	2	2	1	1	1	1	0	0	0	1	1	2	2	
	45	2	2	1	1	1	1	1	0	0	0	1	2	2	
	50	2	2	2	1	1	1	1	1	1	0	0	0	0	
	55	2	2	2	2	2	1	1	1	1	0	0	0	0	
	60	2	2	2	2	2	2	1	1	1	1	0	0	0	

- 4) 異部品混入を防ぐため, グループ G が同じ他部品の処理が連続しない様にする.

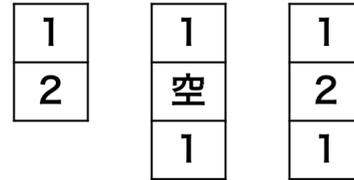


Fig. 2. 必要な空タイムスロットのイメージ (数字はグループ)

- 5) 処理可能設備 F のいずれかのタイムスロットに割り振ること.
- 6) 優先度 P_I が高い注文 I をできるだけ早いタイムスロットに配置すること.

V. 数理モデルの作成

本節では, 前節で示したスケジューリング問題を解くための数理モデルの実装を紹介する.

はじめに, 設定した変数について紹介する. 以下, 特に指定がない場合 i は注文番号, j は設備番号, k はタイムスロット番号とする.

まず, タイムスロットを表す変数として, 配置されていたら1, そうでなければ0となるバイナリ変数 $x_{i,j,k}$ を定義した. タイムスロットは, 各設備ごとに24時間を1タイムスロットあたりの処理時間 n 分で分割した数 $\frac{1440}{n}$ 個を設置した. 注文番号 i の注文が, 設備 j で, k 番目のタイムスロットに配置されている場合, $x_{i,j,k} = 1$ となる.

次に, ある注文 i がある設備 j で, 連続するタイムスロットに配置されている時, その先頭タイムスロット k だけ1となるようなバイナリ変数 $y_{i,j,k}$ を定義した. これは1)の制約を実現する上で必要な変数で, x と同じサイズの3次元変数とする. 総タイムスロット数 $T_i = 3$ の注文 i について, $x_{i,j,k}$ に対する $y_{i,j,k}$ の値の関係を表 II に示した.

TABLE II
 x と y の値の関係

k	$x_{i,j,k}$	$y_{i,j,k}$
0	0	0
1	1	1
2	1	0
3	1	0
4	0	0

次に, 目的関数を紹介する. 今回は, 全ての注文 i ができるだけ早く処理されるスケジュールを求めるために, 全ての注文 i ができるだけ早いタイムスロットに配置されることを目的とする. また同時に, 優先度 P_i の高い注文が低い注文より早いタイムスロットに配置されることも目指す. 設定した目的関数は以下ようになった.

$$\text{objective} = \text{Minimize} \sum_i \sum_j \sum_k P_i \cdot k^2 \cdot x_{i,j,k}$$

最後に、各制約を紹介する。まず、全てのタイムスロットにおいて、配置可能な注文 i を 1 つまでにするための制約を設定した。

$$\forall i \quad \sum_j \sum_k x_{i,j,k} \leq 1$$

$$\forall i \quad \sum_j \sum_k y_{i,j,k} \leq 1$$

次に、 y が先頭タイムスロットのみ 1 とするために、以下の 3 つの制約を設定した。

$$y_{i,j,k} \geq x_{i,j,k} - x_{i,j,k-1}$$

$$y_{i,j,k} \geq x_{i,j,k}$$

$$y_{i,j,k} \leq 1 - x_{i,j,k-1}$$

1) の条件は、 y を用いた制約の追加で満たすことができる。 y は、注文の先頭タイムスロットのみ 1 になる変数である。したがって、注文 i の全てのタイムスロットにおける y の合計が 1 であれば開始点が 1 つとなるため、 i のタイムスロットがどこかのタイムスロットに配置されると、必ず残りの全てのタイムスロットも連続して配置されることになる。この制約を数式で表すと、以下のようになる。

$$\forall i \quad \sum_j \sum_k y_{i,j,k} = 1$$

2) の条件は、各注文 i について、全てのタイムスロットにおける $x_{i,j,k}$ の合計が必要タイムスロット数 T_i となるようにする制約の追加で満たすことができる。この時、 j を製品 I_i の処理可能設備 F_{I_i} に限ることで、同時に 5) の制約も実現できる。この制約を数式で表すと、以下のようになる。

$$\sum_i \sum_{j \in F_{I_i}} \sum_k x_{i,j,k} = T_i$$

3) の条件は、ある注文 i の製品 I_i と、 i 以外のある注文 i' の製品 $I_{i'}$ の処理条件から、間に挟む必要のあるタイムスロット数 δ を求め、そのタイムスロット数分の空タイムスロットを挿入する制約の追加で満たすことができる。この制約を数式で表すと、以下のようになる。

$$\forall i, i' \quad \text{s.t. } i \neq i' \quad \sum_j \sum_k \sum_{l=k+1}^{k+1+\delta} (x_{i,j,k} + x_{i',j,l}) \leq 1$$

この制約により、注文 i が設置されているタイムスロットの後に、注文 i' のタイムスロットを設置するためには、 δ 個のタイムスロットを開けることが強制される。

4) の条件は、ある注文 i の製品 I_i のグループ G_I と、 i 以外のある注文 i' の製品 $I_{i'}$ のグループ $G_{I'}$ が同じ場合、連続しないようする制約の追加で満たすことができる。この制約を数式で表すと、以下のようになる。

$$\forall i, i' \quad \text{if } G_{I_i} = G_{I_{i'}} \quad \text{s.t. } i \neq i' \quad \sum_j \sum_k (x_{i,j,k} + x_{i',j,k+1}) \leq 1$$

VI. 性能評価

本項では、作成したモデルで実際のスケジュールを最適化した結果と、作業員が手作業で作成したスケジュールを比較することで、モデルの性能評価を行う。なお、最適化に当たってソルバーは Gurobi (バージョン 11.0.3) を使用した。実行環境は、2.3GHz クアッドコア Intel Core i5, RAM 16GB のノートパソコンである。

今回は実際に用いられた注文データ 5 例についてスケジュール作成を行い、結果の比較を行った。各場合におけるスケジュールの、各値の比較は表 V に示した。

以下に、表 V における、注文数 17 のスケジュールの例を具体的に示す。この例において扱う製品を表 III に、注文を表 IV に示す。優先度は作業員が入力したものを使用した。

TABLE III
製品一覧

製品 I	処理条件 C_I	処理可能設備 F_I	グループ G_I
A	50	α	0
B	60	α	1
C	60	α	1
D	60	α	1
E	60	β	1
F	50	β	2
G	60	α, β	2
H	60	β	0
I	60	β	1
J	50	β	0
K	40	β	0
L	50	β	2
M	15	α, β	1

TABLE IV
ある日の注文一覧

注文 i	製品 I_i	総タイムスロット数 T_i	優先度 P_i
1	A	65	99
2	B	109	88
3	A	20	1
4	C	86	99
5	A	20	1
6	D	68	77
7	D	55	1
8	E	14	100
9	F	28	99
10	G	59	50
11	H	9	99
12	I	11	99
13	J	14	99
14	E	9	99
15	K	72	50
16	L	63	88
17	M	60	88

次に、作業員が手作業で作成したスケジュールを表 VI に、本モデルで作成したスケジュールを表 VII に示す。ただし、注文 i および製品 I_i 、優先度 P_i がハイフンとなっている行は、空タイムスロットである。結果を出力するまでにかかった時間は約 430 秒で、最適解が得られた。想定した通り、制約を満たしていることも確認できた。

まず、スケジュールに含まれる総タイムスロット数によりスケジュールの最適性を比較する。表 VI と表 VII を比較すると、両設備ともに作業員作成のスケジュールの方が総タイ

ムスロット数は少なくなっている。これは本モデルが、優先度 P による注文の上下関係を厳密に遵守したことから生じた空タイムスロットの挿入によるものである。なお、両者の総タイムスロット数の差は合計で3つにとどまっており、合計の作業時間はほぼ同等であると言える。また、比較した5例のうち3例においては本モデル作成のスケジュールの方が総タイムスロット数が少なくなっている。このことから、総タイムスロット数がほぼ同等かつ優先度 P を厳密に遵守した結果を出力できており、本モデル作成のスケジュールが、作業員作成のスケジュールと比較してより最適な結果を出力しているといえる。

最後に、作成時間を比較する。今回の注文データにおける作業員作成のスケジュールは、作成に約60分を要した。これは注文数や設備数によってさらに伸びる可能性がある上、知識や経験が豊富な作業員の時間を拘束することになる。一方で本モデル作成のスケジュールの作成は7分程度で、作成に作業員の知識や経験を要さない。

以上より、本モデルは作業員作成時と比べて、より効率的の良いスケジュールを、より短い時間で作成することができた。

VII. まとめ

本研究では、数理最適化を用いて工場におけるスケジュール作成を行う数理モデルを作成した。いくつかの制約の元で注文をタイムスロットに割り当てていき、全ての注文の処理ができるだけ早く終わるようなスケジュールを出力する。結果として、作業員が作成するものと同等以上のスケジュールを、より短い時間で作成することができるモデルが実現できた。

REFERENCES

- [1] Z. Lan, S. He, R. Song, and S. Hao, “Optimizing vehicle scheduling based on variable timetable by benders-and-price approach,” *Journal of advanced transportation*, vol. 2019, no. 1, p. 2781590, 2019.
- [2] G. Correa Issi, R. Linfati, and J. W. Escobar, “Mathematical optimization model for truck scheduling in a distribution center with a mixed service-mode dock area,” *Journal of Advanced Transportation*, vol. 2020, no. 1, p. 8813372, 2020.
- [3] F. Pilar, E. Costa e Silva, and A. Borges, “Optimizing vehicle repairs scheduling using mixed integer linear programming: A case study in the portuguese automobile sector,” *Mathematics*, vol. 11, no. 11, p. 2575, 2023.
- [4] R. Silva-Lugo, R. Kozma, and L. Werbos, “Optimization of scheduling for larg-scale logistics tasks,” in *5th multidisciplinary international scheduling conference*, 2011.
- [5] S. Lampoudi, E. Saunders, and J. Eastman, “An integer linear programming solution to the telescope network scheduling problem,” *arXiv preprint arXiv:1503.07170*, 2015.

TABLE V
 スケジュールの比較

注文数	作業員				本モデル			
	目的関数値	終了タイムスロット		総タイムスロット数	目的関数値	終了タイムスロット		総タイムスロット数
		設備 α	設備 β			設備 α	設備 β	
17	2285446288	422	340	764	1771310676	423	342	767
19	17670916	222	207	431	11008071	210	218	430
15	20745914	183	215	400	9269872	216	189	407
16	17069636	211	213	426	9678111	248	175	425
12	16394874	183	219	407	8832557	189	219	410

TABLE VI
 作業員作成のスケジュール

設備 α					設備 β				
使用タイムスロット	注文 i	製品 I_i	タイムスロット数	優先度 P_i	使用タイムスロット	注文 i	製品 I_i	タイムスロット数	優先度 P_i
0-64	1	A	65	99	0-13	8	E	14	100
65-173	2	B	109	88	14-41	9	F	28	99
174-193	3	A	20	0	42-100	10	G	59	50
194-279	4	C	86	99	101-109	11	H	9	99
280-299	5	A	20	1	110-120	12	I	11	99
300-367	6	D	68	77	121-134	13	J	14	99
368-422	7	D	55	1	135-143	14	E	9	99
					144	-	-	1	-
					145-216	15	K	72	50
					217-279	16	L	63	88
					280	-	-	1	-
					281-340	17	M	60	88

TABLE VII
 本モデル作成のスケジュール

設備 α					設備 β				
使用タイムスロット	注文 i	製品 I_i	タイムスロット数	優先度 P_i	使用タイムスロット	注文 i	製品 I_i	タイムスロット数	優先度 P_i
0-85	4	C	86	99	0-13	8	E	14	100
86-150	1	A	65	99	14-41	9	F	28	99
151-259	2	B	109	88	42-52	12	I	11	99
260	-	-	1	-	53-66	13	J	14	99
261-328	6	D	68	77	67-75	14	E	9	99
329-348	3	A	20	1	76-84	11	H	9	99
349-403	7	D	55	1	85-147	16	L	63	88
404-423	5	A	20	1	148	-	-	1	-
					149-208	17	M	60	88
					209-210	-	-	2	-
					211-282	15	K	72	50
					283	-	-	1	-
					284-342	10	G	59	50