

生産現場の運用自動化を支える システム制御計画作成技術

中井敦子*
Atsuko Nakai
矢口喬脩*
Takanobu Yaguchi
八田夏美*
Natsumi Hatta

海江田広和*
Hirokazu Kaieda
中川晃一*
Koichi Nakagawa

*先端技術総合研究所

*Making Control Plans to Automate System Operations in
Manufacturing Site*

要 旨

生産現場では、労働力不足を補うための運用自動化ニーズが高まっている。生産現場での生産スケジュールは現場監督者が作成することが多いが、非常に労力がかかる作業になっていた。この課題を解決するために、現場作業者と生産設備が混在して作業する生産現場で生産能力が低下した際に、現場作業者と生産設備各々についてあらかじめ定義したプロフィール情報を用いて最もコスト効率の高い対策を自動立案し、適用する手法を開発した。また、生産設備での非稼働時間を最小化する生産スケジュールを高速に作成するために、整数最適化問題として定式化し処理する部分とルールベースで処理する部分を組み合わせた最適化手法を開発した。

1. ま え が き

近年、日本の生産現場では、生産年齢人口の減少や消費者ニーズの多様化による多品種少量生産への転換等、生産技術の革新が求められている⁽¹⁾。特に、食品工場や物流現場等では、夜間や休日の操業が不可欠であり、労働力の確保を更に困難なものにしている。労働力不足を補うために、生産設備の自動化が進められており、現場作業者の省人化による労働力の安定確保や作業負担軽減による生産効率の向上が見込まれる一方、生産設備と現場作業者の性能差が大きくなって、製造ラインの性能を保つことが困難になるという課題が生じている。また、生産の効率化を狙って、現場監督者が勘と経験によって生産スケジュールを作成しているが、これが現場にとって大きな負担になっていた。これらの課題を解決するために、現場作業者と生産設備が混在する生産現場で生産能力低下時に最適な対策を自動立案・適用する手法を考案した。具体的には、現場作業者と生産設備それぞれについてあらかじめ定義したプロフィール情報を活用し、最もコスト効率の高い対応策を自動的に導出する方式を開発した。また、生産設備での非稼働時間を最小化する生産スケジュールを高速に作成するために、考慮すべき多数のパラメーターを整数最適化問題として定式化し処理する部分とルールベースで処理する部分に分けて組み合わせることで、数十秒以内での生産スケジュール作成を実現した最適化手法を開発した。生産現場が要求する制約条件のうち、変動性の高い制約条件を整数最適化問題として定式化し、ソルバーを用いて解を求める。次に、生産現場に汎用的な制約条件をルールベースのアルゴリズム処理としてソルバーで求めた解に適用することで、生産スケジュールを作成する。

2. 生産設備と現場作業者のプロフィール情報を用いた生産システム運用制御方式

この章では、生産設備と現場作業者それぞれについて定義したプロフィール情報を用いて自動的に制御指令を作成する方式について述べる。提案方式は、ロボットや自動搬送機、制御装置等で構成される生産設備のプロファイル情報と、ステータス値やセンサー値、操作履歴等を含む作業者のプロフィール情報とを逐次的に収集・分析し、過去に同様の作業を実施した際のプロファイル情報と比較する。場の状況に応じて、生産設備や現場作業者のプロフィール情報をどのように変化させる又は変化させないことで所望の性能に到達するかをシミュレーションする。そして、所望の性能を実現するための制御内容を決定し、作業者と生産設備に対して配信することを目的としている。

プロフィール情報とは、生産設備固有の情報又は現場作業者固有の情報である属性情報と、生産システムの中で生産設備又は現場作業者が発揮する作業能力に関する情報である性能指標情報とで構成される。すなわち、プロフィール情報は、現場作業者と生産設備による総合的な作業能力を示す情報と定義する。生産設備としてロボットを例に具体的なプロフィール情報項目例を示すと、属性情報としてはカタログスペック値のほかに、“設置ライン名”などのロボットの運用に関わる情報がある。また、性能指標情報としては、作業対象物ごと又は作業条件ごとに算出される“作業速度”又は“作業不良

発生率”などの実稼働データから算出される情報を含む。現場作業員に対応するプロファイル情報項目例を示すと、属性情報としては人事情報や従事する業務内容に関わる情報などがある。また、性能指標情報としては、生産設備の場合と同様に、作業内容ごと又は作業条件ごとに算出される“作業速度”又は“作業不良発生率”などの実稼働データから算出される情報を含む。プロファイル情報は、工場種別や導入設備の違い等によって異なる部分もあるが、最低限の標準化を行う必要がある。そのため、プロファイル情報のひな型として、OPC UA^(注1)の情報モデルやコンパニオン仕様を活用することとした。例えば、ロボットに対してはOPC UA for Roboticsが策定されている。これらの情報モデルを基に、プロファイル情報に関連する項目を追加・拡張させることで、プロファイル情報の標準化を実現するとともに、プロファイル情報の収集や評価にかかるコストを低減できる。

この方式は、工場内に設置されたエッジプラットフォーム端末を中核にして動作するように設計した。図1にシステム構成を示す。エッジプラットフォーム端末内には、生産現場のデータ収集・加工・配信等を汎用的に行うために、一般社団法人 Edgexcrossコンソーシアムが提供するエッジコンピューティング領域のオープンなソフトウェアプラットフォームであるEdgexcross^{(注2)(2)}を採用した。今回は、OPC UA対応の生産設備や他システムとEdgexcrossの間でプロファイル情報を共有し、OPC UAのAPI(Application Programming Interface)を用いて互いに配信する仕組みとした⁽³⁾。

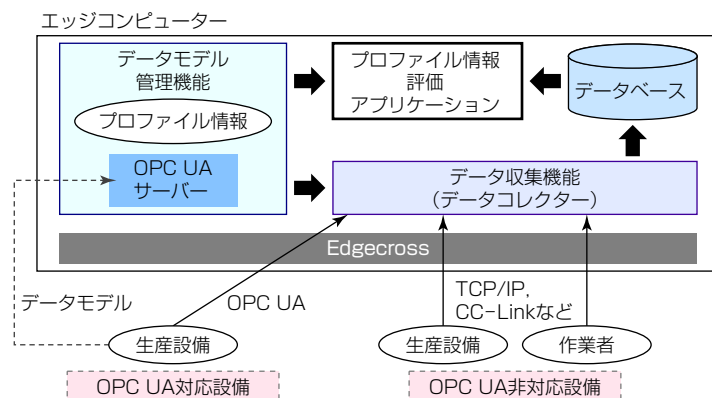


図1-システム構成

今回、エッジプラットフォーム端末内部にプロファイル情報評価アプリケーションを作成した。このアプリケーションは、Edgexcrossで収集したプロファイル情報を逐次評価し、生産システムの性能低下が見込まれる場合に、あらかじめ管理者が設定した評価指標のうち向上効果の高いプロファイル情報項目を変化させるための制御指令を出したときの効果をシミュレーションする。仮想的に生成した作業員及び生産設備のプロファイル情報値に基づいて、逐次的に分析を行い、性能低下を予測できるか、また、性能改善のための制御指令を作成できるかを検証するアプリケーションを試作し、市販のエッジコンピューター上で検証を実施した。図2に試作した検証画面例を示す。



図2-生産システム運用制御方式の検証画面例

この例では、2台のピッキングロボットと複数の作業者が弁当の具材を詰める作業ラインで、Edgexcrossで具材残量やロボットの稼働状況を収集する状況を模している。具材残量を予測しながら、具材残量が0になってラインが停止することのないようにあらかじめ制御指令を作成できるかを検証した。その結果、ほぼ遅延なくプロファイル情報の評価と制御指令の作成が可能であることを確認できた⁽⁴⁾。

(注1) 産業オートメーション分野で、異なるメーカーの機器やシステムが安全かつ確実にデータを交換するための国際標準規格。OPC UAは、OPC Foundationの登録商標である。

(注2) Edgexcrossは、一般社団法人 Edgexcrossコンソーシアムの登録商標である。

3. 処理優先度を考慮した並列機械工程スケジューリング方式

この章では、多品種少量生産を行う生産現場で、複数の生産設備に対する生産スケジュールを高速に作成する手法を提案する。

3.1 問題設定

想定したスケジューリング問題の前提条件について述べる。生産設備が m 台稼働している生産現場で、 n 個の“ジョブ”と呼ばれる生産工程が実行待ちになっている状態を仮定する。各ジョブ j はどの順番で実行してもよく、また、どの生産設備で処理してもよい。各ジョブ j の処理時間 l_j は使用する生産設備によって変化しない。また、その処理時間の最大値は23時間以内と定義する。各ジョブ j には、ジョブが実行完了されていることが望ましい期日として、納期 d_j がそれぞれ設定されている。各ジョブ j は1台の生産設備で処理できる単位で作られている。ジョブの種類としては、事前に作成する“通常ジョブ”に加えて、処理失敗や処理が不十分な場合等に急きょ作成する“割り込みジョブ”の2種類があり、各ジョブはそのいずれかに該当する。これらの前提条件に基づいて、次のように制約条件を設定する。

制約①納期遅れ和最小化：各ジョブ j に設定される納期 d_j は日単位で設定される。納期 d_j とスケジューリング後の処理日との差を“納期遅れ”として定義し、処理日が納期前であれば納期遅れは0とする。各ジョブの納期遅れの総和が小さくなるようにスケジュールを作成する。

制約②優先度制約：割り込みジョブにはそれぞれ優先度 p を設定し、優先度が高いものからスケジューリングされる。通常ジョブは全ての割り込みジョブが処理完了する日時以降でスケジューリングされる。

制約③稼働時間制約：各生産設備 i にはそれぞれ稼働時間上限 t_i を決めることができる。その場合、各生産設備 i が処理する各ジョブの処理時間の合計は t_i を超えない。

制約④負荷分散制約：各生産設備 i で処理するジョブの総処理時間がおおよそ均一になるように負荷分散を考慮したスケジュールを作成する。

制約⑤逐次実行制約：各生産設備 i で処理するジョブは逐次的に実行される。

3.2 提案手法

ここでは、3.1節で述べた前提条件の下、制約条件を満たすスケジュールを求める手法として、整数最適化問題で定式化する部分とルールベースのアルゴリズムで処理する部分に分ける方針とした。

整数最適化処理部分は、3.1節で述べた前提条件と制約①から制約④までの制約条件を整数最適化問題として定式化する。制約②と制約③は求めるスケジュールが必ず満たすべき条件であるため、絶対制約として最適化問題での制約条件式に組み込む。制約①と制約④は更に良い解を得るために満たすことが望ましい条件であるため、考慮制約として、最適化問題での目的関数に組み込む。定式化は次のとおりになる。

$$x_{ij} = \begin{cases} 1 & (\text{生産設備 } i \text{ にジョブ } j \text{ を割り当て}) \\ 0 & (\text{otherwise}) \end{cases} \quad \dots\dots\dots (1)$$

$$l_j \in \mathbb{Z} : \text{ジョブ } j \text{ が生産設備で処理される時間} \quad \dots\dots\dots (2)$$

$$J : \text{全てのジョブの集合} \quad \dots\dots\dots (3)$$

$$d_j \in \{0, 1, 2, \dots\} : \text{ジョブ } j \text{ の納期までの日数} \quad \dots\dots\dots (4)$$

$$t_i : \text{生産設備 } i \text{ の最大稼働時間} \quad \dots\dots\dots (5)$$

$$p \in \{1, 2, \dots\} : \text{各ジョブ } j \text{ に設定された優先度} \quad \dots\dots\dots (6)$$

$$\begin{aligned}
 J_p \subseteq J: & \text{優先度 } p \text{ のジョブの集合} \cdots \cdots \cdots (7) \\
 b_p \in \{0, 1\}: & \text{優先度制約に関する補助変数} \cdots \cdots \cdots (8) \\
 e_j = f(d_j): & \text{納期遵守に対する利得} \cdots \cdots \cdots (9) \\
 \text{minimize} & \quad -P_1 - P_2 \cdots \cdots \cdots (10) \quad \text{制約①+制約②の目的関数} \\
 \text{subject to} & \quad P_1 = \sum_i \sum_j x_{ij} e_j \cdots \cdots \cdots (11) \quad \text{制約①納期遅れ和最小化} \\
 & \quad P_2 = y \cdots \cdots \cdots (12) \quad \text{制約④負荷分散制約} \\
 & \quad y \leq \sum_{j \in J} l_j x_{ij} \quad (i=1, 2, \dots, m) \cdots \cdots \cdots (13) \quad \text{制約④負荷分散制約} \\
 & \quad \sum_{j \in J} l_j x_{ij} \leq t_i \quad (i=1, 2, \dots, m) \cdots \cdots \cdots (14) \quad \text{制約③稼働時間制約} \\
 & \quad \sum_i x_{ij} \leq 1 \quad (j=1, 2, \dots, n) \cdots \cdots \cdots (15) \\
 & \quad b_p \geq b_{p+1} \quad (p=1, 2, \dots) \cdots \cdots \cdots (16) \quad \text{制約②優先度制約} \\
 & \quad \sum_i x_{ij} \leq b_p \quad (j \in J_p, p=1, 2, \dots) \cdots \cdots \cdots (17) \quad \text{制約②優先度制約} \\
 & \quad b_{p+1} \leq \sum_i x_{ij} \quad (j \in J_p, p=1, 2, \dots) \cdots \cdots \cdots (18) \quad \text{制約②優先度制約}
 \end{aligned}$$

続いて、ルールベース処理部分について述べる。3.1節で述べた各制約条件のうち、制約⑤をルールベースのアルゴリズムで処理する。具体的には、式(1)~(18)で表現した整数最適化問題を解くことによって得られる、各ジョブと処理する生産設備との対応関係を用いて各生産設備の時系列スケジュールを作成する処理を、ルールベースのアルゴリズムを用いて実行する。用いるアルゴリズムを次に示す。

Step 1: 各生産設備に割り当てられたジョブを集める。

Step 2: 集めたジョブのうち、割り込みジョブを優先度が高い順に並べる。優先度が同一の場合は納期に余裕がないジョブから並べる。

Step 3: 全ての割り込みジョブの後に、通常ジョブを納期に余裕がないジョブから順に並べる。

Step 4: 並べたジョブと時間軸を対応させることで時系列に並んだ生産スケジュールを作成する。

3.3 数値実験

生産現場での標準的な放電加工工程を模したジョブを用意し、それらを提案手法に適用することによって、生産スケジュールを求める数値実験を行った。実験方法について述べる。用意するジョブは幾つかのテストケースを想定して作成し、それぞれに対して想定例となる生産スケジュールをあらかじめ求めておく。各テストケースに対して提案手法を適用することで求めたスケジュールと想定例のスケジュールを比較し、提案手法の有効性を確認する。

今回の数値実験は、次に示す設定で実施した。

- ①スケジュール作成期間 : 7日間
- ②稼働加工機台数 : 3台
- ③各加工機の1日当たりの稼働時間 : 23時間
- ④各テストケースの実験回数 : 20回
- ⑤スケジューリング実施単位 : 1日
- ⑥実験パソコンのCPU : 11th Gen Intel Core^(注3) i5-1145G7@2.60GHz
- ⑦実験パソコンのメモリー : 16GB

スケジューリング実施単位とは、提案手法を1回適用することによって求めるスケジュールの期間である。今回の数値実験では、日付をまたぐジョブが発生しないようにするために、スケジューリング実施単位を1日とした。

テストケースは次のような三つの例を用意し、それぞれ想定例となるスケジュールを作成した。Case 1, Case 2, Case 3の順に、より難易度が高くなるようにテストケースの設計を行った。

Case 1: 1日分として処理時間の和が23時間になるようにジョブを作成する。これをスケジュール作成期間、及び稼働加工機台数だけ用意する。このとき、各ジョブの優先度は同一である。また、想定例で各日にスケジューリングされるジョブはその日が納期になっている。また、各ジョブの処理時間は60分の倍数とする。

Case 2: Case 1で作成した各ジョブについて、優先度を設定する。

Case 3: Case 2で作成した各ジョブに、処理時間が60分で割り切れない“500分”のジョブを稼働加工機台数分追加する。“500分”のジョブの納期は7日目として、一つでもスケジューリングすると1日当たり23時間分のジョブを隙間なく計画できなくなるため、納期違反が発生することが予想される。

各テストケースの想定例として、1日分のスケジュール例を図3に示す。

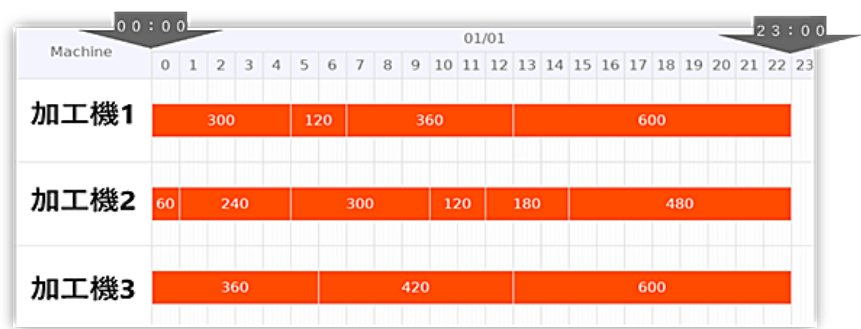


図3-並列機械工程スケジューリング方式の想定例

図3は3台の加工機の1日のスケジュール例を示す。横軸が1時間ごとの時間軸、縦軸が各加工機のスケジュール結果である。上から加工機1、加工機2、加工機3で、割り込みジョブは赤色、通常のジョブは青色で表記している。また、各ジョブの矩形(くけい)内の数値は加工時間(分)を表す。このように想定例では0時から23時まで空き時間のないスケジュールを作成できる。したがって、提案手法で求めたスケジュールに空き時間が存在しなければ、想定例と同様の最適解を得ることができたと考えられる。提案手法で求めたスケジュールに空き時間が存在する場合、求めたスケジュールがどの程度最適解に近いかを想定例との乖離(かいり)程度から評価することにして、次のような評価指標を設定した。どの指標も、想定例と同等又はそれ以下であることが評価条件になる。

- ①納期違反数：各ジョブをスケジューリングした結果、あるジョブが加工される計画日とそのジョブの納期よりも後である場合、納期違反になる。この納期違反を各ジョブに対して検査することで求めた納期違反数を評価指標とする。
- ②納期違反量：各ジョブをスケジューリングした結果、あるジョブが加工される計画日とそのジョブの納期よりも後である場合、そのジョブの加工時間を納期違反量とする。ただし、計画日が納期より前である場合は納期違反量が0であるとする。このとき、各ジョブの納期違反量の総和を取ったものを評価指標とする。
- ③演算時間：生産現場ではスケジューリングで許容できる時間に制限があることが多いため、上限を60秒として、この上限以内に計算が終了しているか否かを評価指標とする。

式(11)で、最小化している納期遅れ和は上記評価指標のうち、納期違反数と納期違反量で評価する。これは、納期遅れが1日単位で算出されるため、加工時間の大きいジョブが納期遅れになっても、加工時間の小さいジョブが納期遅れになっても同一評価になることを防ぐためである。提案手法によって求めたスケジュール、想定例のスケジュールのそれぞれに対して、これらの評価指標で評価を行い、その差を用いて提案手法によって求めたスケジュールが妥当であることを検証する。

テストケース(Case 1～3)に対して、提案手法を適用することによって求めたスケジュールが制約①納期遅れ和最小化～⑤逐次実行制約を満たしているかを検証する。まず、提案手法を各テストケースに適用することで、生産スケジュールを求める。次に、これらの求めた生産スケジュールに対して、各評価指標に基づいて評価値を算出する。この評価結果を表1に示す。各評価指標の値は、20回のスケジューリングを実行して得られたデータの平均値である。

表1-各評価指標での提案手法の評価結果

	Case 1	Case 2	Case 3
納期違反数	0	0	3
想定例での納期違反数	0	0	3
納期違反量(分)	0	0	1,500
想定例での納期違反量(分)	0	0	1,500
平均演算時間(秒)	1.836	1.970	2.589

表1に示すように、Case 1～3の全てのテストケースで、あらかじめ作成しておいた想定例と比較すると、全ての評価指標が想定例と同等のスケジュールを得た。また、最も複雑なテストケースであるCase 3で、表1には表れない制約②優先度制約の条件が満たされているを確認するために、作成されたスケジュールを図示したものを図4に示す。図4

は、提案手法によるスケジューリング結果のうち、7日目と8日目を抜粋したものである。7日目以前のスケジュール期間内に“500分”のジョブがスケジューリングされた場合、想定例での各評価指標の値よりも悪化することが考えられる。しかし、図4の結果では想定したスケジュールと同等であり、各評価指標の値も同等であることが確認できた。また、図4上部に表示した1～8日目のスケジューリング結果全容を見ると、1～3日目には、割り込みジョブを表す赤色のジョブが固まっており、通常のジョブを表す青色のジョブは4日目以降に固まってみられることから、制約②優先度制約も満たされていると言える。以上の結果から、式(1)～(18)によって、最適スケジュールを作成できたと考えられる。すなわち、提案手法は、ジョブ数が少ない簡単なテストケースで、最適なスケジュールを求めるという基本性能を持っていることが確認できた⁽⁵⁾。

(注3) Intel Coreは、Intel Corp.の登録商標である。



図4-Case3のスケジューリング結果(7～8日目)

4. む す び

生産現場で、現場作業者と生産設備とが混在している状況で、運用を自動化するための手法として、ライン稼働データをエッジプラットフォームで収集・分析することによってライン全体の統合監視を行う性能プロファイル連携制御方式と、生産現場で高速に工程実行順序を決定するために、制約条件を整数最適化処理部分とルールベース処理部分とに分けたことを特徴とする工程スケジューリング方式について述べた。どちらの方式も生産現場で要求される秒単位での計画作成を達成できた。

参 考 文 献

- (1) 産業競争力懇談会COCN：「人」が主役となる新たなものづくり，産業競争力懇談会 2017年度 プロジェクト 最終報告 (2018)
<http://www.cocn.jp/report/theme97-L.pdf>
- (2) 一般社団法人 Edgecross コンソーシアム：Edgecross Consortium
<https://www.edgexcross.org/ext/ja/index.html>
- (3) 中井敦子：エッジプラットフォームを用いたROSと生産システムとの連携方式に関する一検討，2022年電気学会電子・情報・システム部門大会，OS1-6 (2022)
- (4) 中井敦子：作業者と生産設備とのプロファイル情報を用いた生産システム運用制御方式の検討，第22回情報科学技術フォーラム，J-022 (2023)
- (5) 矢口喬脩，ほか：処理優先度を考慮したFA向け並列機械工程スケジューリング，2023年電気学会電子・情報・システム部門大会，OS1-2-5 (2023)