FMSにおける加工機械と複積載搬送ロボットの同時スケジューリング問題

Simultaneous Scheduling Problem of Processing Machines and Multi-Load Transport Robots in the FMS

汪 永豪 [*1]　　山田 孝子 [*2]
Wang Yonghao　　Takako Yamada

[*1]関西学院大学大学院総合政策研究科
Graduate School of Policy Studies, Kwansei Gakuin University

[*2]関西学院大学総合政策学部メディア情報学科
Department of Applied Informatics, School of Policy Studies, Kwansei Gakuin University

With the globalization of the world economy, manufacturing companies must accept orders at lower prices than their competitors. As a result, achieving low-cost and high-quality production management has become increasingly important, posing a challenging issue that all manufacturers must overcome. To reduce costs, many companies have adopted Automated Guided Vehicles (AGVs) instead of human workers to transport parts between machines in a flexible manufacturing system (FMS). Additionally, since machines in an FMS are not always in operation, it is crucial to utilize their idle time effectively by assigning them varying tasks. This study aims to achieve simultaneous scheduling of both machines and AGVs, minimizing the completion time of existing tasks while making effective use of machine idle time to process additional parts.

1.　はじめに

世界経済のグローバル化に伴い、製造業企業は他社より安い価格で注文を受ける必要がある．いかにして低コストな生産システムと高品質な品質管理を両立させ，安定的に維持するかは、製造業が必ず直面すると言って過言ではない難しい問題である．昨今ではこの両立にさらに安全管理を含むワークライフバランスを考えた労務管理が加わり，生産現場では達成するべき課題が複雑化している．コスト削減のため、多くな企業は人間作業員の代わりに、製造工程のロボットを採用し，工場内物流には複積載搬送ロボット (AGV) の導入をすすめている．AGV の導入により生産機械間の半製品や仕掛品の運搬は自動化された．生産計画には, スケジューリングだけではなく,AGV の搬送計画も含まれ, 経路や運搬すべき半製品の決定までを生産計画に含むことになった．本研究では，以下のシステムを順次作成し，生産管理計画システムとして実装し，最適解を求める計算機実験を実施，システムの性能を検証した．本研究でこれまで行ったことをまとめる

1. 実験用ジョブセットとして，製品種別ごとの加工機械の利用順番，利用時間の組み合わせデータの自動生成システムの作成
2. 生成されたデータを用いた生産機械のスケジューリングを総当たり法で厳密解を求める最短生産スケジュール算出システムの作成
3. 生産スケジュールの可視化システムの作成
4. 遺伝的アルゴリズムをベースとする，緩和手法による許容解算出と解の改善システムの作成
5. 生産機械に加え複数台 AGV を導入した生産スケジュールの最適化システムの作成
6. 求められた生産スケジュールに基づく生産機械の遊休時間活用する余剰生産品提案システムの作成

以上を行ったうえで、本研究では機械と AGV 両方同時スケジューリングを求め、現有な仕事を最短時間で終了する許容解を導出し, その許容解のもとで, 機械の遊休時間を活用する余剰製品を提案するシステムの開発に取り組む．このような問題は、JSSP(Job Shop Scheduling Problem) 問題と言います。

2.　モデル

2.1　モデル仮定

本モデルでは受注生産を行う工場の製造工程をモデル化する．

1. 注文票はさまざまな注文主からくる。
2. 各注文書には製品 S の種別が 1 種類が記載されている。注文する製品種別が異なる製品は異なる注文書で発注される．

連絡先: 関西学院大学総合政策研究科, 兵庫県三田市学園上ヶ原, ili53007@kwansei.ac.jp

3. 注文書ごとに価格、納期、ロット数がある。
4. 工場は、出荷する出荷日（納期）が同じ注文書 O_i をまとめ、これを注文書セットとする。
5. 製造する製品種別に応じて、使用する加工機械、加工時間とその加工機械の使用順が予め与えられる。
6. 工場は注文書セットで同一出荷として受注した注文書を納期日までに出荷できるよう、加工機械と AGV の最適なスケジューリングを行う。
7. スケジューリングの結果得られた注文書セットに含まれる受注した製品を定められた個数すべて出荷できる日を最速出荷日とする．また最速出荷日は早いほどよいとする．
8. 機械は注文書に記載された製品種別で指定された注文個数をすべて加工が終了するまで，作業を中途で終えたり，途中で他の製造作業を行うような割り込みはできないものとする。
9. 製品種別が異なるための機械のセットアップタイムは考慮しない。
10. 機械は製品種別ごとに定まる順番の通りに機械を使用して加工作業を行わなければならない。
11. 出荷が等しい注文書，すなわちある注文書セットがすべて出荷可能とればその注文書セットでの加工作業は終了となる。

2.2 受注

製品を製造する工場をここでは考える．一定の期間に工場には様々な顧客から注文票が到着する．ある期間中に到着した注文票を 0 とあらわす．この注文票の集合は O_1 から O_i までの i 枚の注文票からなる．各注文票 O_i には，注文する製品名 S_s，納期 d_i，各製品名注文ロット数 lnS_{is}, 注文する製品の単価 $price_s$ が記載されている．注文票に記載できる製品は 1 種類で，複数の製品を発注する場合には，注文票を分けることになっている．工場は当該期間に受注した注文票で最も納期の早い注文票の納期を d_{min} として，その期日に間に合うように，受注した製品の製造計画を立てる．

2.3 製造

製造計画の作成するためには，まず製品種別 S_s ごとに，加工工程として使用する製造機械の順番が決まっている．製造機械を使用する時間は製造する個数による．そのため製品 S_s を製造するための q 番目の加工工程を f_{sq} としてあらわし，そこで使われる製造機械番号を m_{sq}, 加工時間を t_{sq} で表す.

2.4 仕掛品搬送

加工工程で、異なる製造機械の間で仕掛品搬送に AGV(Automated Guided Vehicle) を用いる場合，製品 S_s の第 f_{sq} 工程と第 $f_{s,(q+1)}$ 工程間を AGV_j で搬送するとき，その輸送時間を $ut_{j,s,q,(q+1)}$ で表す．以上を記号として整理すると，以下のようになる．

3. 符号定義

O: 工場が受領した注文書の集合

O_i: 第 i 番に工場が受領した注文書

d_{min}: 受注した注文票で最も納期の早い注文票の納期

d_i: 第 i 番に工場が受領した注文書の納期

Ss: 製品 S の種別番号 s

Sis: 第 i 注文書に記載された注文製品 S の種別番号 s

$price_s$: 第 i 注文書で受注した製品 S の種別番号 s の単価

lnS_{is}: 第 i 注文書に記載された第 s 種別番号製品の注文ロット数

n_s: すべての注文書に記載された第 s 種別番号製品の合計注文個数

$$n_s = 10\sum_{i=1}^{\infty} lnS_{is}$$

N: すべての製品の合計注文個数

$$N = \sum_{s=1}^{\infty} n_s$$

f_{sq}: 第 s 種別製品の第 q 番目の加工工程

m_{sq}: 第 s 種別製品の第 q 番目の加工工程に割り当てられる機械番号

$$M: \text{使用可能な加工機械の集合}$$
$$m_m: \text{第 } m \text{ 番目の加工機械}$$
$$t_{sq}: \text{第 } s \text{ 種別製品の第 } q \text{ 番目の加工工程にかかる加工時間}$$
$$l_1, l_2 \subseteq (L/U, m_1, \cdots, m_m)$$
$$l_1 \neq l_2$$
$$AGV_j: \text{第 } j \text{ 番目の AGV}$$
$$ht_{j,l_1,l_2}: AGV_j \text{ が } l_1 \text{ から } l_2 \text{ までの搬送時間}$$
$$ut_{j,s,q,(q+1)}: AGV_j \text{ が製品 } s \text{ の第 } q \text{ 工程と第 } q+1 \text{ 工程間の輸送時間}$$

3.1 実験用ジョブセット生成

一般的に企業への注文書の具体例は外部には公開されない．そのため本研究では注文書セットを予め様々な確率的に生成する計算機実験用の自動生成システムを作成し，実験用の注文書セットを作成する．本節ではこの自動生成の手続きとそれにより作成した注文書セットについて述べる．

3.2 記号定義

$$\begin{aligned} \mu_{odder}&: \text{納期内の総オーダー数の平均値}\\ \sigma^2_{odder}&: \text{納期内の総オーダー数の分散}\\ \mu_{lot}&: \text{各オーダーに注文された製品のロット数の平均値}\\ \sigma^2_{lot}&: \text{各オーダーに注文された製品のロット数の分散} \end{aligned} \tag{1}$$

3.3 生成ルール

1. 多層回路基盤を作る工場等、同じ工程が何回も加工する加工工程に向けて、生成する。
2. 正規分布を用いてオーダー数 O_i を生成 ($\mu_{odder} = 10$、$\sigma^2_{odder} = 2$)。
3. 負の値やゼロを除外し、サンプルからランダムに 1 つの整数を選んで、今回のシミュレーションの注文数とする。
4. 各注文は最大で 6 種類の製品を含む。
5. 各種別の製品について、固定されている制作工程がある、工程間は転倒しない。
6. 各製品のロット数は正規分布 ($\mu_{lot} = 3$、$\sigma^2_{lot} = 1$) から生成し、負またはゼロを除外。
7. 各製品は注文内で 0.7 の確率で含まれるによる制御。
8. 各製品の総数量は、すべての注文におけるその製品のロット数合計に 10 を掛けて算出。
9. m_1 から m_{13} まで 13 個それぞれ役目が重ねない加工機械がある。
10. 各加工工程の操作時間 T_J: 一部は製品数量に単位時間を掛けて決定、一部は固定時間。
11. 各製品種別に対し、加工工程と対応する操作時間の組み合わせを記録する。

同じジョブの作業について、前作業と後作業があり、作業の前作業と後作業のどれかの作業機械は一致しない.

1. $before\{m_{sq}\} \neq m_{sq}$
2. $after\{m_{sq}\} \neq m_{sq}$

3.4 レイアウト、AGV 搬送設定

1. 工場には L/U(Lording/Unlording)、機械 ($m_1 \sim m_m$) がある。
2. 各 AGV は L/U から出発、各機械の間を部品を搬送する。
3. 製品の最後の一つ加工操作が終わった後に、AGV は最後の一つの加工操作を務めた機械のところで、部品を受取、L/U まで搬送する。
4. 注文書に書かれているすべての製品が L/U まで搬送したら、加工終了し総操作時間加算する。
5. AGV は一回限りに一種類の部品を搬送します、容量は考慮しません。
6. AGV_j が L/U から各機械への搬送時間は同じく $ht_{j,L/U,m_m} = 30$。
7. AGV が機械の間での搬送時間は、相隣の機械は同じく 20、機械の間に挟まれた機械の個数分かける 10 加算する。
8. 一つの搬送作業に対して、AGV1 台で搬送する。
9. 一つの搬送作業に対して、一回しかしない。

4. 提案手法

本研究では、JSSP 問題に対して、GA を用いた探索手法を提案する。また、こういう加工順序が厳格に定められている問題には、PPS(Precedence Preservative Crossover) という交差手法を採用します。

4.1 PPS 交差法

PPS 交差法は、JSSP 問題などの順序制約を伴う組合せ最適化問題において有効な交差手法の一つである。本手法は、親個体の遺伝子情報を組み合わせる際に、工程間の優先順序を保持しつつ、新たな子個体を生成することを目的としている。

1. 2 つの親染色体 (Parent1, Parent2) を選択する。
2. Parent1 から順序情報を、Parent2 から操作集合の構成を主に利用する。
3. 子個体の染色体を初期化する。
4. Parent1 の順序に従い、Parent2 に存在する操作を逐次選びながら、子個体に挿入する。ただし、すでに挿入された操作との順序制約を満たすように注意深く挿入される。
5. すべての操作が挿入されるまで繰り返す。

数式：

- $S = \{S_1, S_2, \ldots, S_s\}$：製品種類の集合（$s$ は製品種類数）
- 各製品 S_s は q 個の加工操作を持ち、$f_s = \{f_{s1}, f_{s2}, \ldots, f_{sq}\}$ とする。
- 親個体は操作列として表現される：$P_1 = [o_1^1, o_2^1, \ldots, o_m^1]$, $P_2 = [o_1^2, o_2^2, \ldots, o_m^2]$
- 子個体 C を空列として初期化：$C \leftarrow [\ ]$

1. $i \leftarrow 1$
2. **while** $|C| < m$ **do**:
   (a) $o \leftarrow P_1[i]$ (親 1 の i 番目の操作)
   (b) **if** $o \in P_2$ かつ precedence_satisfied$(o, C) =$ True **then**:
      - $C \leftarrow C \cup \{o\}$ (子供に操作を追加)
      - $P_2 \leftarrow P_2 \setminus \{o\}$
   (c) $i \leftarrow i + 1$
3. **return** C

ここで、precedence_satisfied 関数は以下で定義される：

$$\text{precedence\_satisfied}(f_{sq}, C) = \begin{cases} \text{True} & \text{if } f_{s,(q-1)} \in C \text{ または } q = 1 \\ \text{False} & \text{otherwise} \end{cases}$$