

バッチサイズをもつ並列機械スケジューリングに対する ヒューリスティックアルゴリズム

05001498 筑波大学 *上田寛人 UEDA Hiroto
01206770 筑波大学 繁野麻衣子 SHIGENO Maiko

1. はじめに

近年製造業はDX化に伴い効率的な生産をおこなっている。一部工程では完全自動化体制を採用して高品質の維持をも実現し、顧客満足度の向上を達成している。しかし自動化の恩恵を享受するためには効率的な生産計画を立てる必要がある。

生産スケジューリングは60年以上前から理論・応用ともに多くの研究がなされてきた[1]。本稿ではUedaら[4]にて扱われた超精密加工品製造企業から派生した生産計画問題を扱い、本問題に対するヒューリスティックアルゴリズムを提案する。

2. 問題構造

生産計画期間の時間集合を T とする。生産する製品をジョブと呼びそれらの集合を J とする。各ジョブ j は仕様 r_j 、原材料使用量 l_j 、処理時間 p_j 、納期 d_j を持ち、機械で1回処理する。原材料使用量と処理時間は比例関係にあり仕様は原材料の特性から決まる。仕様集合を R とし仕様 r を持つジョブ集合を $J(r) = \{j \in J | r_j = r\}$ とする。ジョブの情報は完全で任意の時点から処理を開始できる。

生産工程では同じ仕様を持つ複数のジョブをグループ化しブロック(バッチ)を形成する。あるブロックの処理が開始されるとブロック内の全てのジョブが順番に処理され、ブロックの処理が全て終了するまでブロック内のジョブを取り出すことはできない。各ブロックは原材料使用量を持ち、ブロック内のジョブの原材料使用量の合計で定義される。ただし1ブロックの原材料使用量には上限 $L_r(r \in R)$ が存在し、それを超えてはならない。

機械集合を M とし、各機械 $m \in M$ は処理可能な仕様集合 $R(m) (\subseteq R)$ を持つ。よって仕様 r をもつブロックを $M(r) = \{m \in M | r \in R(m)\}$ 内のいずれかの機械に割り当てる。ただし実務では異なる仕様のブロックを1台の機械に割り当てることは非常に非効率であると見なしている。よって各機械に対し1つの仕様を割り当てる。

各ブロックの処理前後にはオペレータによるセットアップ作業とメンテナンス作業が必要で、それぞれ s, q の時間を要する。事前に割り振られた複数のオペレータ(h 人とする)は誰でもこの作業を行うことができ、担当者によって作業時間は変わらない。各作業はオペレータが1人で行うため、同時刻に最大 h 台の機械で作業を行える。

管理者は、1. 各ジョブの納期を守る。もし納期遅れするジョブがあれば納期遅れ時間を最小にする。2. 原材料ロスを減らすためブロック数を削減。3. 急な注文への対応のため稼働機械台数を削減。を考慮し生産計画を立てる。よって我々もこれら基準の加重和(α_D : 1の重み, α_B : 2の重み, α_M : 3の重み)を最小にする生産計画の作成を目指す。

3. ヒューリスティックアルゴリズム

Uedaら[4]は2節の問題に対し複数定式化を提示し、実験を行いoperator-freeモデルが最良と結論づけた。このモデルはまずオペレータ制約を無視した問題(GBAM問題)を解き、次にオペレータ制約を満たすよう作業時刻を調整して実行可能な計画を作成する。オペレータ制約を満たすためのアルゴリズムはUedaらで提案されており、本稿はGBAM問題に対するアルゴリズムを提案する。

GBAM問題はパラメータを導入し2つの部分問題に分解できる。まず各仕様別に事前に機械を $\lambda_r(r \in R)$ 台に固定する。すると仕様別に独立してジョブのブロックへのまとめ方と λ_r 台の機械への割り当て方を考えれば良い。この各仕様別に独立した問題を解くものが1つ目の部分問題(SP1)である。残りは、各機械が処理可能な仕様集合 $R(m)$ を持つという制約の下、各 $r \in R$ の最適パラメータ λ_r^* を決定する問題(SP2)である。つまりどの機械にどの仕様を割り当てるかを決定する。

3.1. SP1に対するアルゴリズム

SP1を解くアルゴリズムはLinら[3]の動的計画アルゴリズムを元に構築される。まず $J(r)$ に含ま

れる各ジョブを EDD(Earliest Due Date) 順に並べる (Lin らの場合は EDD 順により最適性が保証されるが SP1 では最適性は成立せず近似解法となる). 次に以下 3 つの再帰関数を用意する.

$$h(0; t_1, \dots, t_\lambda) = \begin{cases} 0 & (t_1 = \dots = t_\lambda = 0) \\ \infty & (\text{otherwise}) \end{cases}$$

$$h(j; t_1, \dots, t_\lambda) = \min_{\ell=1, \dots, \lambda} \min \{ \hat{h}_\ell(j; t_1, \dots, t_\lambda), \tilde{h}_\ell(j; t_1, \dots, t_\lambda) \}$$

$$\hat{h}_\ell(j; t_1, \dots, t_\lambda) = \begin{cases} h(j-1; t_1, \dots, t_\ell - p_j, \dots, t_\lambda) + \\ \alpha_D (\max\{t_\ell - d_j, 0\} + \sum_{k \in H_\ell(j-1; t_1, \dots, t_\ell - p_j, \dots, t_\lambda)} \\ \max\{t_\ell - \max\{d_k, t_\ell - p_j\}, 0\}) \\ (H_\ell(j-1; t_1, \dots, t_\ell - p_j, \dots, t_\lambda) \neq \emptyset, \\ \sum_{k \in H_\ell(j-1; t_1, \dots, t_\ell - p_j, \dots, t_\lambda)} l_k + l_j \leq L_r) \\ \infty & (\text{otherwise}) \end{cases}$$

$$\tilde{h}_\ell(j; t_1, \dots, t_\lambda) = \begin{cases} \alpha_B + \alpha_D \max\{t_\ell - d_j, 0\} & (t_\ell = s + p_j) \\ h(j-1; t_1, \dots, t_\ell - p_j - s - q, \dots, t_\lambda) \\ + \alpha_B + \alpha_D \max\{t_\ell - d_j, 0\} & (\text{otherwise}) \end{cases}$$

$h(j; t_1, \dots, t_\lambda)$ は最初の j 個のジョブが処理され、各機械 ℓ の処理完了時間が t_ℓ となるときのブロック数と納期遅れの加重和の最小値である. $\hat{h}_\ell(j; t_1, \dots, t_\lambda)$ と $\tilde{h}_\ell(j; t_1, \dots, t_\lambda)$ は、それぞれジョブ j が新規ブロックとして機械 ℓ に追加されるとき、そのジョブ j が機械 ℓ の最後のブロックを形成する 1 つとなるときのブロック数と納期遅れの加重和である. $H_\ell(j; t_1, \dots, t_\lambda)$ は $h(j; t_1, \dots, t_\lambda)$ の値のとき機械 ℓ の最後のブロックに含まれるジョブ集合である. $|J(r)|$ 個のジョブの EDD 順に対し

$$\min \{ h(|J(r)|; t_1, \dots, t_\lambda) \mid t_\ell \in T \cup 0, \ell = 1, \dots, \lambda \}$$

を見つけることで機械台数が λ_r のときの最適状態を得る. h は $O(|J(r)| \cdot |T|^\lambda)$ の状態を持ち、 \hat{h} と \tilde{h} の比較は $O(\lambda)$ を要する. \hat{h} と \tilde{h} はそれぞれ $O(\lambda |J(r)| \cdot |T|^\lambda)$ の状態を持ち、 $\tilde{h}_\ell(j; t_1, \dots, t_\lambda)$ の値は $O(|H_\ell(j; t_1, \dots, t_\lambda)|) = O(|J(r)|)$ で計算可能である. ゆえに最適値を得るための計算量は $O(\lambda |J(r)|^2 |T|^\lambda)$ であり、機械数が固定されているとき擬多項式アルゴリズムである.

3.2. SP2 に対するアルゴリズム

SP2 を解くアルゴリズムは Hefner ら [2] の Master-Slave マッチング問題を解くアルゴリズムを元に構築される. 実応用では機械を多数使用することはない. よってその特性を活かし、逐次頂点に関する Master-Slave 制約を追加し Hefner らのアルゴリズムを繰り返し解くことで解を得る.

4. 数値実験とまとめ

提案アルゴリズムの性能と計算時間を比較するため、SP1, SP2 に対する整数計画モデルを作成し数値実験をおこなった. 表 1 はその結果であり 10 個のインスタンスに対する平均を表す. なお全ての重みは 1 である. 規模の大きな問題は Gurobi 9.1.1 を用いても 1 時間で最適解が得られず、提案アルゴリズムは実用的な解を高速に導出した.

表 1: 提案アルゴリズムと整数計画法との比較

(a) $ J = 100$				
	ブロック数	機械数	納期遅れ (時間)	計算時間 (秒)
提案法	31.2	11.1	3.1	0.67
整数計画法	31.8	10.1	3.1	306.91
(b) $ J = 200$				
	ブロック数	機械数	納期遅れ (時間)	計算時間 (秒)
提案法	59.3	17.6	7.4	1.86
整数計画法	58.0	17.3	5.7	3781.31

参考文献

- [1] H. Y. Fuchigami and S. Rangel: A survey of case studies in production scheduling: Analysis and perspectives. *J. Sci. Comput.*, **25** (2018), 425–436.
- [2] A. Hefner and P. Kleinschmidt: A constrained matching problem. *Ann. Oper. Res.*, **57** (1995), 135–145.
- [3] B. M. T. Lin and A. A. K. Jeng: Parallel-machine batch scheduling to minimize the maximum lateness and the number of tardy jobs. *Int. J. Prod. Econ.*, **91** (2004), 121–134.
- [4] H. Ueda, Y. Chen, K. Sato, M. Shigeno, and U. Sumita: A comparison of integer programming based models for parallel machine batch scheduling. *Proc. the International Symposium on Scheduling 2021*, 24–29.