Vol.2019-AL-175 No.8
2019/11/28

処理速度可変な並列機械でのスケジューリングにおける終了時間とエネルギー量の和の最小化

藤森 友誠[1]　河瀬 康志[1]　松井 知己[1]　塩浦 昭義[1,a)]

1. はじめに

本稿では，ジョブの処理速度が可変な並列機械におけるスケジューリング問題を扱う．目的は，ジョブの総終了時間と機械の総消費エネルギー量の重み付き和を最小化することである．この問題は Bampis ら [1] により考えられ，そこでは機械が同質 (homogeneous) の場合 (以下 (HO)) が扱われているが，本稿ではより一般的な，機械が異質 (heterogeneous) な場合 (以下 (HE)) についても扱う．

Bampis ら [1] は問題 (HO) が最小重み最大マッチング問題へ帰着できることを証明した．そして，マッチング問題に対する既存のアルゴリズムを適用することにより，問題 (HO) が $O(n^3m^2 + n^2m^2 \log m)$ 時間で解けることを示している．ここで m は機械数，n はジョブ数である．

本稿ではまず，Bampis ら [1] の手法を一般化して，問題 (HE) においても，最小重み最大マッチング問題へ帰着することができることを示す．次に，マッチング問題への帰着で用いられる二部グラフの枝重みのもつ性質を明らかにする．この性質を利用することにより，このマッチング問題が容易に解けることを示す．その結果，問題 (HO) は $O(n \log n)$ 時間で，問題 (HE) は $O(n \log mn)$ 時間で，それぞれ解くことができる．本稿ではさらに，総終了時間と総消費エネルギー量に関するパラメトリック最適化を考え，(HO) に対する結果を利用することにより，その最適解すべてが効率的に計算できることを示す．

2. 問題の定義

機械の集合を $M = \{1, 2, \ldots, m\}$，ジョブの集合を $N = \{1, 2, \ldots, n\}$ とおく．ジョブ $j \in N$ の仕事量を w_j とする．各ジョブを処理する際には分割は不可能であり，処理を開始してからの機械の変更は不可とする．ジョブ j の処理終了時間を C_j と表す．機械 $i \in M$ におけるエネルギー消費率関数 $P_i(s)$ は処理速度 s を引数とし，非負の値をとる狭義単調増加な凸関数であり，微分可能とする．さらに，$P_i(s)/s$ は s に関して凸であると仮定する．そのようなエネルギー消費率関数として，$P_i(s) = s^\alpha$ (α は $2 \leq \alpha \leq 3$ を満たす定数) がよく使われる．エネルギー消費率関数 $P_i(s)$ が機械に依存せず等しい場合，並列機械は同質であるといい，それ以外の場合は異質であるという．問題 (HO) および (HE) の目的は，非負定数 β に対して

$$\sum_{j \in N} C_j + \beta \sum_{i \in M} (\text{機械 } i \text{ での消費エネルギー量})$$

を最小化するスケジュール，つまり各ジョブの機械への割当てと処理の順番，および各時点での各機械の処理速度を求めることである．

問題 (HO) および (HE) においては，関数 $P_i(s)$ の凸性より，各ジョブを一定の速度で処理する最適スケジュールが存在する．したがって，ジョブ j の処理速度を変数 $s_j > 0$ により表し，またジョブ j を処理する機械を i_j とすると，機械での総消費エネルギー量は

$$\sum_{j \in N} (w_j/s_j) P_{i_j}(s_j)$$

と書くことができる．

3. マッチング問題への帰着

問題 (HE) が最小重み最大マッチング問題に帰着できることを示す．ジョブ $j \in N$ を機械 $i \in M$ において後ろから k 番目で，速度 s_j で処理するとする．このとき，ジョブ j の処理時間は w_j/s_j であるが，この時間は同じ機械で後ろから $1, 2, \ldots, k$ 番目に処理されるジョブの終了時間に加算される．また，ジョブ j の処理に必要なエネルギー量は $(w_j/s_j)P_i(s_j)$ となる．よって，問題 (HE) の目的関数にお

[1] 東京工業大学経営工学系
Department of Industrial Engineering and Economics, Tokyo Institute of Technology, Tokyo 152-8550, Japan
[a)] shioura.a.aa@m.titech.ac.jp

けるジョブ j に関する部分は

$$\kappa_{i,j,k} = (w_j/s_j)(k + \beta P_i(s_j)) \qquad (1)$$

となる．なお，処理速度 s_j はこの値 $\kappa_{i,j,k}$ を最小化するように設定される．$P_i(s)/s$ は s に関して凸であるという仮定より，そのような最小解 s_j は必ず存在し，方程式

$$sP_i'(s) - P_i(s) = k/\beta \qquad (2)$$

の解である．ここで $P'(s)$ は $P(s)$ の導関数を表す．

以上の事実をふまえ，次のように完全二部グラフ G を定義する．G の頂点集合は $U = N$, $V = \{(i,k) \mid i \in M,\ 1 \le k \le n\}$ とする．二部グラフの枝 $(j,(i,k))$ を選ぶことは，ジョブ j を機械 i に割り当て，後ろから k 番目で処理することを意味する．枝 $(j,(i,k))$ の重みを $\kappa_{i,j,k}$ とする．このとき，二部グラフ G における最小重み最大マッチング（枝数 n のマッチングの中で重みが最小のもの）は，最適なスケジュールにおける各機械へのジョブの割当ておよび処理の順序を与え，その重みは問題 (HE) の最適値に等しい．

4. 枝重みの性質とアルゴリズム

式 (2) の解 s_j の値はジョブ j には依存せず，機械 i および β と k に依存するので，$s(i,\beta,k)$ と書く．また，値 $sP_i'(s) - P_i(s)$ は s に関して単調非減少なので，$s(i,\beta,k)$ の値は k に関して単調非減少である．式 (2) に $s = s(i,\beta,k)$ を代入した式より，

$$\kappa_{i,j,k} = w_j \beta P_i'(s_j) = w_j \beta P_i'(s(i,\beta,k)) \qquad (3)$$

が得られる．つまり，$U = N$ に含まれる各頂点 j に対し重み w_j を割り当て，V に含まれる各頂点 (j,k) に対し重み $r_{i,k} \equiv \beta P_i'(s(i,\beta,k))$ を割り当てると，各枝 $(j,(i,k))$ の重み $\kappa_{i,j,k}$ は接続する頂点の重みの積で表現される．

この枝重みの性質に基づき，グラフ G の最小重み最大マッチングは以下の手順で計算できる．

ステップ 1: 頂点集合 V から重みの小さい順に n 個の頂点 (i_t, k_t) $(t = 1, 2, \ldots, n)$ を選ぶ.
ステップ 2: 頂点集合 U の頂点 (ジョブ) を重みの大きい順にソートする．一般性を失うことなく $w_1 \ge w_2 \ge \cdots \ge w_n$ とする.
ステップ 3: 各ジョブ j を頂点 (i_j, k_j) に割り当てる.

枝重みの式 (3) より，最適なマッチングにおいては，U の頂点はステップ 1 で選ばれる V の頂点に接続されることが分かる．さらに，ステップ 1 で選ばれる頂点を U の頂点と接続するときは，ステップ 3 のやり方で割り当てるのが最適であることが容易に示せる．したがって，上記のアルゴリズムにより最小重み最大マッチングが得られる．

次に，このアルゴリズムの計算時間を解析する．本稿では，方程式 (2) の解は定数時間で計算できると仮定する．頂点 $(i,k) \in V$ の重み $r_{i,k}$ は k に関して単調非減少なので，ステップ 1 は $O(n \log m)$ 時間で実行できる．とくに，同一機械の場合は，機械の数は n 以下で十分なので，計算時間は $O(n)$ 時間に減少する．ステップ 2,3 は $O(n \log n)$ 時間で実行できる．よって，問題 (HE) の場合は $O(n \log mn)$ 時間で，問題 (HO) の場合は $O(n \log n)$ 時間で，それぞれ最小重み最大マッチングが計算でき，最適なスケジュールが得られる．

5. パラメトリック最適化

問題 (HO) において，すべての $\beta > 0$ に対する最適解を求めるパラメトリック最適化問題を考える．以下では，任意の β に対する (HO) の最適解において，機械への割り当ておよび処理順序は同一としてよいことを証明する．つまり，パラメトリック最適化問題を解くには，ある 1 つの β に対する (HO) の最適解を求めれば十分である．

第 4 節では，グラフの各頂点に重みを割り当て，それを用いてマッチングを計算した．頂点集合 U に割り当てた重みは w_j であり，β の値が変わっても変化しない．一方，頂点集合 V に割り当てた重み $\beta P_i'(s(i,\beta,k))$ は機械 i に依存せず，また k に関して単調非減少なので，β の値が変わっても，重みの間の大小関係は変化しない．したがって，第 4 節のアルゴリズムにより得られるマッチングは β の値が変わっても最小重み最大マッチングである．つまり，任意の β に対する機械への割り当ておよび処理順序は同一としてよいことになる．

参考文献

[1] E. Bampis, D. Letsios, G. Lucarelli: Green scheduling, flows and matchings, Theoretical Computer Science 579 (2015), 126–136.