機械の取替時間を考慮する多機械スケジューリング問題に対する厳密解法

05001507　東京理科大学　*沼口寛樹　NUMAGUCHI Hiroki
05000348　静岡大学　呉偉　WU Wei
05000218　東京理科大学　胡艷楠　HU Yannan

1.　はじめに

機械スケジューリング問題は機械と仕事に関する制約のもとで，与えられる仕事を効率よく処理するスケジュールを求める問題である [1]．仕事を構成する順序付き作業を異なる機械で処理するときに発生する機械の取替時間を考慮することで，より現実に即したモデルを構築することが可能になる．

機械の取替時間を考慮する多機械スケジューリング問題として，順序制約を持つ作業から構成される仕事の集合，各作業を処理可能な機械の集合，および機械の取替時間が与えられるとき，すべての仕事を処理し，最大完了時刻（メイクスパン）を最小にする問題を扱う．3 つの特殊ケースに対し，最適なスケジュールを求める厳密解法を提案する．これらのケースは，生産計画やサプライチェーンなど実社会の様々な課題への応用に繋がる．

2.　問題説明

機械の集合 $M = \{1, 2, \ldots, i, \ldots, m\}$ と，機械の取替時間 $c_{i,i'}$ $(i, i' \in M)$，仕事の集合 $J = \{1, 2, \ldots, j, \ldots, n\}$，仕事 j $(\in J)$ を構成する作業の数 n_j，構成する順序付き作業の集合 $O_j = \{j_1, j_2, \ldots, j_k, \ldots, j_{n_j}\}$，作業 j_k $(j \in J, j_k \in O_j)$ を処理可能な機械の集合 $\mu_{j,k}$ $(\subseteq M)$ と処理時間 p_k^j が与えられる．

すべての仕事を処理し，最大完了時刻 $C_{\max} = \max_{j \in J} C_j$ が最小になるスケジュールを求める．

本研究では，仕事が 1 つで機械の数が任意であるケース（single-job m-machine scheduling problem，1-m-MSP），仕事と機械ともに 2 つであるケース（2-job 2-machine scheduling problem, 2-2-MSP），仕事が 3 つ，機械が 2 つであるケース（3-job 2-machine scheduling problem, 3-2-MSP）について検討する．2-2-MSP と 3-2-MSP については，作業を処理可能な機械の集合に制約を定める．

3.　1-m-MSP

仕事が 1 つであるから，他の仕事の影響を受けて待ち時間が発生することがなく，機械の取替時間が最小になるようなスケジュールを求めればよい．

1-m-MSP が最短路問題に帰着できることを示す．作業 1_k が機械 i $(\in M)$ で処理可能であることを頂点 $v_{k,i}$ で表す．ダミー頂点 $v_{0,0}, v_{n_1+1,0}$ を加え，頂点集合 V，辺集合 E，有向辺 $(v_{k,i}, v_{k+1,i'})$ の重み $w(v_{k,i}, v_{k+1,i'})$ を次のように定めて，グラフ $G = (V, E)$ を構築する:

$$V_k = \begin{cases} \{v_{k,0}\} & (k = 0, n_1 + 1), \\ \{v_{k,i} \mid i \in \mu_{1,k}\} & (1 \leq k \leq n_1), \end{cases}$$

$$V = \bigcup_{0 \leq k \leq n_1+1} V_k,$$

$$E = \{(v_{k,i}, v_{k+1,i'}) \mid 0 \leq k \leq n_1, \quad v_{k,i} \in V_k, v_{k+1,i'} \in V_{k+1}\},$$

$$w(v_{k,i}, v_{k+1,i'}) = \begin{cases} 0 & (k = 0), \\ p_k^1 + c_{i,i'} & (1 \leq k \leq n_1 - 1), \\ p_k^1 & (k = n_1). \end{cases}$$

仕事 1 の完了時刻 C_1 を最小にする問題の部分問題を「作業 1_k を機械 i' で処理するとき，処理開始時刻の最小化」とし，最適値を $g_k(i')$ とおく．C_1 の最小値は $g_{n_1+1}(0)$ となる．

頂点 $v_{0,0}$ を始点，頂点 $v_{n_1+1,0}$ を終点とするパスとスケジュールが一対一に対応する．初期値を $g_0(0) = 0$ と定めると，g_k の漸化式は

$$g_k(i') = \min\{g_{k-1}(i) + w(v_{k-1,i}, v_{k,i'}) \mid v_{k-1,i} \in V_{k-1}\} \quad (1 \leq k \leq n_1 + 1)$$

と定義できるため，k の小さい順に動的計画法を適用すれば，計算量は $\mathcal{O}(|E|) = O(n_1 m^2)$ である．

4. 2-2-MSP

機械 1 で仕事 1 の奇数番目の全作業，機械 2 で仕事 1 の偶数番目の全作業と仕事 2 の全作業を処理可能であるとき，各作業の処理時間に着目して最短路モデルに基づくアルゴリズムを提案する．

まず，仕事 1 の偶数番目の作業の前後の空き空間の数を $u = \lfloor n_1/2 \rfloor + 1$ とおくと，仕事 2 の作業は u 個の空き空間のいずれかで処理される．

仕事 2 の作業 $2_l, 2_{l+1}, \ldots, 2_{l'-1}$ が先頭から u' 番目の空き空間で処理されることを有向辺 $(v_{u'-1,l}, v_{u',l'})$ によって表現し，重みは

$$\max\left\{p^1_{2u'-1} + c(u'), \sum_{k=l}^{l'-1} p^2_k\right\} + p^1_{2u'},$$

$$c(u') = \begin{cases} 0 + c_{1,2} = c_{1,2} & (u' = 1), \\ 0 + 0 = 0 & (u' = u \wedge n_1 \text{ is even}), \\ c_{2,1} + 0 = c_{2,1} & (u' = u \wedge n_1 \text{ is odd}), \\ c_{2,1} + c_{1,2} & (\text{otherwise}) \end{cases}$$

とする．頂点集合，辺集合を次のように定義する:

$$\begin{aligned} V = & \{v_{u',k} \mid 1 \le u' \le u-1, 1 \le k \le n_2+1\} \\ & \cup \{v_{0,1}, v_{u,n_2+1}\}, \\ E = & \{(v_{u'-1,k}, v_{u',k'}) \mid 2 \le u' \le u-1, \\ & \qquad 1 \le k \le n_2+1, k \le k' \le n_2+1\} \\ & \cup \{(v_{0,1}, v_{1,k}) \mid 1 \le k \le n_2+1\} \\ & \cup \{(v_{u-1,k}, v_{u,n_2+1}) \mid 1 \le k \le n_2+1\}. \end{aligned}$$

頂点 $v_{0,1}$ を始点，頂点 v_{u,n_2+1} を終点とするパスがスケジュールと一対一に対応し，有向非巡回グラフの最短路問題に帰着するので，計算量は $\mathcal{O}(|E|) = \mathcal{O}(n_1 n_2^2)$ である．

5. 3-2-MSP

機械 1 で仕事 1 の全作業と仕事 3 の奇数番目の作業，機械 2 で仕事 2 の全作業と仕事 3 の偶数番目の作業が処理可能であるケースを考える．

部分問題を「k が偶数（resp. 奇数）のとき，仕事 1 の s 番目，仕事 2 の t 番目，仕事 3 の k 番目までの作業が処理され，機械 2（resp. 機械 1）で最後に処理されるのが 3_k である場合に，3_{k+1} の処理開始可能な最早時刻」とし，最適値を $f(s,t,k)$ とおく．$C_{\max}$ の最適値は $f(n_1, n_2, n_3+1)$ となる．境界条件と初期値を次のように定める:

$$\begin{aligned} & p^l_k = 0 \text{ if } k > n_l \text{ or } k \le 0, \quad l \in \{1,2\}, \\ & f(i,j,-1) = \sum_{j'=1}^{j} p^2_{j'}, \quad f(i,j,0) = \sum_{i'=1}^{i} p^1_{i'}, \\ & g(i,j,-1) = g(i,j,0) = 0. \end{aligned}$$

k が偶数（resp. 奇数）のとき，3_{k-1} の開始時刻より前に処理を終了する仕事 1（resp. 仕事 2）の最後の作業のインデックスで場合分けすることで，漸化式を次のように定義し，$f(s,t,k)$ を与えるインデックスを $g(s,t,k)$ として記録する:

$$f(s,t,k) =$$
$$\begin{cases} \displaystyle\min_{s' \in \{0,\ldots,s\}} \max\Big\{ f(s',t,k-1) + p^3_k + c(k), \\ \qquad f(s', g(s',t,k-1), k-2) + p^3_{k-1} + \displaystyle\sum_{s''=s'+1}^{s} p^1_{s''} \Big\} \\ \qquad\qquad (k \text{ is even}), \\ \displaystyle\min_{t' \in \{0,\ldots,t\}} \max\Big\{ f(s,t',k-1) + p^3_k + c(k), \\ \qquad f(g(s,t',k-1), t', k-2) + p^3_{k-1} + \displaystyle\sum_{t''=t'+1}^{t} p^2_{t''} \Big\} \\ \qquad\qquad (k \text{ is odd}), \end{cases}$$

$$c(k) = \begin{cases} 0 & (k \ge n_3), \\ c_{2,1} & (k < n_3 \text{ and } k \text{ is even}), \\ c_{1,2} & (k < n_3 \text{ and } k \text{ is odd}). \end{cases}$$

$f(s,t,k), g(s,t,k)$ は $\mathcal{O}(n_1+n_2)$ 時間で計算可能であるため $\mathcal{O}(n_1 n_2 n_3 (n_1+n_2))$ 時間で最適解が求まる．

6. おわりに

本研究では，機械の取替時間を考慮する多機械スケジューリング問題の機械の数と仕事の数に限りがある 3 つの特殊ケースについて，動的計画法を用いた厳密解法により，多項式時間で最適スケジュールが求められることを示した．

参考文献

[1] P. Brucker: Scheduling Algorithms, (Springer, 2007).