A-006

マルチスロット活動選択問題
Multislot Activity-Selection Problem

千葉 英史*　　軽野 義行†
Eishi Chiba　　Yoshiyuki Karuno

1 はじめに

活動選択問題での目的関数は，納期ちょうどに処理を完了するジョブ数の最大化である [5]．そこでは，1 機械の場合のみ扱っている．文献 [2] では，並列 m 機械の場合に拡張した．文献 [7] では並列 m 機械上で，重み付きジョブとセットアップタイムを考慮し，目的は納期ちょうどに完了するジョブの重み和の最大化である．これらの問題はすべて多項式時間で解くことができる．

上で述べた問題は，処理されなかったジョブを考慮していないが，実際には処理できなかったジョブは，次の機会（例えば，来週，来月など）に処理され，そこでも納期ちょうどに完了することが求められる．

本稿では，このような状況をマルチスロットという概念を用いて定式化する．一般には NP 困難であることを示し，セットアップタイムを考慮しない場合に，多項式時間で解けることを示す．

さらに，関連問題であるスロット数最小化に対して，妥当なセットアップタイム制約の下で，新しい近似アルゴリズムを与える．

2 問題定義

- $M_1, M_2, \ldots, M_m$: m 台の同種機械．
- $J = \{1, 2, \ldots, n\}$: n 個のジョブ集合．
- p_j (> 0): ジョブ $j \in J$ の処理時間．
- d_j $(\geq p_j)$: ジョブ $j \in J$ の納期．
- w_j (> 0): ジョブ $j \in J$ の重み．
- s_{jk} (≥ 0): 異なる 2 ジョブ $j, k \in J$ の順序対に対するセットアップタイム．すなわち，ジョブ j と k が同一機械上で，ジョブ j, k の順に連続して処理されるならば，ジョブ j の処理完了時刻とジョブ k の処理開始時刻の差が，少なくとも s_{jk} でなくてはならない．
- L $(\geqq \max_{j \in J}\{d_j\})$: タイムスロット長．

K 個の周期的なタイムスロットを考慮し，それぞれのタイムスロット中にジョブの納期が与えられる．すなわち，ジョブ j の納期は d_j, $L + d_j$, $2L + d_j$, ..., $(K-1)L + d_j$ である．ジョブ j を選択するとは，これらの納期中のどれか一つの時点でジョブ j の処理を完了することである．選択するジョブの部分集合を $F \subseteq J$ とすると，選択するジョブ $j \in F$ のスケジュールは写像 $S : j \to (M_j^S, C_j^S)$ と書くことができる．ここで，M_j^S はジョブ j が処理される機械であり，C_j^S はジョブ j の完了時刻である．すなわち，ジョブ j は M_j^S で区間 $[C_j^S - p_j, C_j^S)$ の間処理される．スケジュール S が，以下の条件を満たす時，実行可能であると言う．

- 各ジョブ $j \in F$ に対して，$C_j^S = r_j^S L + d_j$ を満たす非負整数 r_j^S が存在する．
- 異なる 2 ジョブ $j, k \in F$ の順序対に対して，ジョブ j の後にジョブ k が同じ機械で連続して処理されるならば，$C_k^S \geqq C_j^S + s_{jk} + p_k$ が成立つ．

各ジョブは m 台の機械上で並列に処理される．処理の中断は許されない．目的関数は，K タイムスロット中で，選択するジョブの重み和を最大にすることである．

マルチスロット活動選択問題（処理ジョブ数最大化）

入力: n 個のジョブ集合 J，機械台数 m，各ジョブ $j \in J$ の処理時間 p_j，各ジョブ $j \in J$ の納期 d_j，各ジョブ $j \in J$ の重み w_j，異なる 2 ジョブ $i, j \in J$ のセットアップタイム s_{jk}，タイムスロット長 L，タイムスロット数 K．

出力: 実行可能なスケジュール S．

目的関数: $\sum_{j \in F} w_j \to$ 最大．

定理 1 マルチスロット活動選択問題は，NP 困難である．(証明略)

関連問題として，すべてのジョブを選択するとき (すなわち，$F = J$)，次のような問題がある．

スロット数最小化

入力: n 個のジョブ集合 J，機械台数 m，各ジョブ $j \in J$ の処理時間 p_j，各ジョブ $j \in J$ の納期 d_j，異なる 2 ジョブ $i, j \in J$ のセットアップタイム s_{jk}，タイムスロット長 L．

出力: 実行可能なスケジュール S．

目的関数: $\max_{j \in J}\{r_j^S\} \to$ 最小．

先行研究から，スロット数最小化問題が NP 困難であること [1]，および近似不可能であること [4] が知られている．

3 処理ジョブ数最大化

セットアップタイムの存在のために，処理ジョブ数最大化問題が NP 困難であることが示される．本章では，セットアップタイムを考慮しない場合に，多項式時間アルゴリズムを与える．

3.1 ジョブの重みを考慮する場合

文献 [7] では，$K = 1$ とした場合（すなわち，タイムスロットを考慮しない）の手法を示した．そこでは，元の問題

*法政大学
†京都工芸繊維大学

をグラフ上の問題へと帰着し，ネットワークフローアルゴリズムを適用して，最適スケジュールを求める．この手法を利用したのが以下のアルゴリズムである．

アルゴリズム MAX_WEIGHTED_JOB

Step 1: mK 台の機械 ($\{N_{i,j} \mid i = 1, 2, \ldots, m, j = 1, 2, \ldots, K\}$) に対して，文献 [7] の手法を使ってスケジュールを求める．

Step 2: $N_{i,j}$ のスケジュールを，機械 M_i で j 番目のタイムスロットに割り当てる．

Step 1 で得られたスケジュールを利用して，Step 2 では，m 台の機械でのスケジュールを構成する．その際，セットアップタイムが 0 なので，任意のタイムスロットへの割り当てが可能である．また，Step 1 は多項式時間で解くことができるが [7]，その計算時間は $\Omega(n^4 \log n)$ である [2]．$n \leq mK$ のときは，全てのジョブを選択することができるので，$n > mK$ を仮定する．よって，全体でも多項式時間である．

3.2 ジョブの重みを考慮しない場合

ジョブの重みを考慮しない場合（すなわち，$w_j = 1$），文献 [2] の手法を適用することで高速化できる．そこでは，$K = 1$ の場合に納期ちょうどに処理を完了するジョブの個数を最大化する単純な貪欲法を与えた．これを利用すると，アルゴリズム MAX_JOB が得られる．すなわち，アルゴリズム MAX_WEIGHTED_JOB の Step 1 で，文献 [2] の手法を適用するように修正するだけでよい．文献 [2] の手法の計算時間は $O(n \log n + nmK)$ である．よって，$n > mK$ のときに，全体では $O(n^2)$ となる．

4 スロット数最小化

本章では，スロット数最小化問題に対して，自然なセットアップタイムの制約条件を付け加える．すなわち，任意の異なる 3 つのジョブ j, k, i に対して，三角不等式 $s_{jk} \leq s_{ji} + s_{ik}$ を満たす．この制約は文献 [7] で与えられた．

以下では，機械数が一つの場合の近似アルゴリズムを与える．ジョブ j と k のすべての順序対に対して，次の制約式を満たす非負整数 g_{jk} を導入する．

$$(g_{jk}-1)\cdot L + d_k < d_j + s_{jk} + p_k \leqq g_{jk}\cdot L + d_k. \quad (1)$$

この制約式は，ジョブ j, k が同じ機械上で，ジョブ j, k の順に連続して処理されるならば，ジョブ j を処理した後，g_{jk} タイムスロット先にはじめてジョブ k の処理を開始することができることを意味する．入力が与えられると，各 g_{jk} が一意に定まる．

補題 1 任意の異なる 3 ジョブ j, k, i に対して，三角不等式 $g_{jk} \leq g_{ji} + g_{ik}$ が成り立つ．(証明略)

次の重み付き有向グラフ $G = (V, A)$ を構成する．ここで，V は点集合，A は枝集合，重み関数を $w : A \to \mathbb{Z}$ とする．

- $V = s \cup t \cup \{v_i | i \in J\}$.
- A は以下の枝から成る．

  $w(s, v_j) = 0$ を付した (s, v_j) $(j \in J)$.

  $w(v_j, t) = 0$ を付した (v_j, t) $(j \in J)$.

  $w(v_j, v_k) = g_{jk}$ を付した (v_j, v_k) $(j \neq k \in J)$.

元の問題は，このグラフ上で全ての点をちょうど一度訪問する最小重み和の s-t パスを求める問題に帰着される．一般には，この問題は NP 困難であり [6]，近似アルゴリズムの研究も少ないが，枝重みが三角不等式を満たす一般の非対称グラフに対して，文献 [8] は近似比 $O(\sqrt{n})$ のアルゴリズムを与えた．文献 [3] では近似比 $O(\log n)$ に改善した．なお，得られた重み和に 1 を加えた値が，元問題のタイムスロット数になる．

s-t パスから次のようにして実行可能スケジュールを得る．s-t パスから，正の重みを持つ全ての枝を取り除くことによって，複数のパス $\pi_1, \pi_2, \cdots, \pi_l$ に分解し，以下の様にタイムスロットに割り当てる．

(i) π_1 を最初のタイムスロットに割り当てる．

(ii) π_j を k 番目のタイムスロットに割り当てると仮定する．この時，π_j と π_{j+1} を接続する枝の重みが k' であるならば，π_{j+1} を $(k + k')$ 番目のタイムスロットに割り当てる．

$O(\log n)$-近似アルゴリズム

Step 1: グラフ G を構成する．

Step 2: 文献 [3] で与えられた手法をサブルーチンとして実行する．

Step 3: 上で得られたパスから実行可能スケジュールを得る．

参考文献

[1] O. Čepek and S. C. Sung, Just in time scheduling with periodic time slots, Scientiae Mathematicae Japonicae, vol. 60, no. 2, pp. 295–301, 2004.

[2] O. Čepek and S. C. Sung, A quadratic time algorithm to maximize the number of just-in-time jobs on identical parallel machines, Computers & Operations Research, vol. 32, pp. 3265–3271, 2005.

[3] C. Chekuri and M. Pál, An $O(\log n)$ approximation ratio for the asymmetric traveling salesman path problem, Theory of Computing, vol. 3, pp. 197 – 209, 2007.

[4] E. Chiba and K. Hiraishi, A heuristic algorithm for one-machine just-in-time scheduling problem with periodic time slots, IEICE Trans., vol. E88-A, no. 5, pp. 1192 – 1199, 2005.

[5] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, Introduction to algorithms, Second edition, MIT Press, 2001.

[6] M. R. Garey and D. S. Johnson, Computers and intractability: A guide to the theory of NP-completeness, W. H. Freeman and Company, San Francisco, 1979.

[7] K. Hiraishi, E. Levner and M. Vlach, Scheduling of parallel identical machines to maximize the weighted number of just-in-time jobs, Computers & Operations Research, vol. 29, pp. 841–848, 2002.

[8] F. Lam and A. Newman, Traveling salesman path problems, Mathematical Programming, vol. 113, no. 1, pp. 39 – 59, 2008.