3プロセッサスケジューリングに関する一考察

4C－9

下岡 健一　大山口 通夫　太田 義勝
三重大学 工学部 情報工学科

1　はじめに

複数の命令を並列に実行できるマルチプロセッサシステムにおいて、その性能を最大限に発揮するためには、タスク (命令) スケジューリングが重要な役割を果たす。しかし、最適スケジューリング問題は一般にNP完全である [1]。従って、良い近似解を与える発見的アルゴリズムや、多項式時間で最適解を保証し広いクラスに適用可能な十分条件が求められている。

ここで、最適スケジューリング問題とは、各タスクを頂点$v \in V$としタスク間の依存関係を辺$e \in E$として記述した有向非循環グラフ (タスクグラフ)$G = (V, E)$ とマルチプロセッサの構成を与えて、最短時間ですべてのタスクの実行が終了するようなスケジュールを求める問題である。

本稿では、プロセッサ数が3の場合のスケジューリング問題について、多項式時間で最適解が求まるためのいくつかの十分条件を与える。

2　スケジュール対象の条件

本稿では、次の条件を仮定した最適スケジューリング問題について考察する。

条件: (1) プロセッサ数は3。(2) 各プロセッサは均質な機能を持つ。(3) すべてのタスクの実行時間は等しい。(4) タスク間の通信時間は考慮しない。

条件 (1) の代わりにプロセッサ数を 2 とした場合には、多項式時間最適スケジューリングアルゴリズムが知られている [2][3]。しかし、プロセッサ数を一般にk(kは変数) とした場合には、最適スケジューリング問題はNP完全であり [1]、また、プロセッサ数を 3 以上の定数とした場合には、最適スケジューリング問題がNP完全となるかどうかは現在未解決である [4]。

なお、プロセッサ数が3のとき、各タスクの親が高々3個であり、タスクグラフにおいて高さが等しい頂点が常に5個以上存在する場合に最適スケジュールを与える線形時間アルゴリズムが知られている [5]。ここで、頂点の高さとは、その頂点から葉に至るパスの最大長で定義される。

3　結果

本研究で得られた結果を以下に示す。

定理 1 タスクグラフ $G = (V, E)$ が高さ *2* 以下ならば、多項式時間で最適スケジューリング問題を解くことができる。ここで、タスクグラフ G の高さとは、根から葉に至るパスの最大長で定義される。 ■

定理 2 タスクグラフ $G = G_1(V_1, E_1) + G_2(V_2, E_2) + G_3(V_3, E_3)$(*3*個のグラフの直和) で、$G_1, G_2$に対する スケジュールが既に求まっているとき、次の条件 *(1),(2)* を満たせば G の最適スケジュールは線形時間で計算できる (アルゴリズム$\boldsymbol{A_1}$)。

(1) G_1, G_2の各スケジュールにおいて、空きプロセッサの存在する時間がそれぞれ $|V_3| + 1$ 以下である。

(2) $|V_1| \geq |V_3|$ かつ $|V_2| \geq |V_3|$ □

<u>**アルゴリズム$\boldsymbol{A_1}$**:</u> $A'_1(0, |V_1|, |V_2|, |V_3|)$ □

<u>**サブアルゴリズム$\boldsymbol{A'_1(t, n_1, n_2, n_3)}$**:</u>

1. $n_1 \geq n_3 + 3$(または $n_2 \geq n_3 + 3$) かつ時刻 t に *3* 個割当可能なタスクが G_1(または G_2) に存在するとき、時刻 t にそのタスクの集合 U を割当て、$A'_1(t+1, n_1 - 3, n_2, n_3)$ (または $A'_1(t+1, n_1, n_2 - 3, n_3)$)
2. *1*でなく、$n_3 > 0$ のとき、
   - $n_1 \geq n_3 + 2$(または $n_2 \geq n_3 + 2$) かつ時刻 t に *2* 個割当可能なタスク (その集合 U) が G_1(または G_2) に存在するとき、

A Study of Optimal Scheduling for 3-Processor Systems
Ken-ichi Shimooka, Michio Oyamaguchi, Yoshikatsu Ohta
Department of Information Engineering, Faculty of Engineering, Mie University
1515 Kamihama-cho, Tsu-shi, Mie-ken, 514-8507 Japan

- $n_2 > n_3$(または $n_1 > n_3$)のとき、時刻 t に U と G_2(または G_1)で割当可能なタスク 1 個を割当て、$A'_1(t+1, n_1-2, n_2-1, n_3)$(または $A'_1(t+1, n_1-1, n_2-2, n_3)$)
- $n_2 = n_3$(または $n_1 = n_3$)のとき、時刻 t に U と G_3で割当可能なタスク 1 個を割当て、$A'_1(t+1, n_1-2, n_2, n_3-1)$ (または $A'_1(t+1, n_1, n_2-2, n_3-1)$)

• それ以外のとき、時刻 t に G_1, G_2, G_3から 1 個ずつ割当可能なタスクを割当て、$A'_1(t+1, n_1-1, n_2-1, n_3-1)$

3. 1でなく、$n_3 = 0$ のとき ($i = 1, 2$ に対して、G_iのスケジュールにおいて空きプロセッサの存在する時刻は 1 以下)、G_1, G_2の時刻 t に割当可能なタスクの集合を U_1, U_2とする。($|U_1| \leq 2$ かつ $|U_2| \leq 2$)

• $|U_1| = |U_2| = 0$ のとき、終了。
• $|U_1| + |U_2| \leq 3$ のとき、時刻 t に U_1, U_2を割当て、$A'_1(t+1, n_1-|U_1|, n_2-|U_2|, n_3)$
• $|U_1| = |U_2| = 2$ のとき、時刻 $t, t+1$ に U_1, U_2を割当て、$A'_1(t+2, n_1-2, n_2-2, n_3)$ ■

定理 3 タスクグラフ $G = G_1 + G_2$(2 個のグラフの直和)で、G_1, G_2に対する最適なスケジュールが既に求まっているとき、次の条件 (1) を満たせば G の最適スケジュールは線形時間で計算できる。(アルゴリズム A_2)
(1) G_1, G_2の各スケジュールにおいて、各時刻における空きプロセッサ数は高々 1 である。 □

アルゴリズム A_2: $A'_2(T_1 - 1, 0, \mu_1, \mu_2)$
ここで $i = 1, 2$ に対して、G_iのスケジュールの実行時間を T_i、空きプロセッサの存在する時間を μ_i、時刻 t に割り当てられているタスクの集合を S_{it}とする。 □

サブアルゴリズム $A'_2(p_1, p_2, \mu_1, \mu_2)$:

1. $p_1 < 0$(または $p_2 > T_2 - 1$)のとき、G_2(または G_1)のスケジュールを G のスケジュールとする。
2. 1でなく、$\mu_1 + \mu_2 \leq 2$ のとき、G_1のスケジュールの後に G_2のスケジュール を並べ、それを G のスケジュールとする。
3. 2でなく、$|S_{2p_2}| = 3$ かつ $\mu_2 \leq 2\mu_1 - 4$ (または $|S_{2p_2}| = 2$ かつ $\mu_2 \leq 2\mu_1 - 1$)のとき、S_{2p_2}のタスクを G_1のスケジュールの空きプロセッサに早い時刻から移動させ、$A'_2(p_1, p_2+1, \mu_1-3, \mu_2)$ (または $A'_2(p_1, p_2+1, \mu_1-2, \mu_2-1)$)
4. 3でなく、$|S_{1p_1}| = 3$ かつ $\mu_1 \leq 2\mu_2 - 4$ (または $|S_{1p_1}| = 2$ かつ $\mu_1 \leq 2\mu_2 - 1$)のとき、S_{1p_1}のタスクを G_2のスケジュールの空きプロセッサに遅い時刻から移動させ、$A'_2(p_1-1, p_2, \mu_1, \mu_2-3)$ (または $A'_2(p_1-1, p_2, \mu_1-1, \mu_2-2)$)
5. 4でなく ($(\mu_1, \mu_2) = (1,2), (2,1), (2,2), (3,3)$) かつ、$\mu_1 \geq \mu_2$(または $\mu_1 < \mu_2$)のとき、S_{1p_1}(または S_{2p_2})のタスクを μ_2(または μ_1)個、G_2(または G_1) の空きプロセッサに移動させ、S_{2p_2}(または S_{1p_1})のタスクをすべて G_1 (または G_2) の空きプロセッサに移動させ、G_1のスケジュール の後に G_2のスケジュール を並べ、それを G のスケジュールとする。 ■

定理 3は、プロセッサ数を kにした場合にも成り立つ。

定理 4 タスクグラフ $G = G_1 + G_2$(2 個のグラフの直和)で、G_1に対する最適なスケジュールが既に求まっていて、G_2の高さが 1 以下ならば、G の最適スケジュールは線形時間で計算できる。 ■

4 おわりに

本稿ではプロセッサ数が 3 の場合において、タスクグラフが部分グラフの直和の形で表されるとき、それぞれの部分グラフの最適スケジュールが求まっている場合に、全体のタスクグラフに対する最適スケジュールが線形時間で計算できるための十分条件を示した。また、高さ 2 以下のタスクグラフに対して、最適スケジュールが多項式時間で計算できることを示した。

これらの結果をプロセッサ数 $k (\geq 3)$ の一般の場合に拡張することは今後の課題である。

参考文献

[1] J.D.Ullman, "NP-Complete Scheduling Problems", JCSS. 10, pp.384-393, 1975.

[2] M.Fujii, T.Kasami, K.Ninomiya, "Optimal sequencing of two equivalent processors", SIAM J. Appl. Math. 17, pp.784-789, 1969.

[3] E.G.Coffman, R.L.Graham, "Optimal Scheduling for Two-Processor Systems", Acta. Inf. 1, pp.200-213, 1972.

[4] M.R.Garey, D.S.Johnson, "COMPUTERS AND INTRACTABILITY A Guide to the Theory of NP-Completeness", p.287, 1991.

[5] 松原, 大山口, 太田, "マルチプロセッサ向き目的コードスケジューリングについて", 信学技報 COMP98-13, pp.33-40, 1998.