

タスクスケジューリングに関する 新しい近似アルゴリズムについて

A New Approximation Algorithm for Task Scheduling

片柳賢二 砂坂明宏 大山口通夫 太田義勝
(三重大学 工学部)

Kenji KATAYANAGI, Akihiro SUNASAKA, Michio OYAMAGUCHI and Yoshikatsu OHTA
(Faculty of Engineering, Mie University)

概要

Papadimitriou ら (1990) は、タスク複製を許す通信遅延のあるスケジューリング問題が NP 完全であることを示すとともに、近似精度 2 のアルゴリズムを与えた。坂上ら (2001) は、最適解の下界として Papadimitriou らが提案した e-value を用い、与えられたタスクの依存関係を表す DAG の高さを $L(L \geq 3)$ としたときに、近似精度 $2 - \frac{1}{3.2^{L-3}}$ を持つ近似アルゴリズムを示した。本研究では、e-value による下界よりも良い、新たな下界 $low(v)$ を定義すると共に、近似精度を $2 - \frac{1}{3c}$ に改善したアルゴリズムが存在することを示す。ここで、 $c = \lfloor \frac{low(v)}{\tau+1} \rfloor$ (τ は通信遅延時間) であり、 $c \leq L$ を満たす。

1 はじめに

並列処理システムでは、タスクのプロセッサ上での実際の処理時間の他に、タスクを異なるプロセッサに割り当てることで、通信遅延時間が発生する。システムの性能を最大限に発揮し、タスク全体の処理時間の短縮を図るためには、このような通信遅延を考慮に入れた良いタスクスケジューリングアルゴリズムが必要である。しかし、通信遅延を考慮したタスクスケジューリング問題は一般に NP 完全 [1] であり、従って、効率の良い近似アルゴリズムが求められている。

[1] では、タスクの複製を許したときに、この問題の NP 完全性と、近似精度 2 を持つ近似アルゴリズムを示しているが、2 未満の近似精度を持つアルゴリズムが存在するかどうかを未解決問題としていた。ここで、近似アルゴリズムの近似精度を $\frac{2}{\text{最適解}}$ と定義する。[2] では、タスクの DAG の高さを 2 に限定してもこの問題が NP 完全であることと、高さ 2 のとき近似精度 $\frac{3}{2}$ を持つアルゴリズムを示している。さらに、[1] で提案された、最適解の下界を与える e-value を用いる限り、近似精度を $\frac{3}{2}$ 以下に改善することは出来ないことが示された。また、[3] では、与えられた DAG の高さを $L(L \geq 3)$ としたときに、近似精度 $2 - \frac{1}{3.2^{L-3}}$ を持つ近似アルゴリズムを示している。

本研究では [1] ~ [3] と同じ条件の下で、タスクの DAG から再構成される二部グラフの最大マッチングの辺の本数に着目することにより、従来の e-value を改善した、最適解の下界 $low(v)$ を新たに定義すると共に、[3] の結果を改善した、近似精度 $2 - \frac{1}{3c}$ を持つ近似アルゴリズムを示す。ここで、 $c = \lfloor \frac{low(v)}{\tau+1} \rfloor$ (τ は通信遅延

時間) であり、 $c \leq L$ を満たす。

2 準備

有向非循環グラフ (DAG) $G = (V, E)$ に対して、 $(u, v) \in E$ となる頂点 u が存在しないとき、 v を G の根という。 $v \in V$ のレベルとは G の根から v への最大パス長で、 $level(v)$ と表記し、DAG の高さを $\max\{level(u) | u \in V\}$ と定義する。本研究では、

- (1) 等しい機能を持つプロセッサを無限個使用できる
- (2) 各タスクの実行時間は 1
- (3) 各プロセッサ間の通信遅延時間は整数 $\tau(\tau \geq 1)$
- (4) タスクの複製を許す

という [1] ~ [3] と同じ条件を前提としている。

この前提の下で、高さ L の DAG $G = (V, E)$ をスケジューリングの対象とする。ここで、 V はタスクの集合、 E はタスク間の依存関係を表す有向辺の集合である。このとき (1)(4) より、レベル L の頂点はただ一つとしても一般性を失わず、その頂点を v^* とする。

定義 1 $v \in V$ の最適スケジュール時刻を $opt(v)$ と表記する。

定義 2 ([2] [3]) $v \in V$ のスケジュール時刻の下界 e-value $e(v)$ を次のように定義する。

- (1) v が根であるとき、 $e(v) = 0$ とする。
- (2) v が根でないとき、 v のすべての先祖 u を $e(u)$ の降順にソートし、 $e(u_1) \geq e(u_2) \geq \dots \geq e(u_{p(v)})$ とする。ここで $p(v)$ は、 v の先祖の数。 $k(v) = \min(p(v), \tau + 1)$ として、 $e(v) = \max_{1 \leq i \leq k(v)} (e(u_i) + i)$ とする。

$e(v)$ がスケジュール時刻の下界、即ち、 $e(v) \leq opt(v)$ であることは [1] の証明と同様に証明できる ([2] [3] 参照)。

定義 3 タスク u の先祖の集合 $pred(u)$ を次のように定義する。

$$pred(u) = \begin{cases} \phi & (level(u) = 0 \text{ の場合}) \\ \bigcup_{(u', u) \in E} (pred(u') \cup \{u'\}) & (\text{その他の場合}) \end{cases}$$

また、タスクの集合 S に対し、 $pred(S) = \bigcup_{u \in S} pred(u)$ とする。

定義 4 $U_1, U_2 \subseteq V \setminus \{v^*\}$ かつ $U_1 \cap U_2 = \phi$ とする。そのとき、二部グラフ $f_G(U_2 \times U_1) = (V', E')$ を次のように定義する。

$$V' = U_1 \cup U_2, E' = \{(u, u') \in U_2 \times U_1 | u \in pred(u')\}$$

3 下界 $low(v)$

$v \in V$ に対して、本研究で提案する新しい下界 $low(v)$ を以下のように定義する。 $(p(v), k(v))$ については定義 2 を参照)

- $p(v) \leq \tau + 1$ のとき、 $low(v) = e(v)$ ($= p(v)$)
- $p(v) > \tau + 1$ のとき、 $k(v) = \tau + 1$ で、
 - $k(v) \leq e(v) < \frac{3}{2}k(v)$ のとき、 $low(v)$ は次節で述べるスケジューリングアルゴリズムから求められ、 $e(v) \leq low(v) \leq e(v) + \frac{1}{2}k(v)$ を満たす。
 - $\frac{3}{2}k(v) \leq e(v) < 2k(v)$ のとき、 $low(v) = e(v)$
 - $e(v) \geq 2k(v)$ のとき、 v のすべての先祖 u を $low(u)$ の降順にソートし、 $low(u_1) \geq low(u_2) \geq \dots \geq low(u_p(v))$ として、 $low(v) = \max_{1 \leq i \leq k} (low(u_i) + i)$ とする。

$k(v) \leq e(v) < \frac{3}{2}k(v)$ のとき、 $low(v)$ は v の最適スケジュール時刻の下界であることを次節で示す。従って、 $low(v) \leq opt(v)$ であることは、[1] の証明と同様に証明できる。 $e(v) \leq low(v)$ より、 $low(v)$ は $e(v)$ より良い下界を与える。 $e(v)$ を下界として用いた場合の近似精度の限界は $\frac{3}{2}$ である ([3]) が、この下界を用いることで高さ 2 の DAG に対し $\frac{4}{3} + \epsilon$ (但し、 ϵ は任意の正の値) の近似精度を持つスケジューリングが可能になることから、 $e(v)$ よりも真に良い下界を与えるものである。

4 近似アルゴリズム

本節では近似精度 $2 - \frac{1}{3\epsilon}$ の近似アルゴリズムを示す。

タスク v^* をスケジューリングするプロセッサをメインプロセッサ、それ以外のプロセッサを外部プロセッサとよぶ。以下では、タスクの集合 X をプロセッサにスケジューリングするとは、 X の全ての要素を下界 low の値の小さい順にプロセッサにスケジューリングすることを表す。

補題 1 [1]

(1) $p(v^*) \leq \tau + 1$ ならば、 v^* は時刻 $p(v^*)$ でスケジューリングでき、最適スケジュールとなる。

(2) $e(v^*) < \tau + 1$ ならば、 v^* は時刻 $e(v^*)$ でスケジューリングできる。

以下では、 $p(v^*) > \tau + 1$ の場合について述べる。このとき $k(v^*) = \tau + 1$ である。また $opt(v^*) \geq k(v^*)$ である。また、 $v^*, p(v^*), k(v^*), opt(v^*)$ を以下では単に v, p, k, opt と表記する。 v のすべての先祖を e -value の降順にソートしたものを u_1, u_2, \dots, u_p とする。

補題 2 $k \leq e(v) < 1.5k$ のとき、すべての i ($0 \leq i \leq 0.45k$) とすべての x ($(\frac{2}{3} - 0.01)k + \frac{2}{3}i \leq x \leq k - [0.01k] \wedge x \in N$) (N : 自然数の集合) に対して、 $opt \geq k + i$ かつ $e(u_x) \geq 0.01k + i$ ならば、次の (i)~(iii) のいずれかが成立する。

- (i) $e(u_{\lfloor x+0.01k \rfloor}) \geq 0.01k + i$
- (ii) $opt \geq 1.01k + i$
- (iii) 近似精度 $\frac{5}{3}$ でスケジューリング可能。(このとき、 $low(v) = \max(e(v), k + i)$ と定義する。)

証明) $y = \lfloor x + 0.01k \rfloor$ とする。(i) が成立しない場合、即ち $e(u_y) < 0.01k + i$ のときに、(ii) または (iii) が成立することを示す。

$$U_1 = \{u_1, \dots, u_y\}, U_2 = \{u_{y+1}, \dots, u_p\}$$

M を $G' = f_G(U_2 \times U_1)$ の最大マッチングとする。(図 1 参照)

$$M_p = \{u|(u, u') \in M\}, M_c = \{u'|(u, u') \in M\}$$

$$|M| = m, B = U_1 \setminus M_c$$

すべての $u \in pred(B)$ に対して $u \notin M_c$ となるように、必要なら B と M_c の要素を入れ換えることにより、 $pred(B) \subseteq M_p \cup B$ とする。

N を $G'' = f_G((U_2 \setminus M_p) \times M_c)$ の最大マッチングとする。(図 1 参照)

$$N_p = \{u|(u, u') \in N\}, N_c = \{u'|(u, u') \in N\}$$

$$|N| = n, C = M_c \setminus N_c$$

すべての $u \in pred(C)$ に対して $u \notin N_c$ となるように、必要なら C と N_c の要素を入れ換えることにより、 $pred(C) \subseteq M_p \cup N_p \cup B \cup C$ とする。

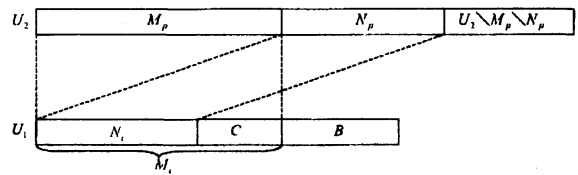


図 1: 最大マッチング

- $m \leq (\frac{2}{3} - 0.01)k + \frac{2}{3}i$ のとき、補題 1-2 より u_y 以降のタスクは時刻 $e(u_y)$ でスケジューリングできるので、 u_{y+1} 以降のタスクを外部プロセッサにスケジューリングし、メインプロセッサには時刻 0 から $pred(B) \cup B$ をスケジューリングし、通信を待って M_c をスケジューリングすれば、 $|pred(B) \cup B| \leq e(u_y) + k$ であることから、 v のスケジュール時刻の上界は $e(u_y) + k + m \leq \frac{5}{3}opt$ となり (図 2)、(iii) が成立する。

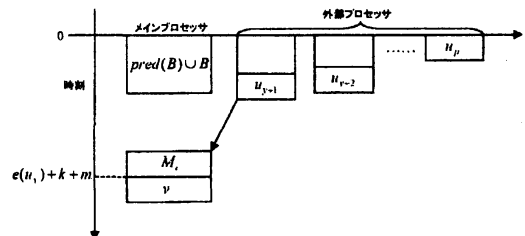


図 2: $m \leq (\frac{2}{3} - 0.01)k + \frac{2}{3}i$ の場合のスケジュール

- $m > (\frac{2}{3} - 0.01)k + \frac{2}{3}i$ のとき、
 - $m - [0.02k + i] + x \geq 1.01k + i$ ならば、どのようにスケジュールしたとしても v のスケジュール時刻が $1.01k + i$ 以降になることを以下に示す。即ち、(ii) が成立する。

- * $\{u_1, \dots, u_x\}$ のうち 1 個以上のタスクを外部プロセッサにスケジュールすると、 v のスケジュール時刻は $e(u_x) + k > 1.01k + i$ 以降となる。
- * $\{u_1, \dots, u_x\}$ を全てメインプロセッサにスケジュールし、
 - M_p のうち $[0.02k + i] + 1$ 個以上のタスクを外部プロセッサにスケジュールすると、マッチングにおいて対応している $M_c \cap \{u_1, \dots, u_x\}$ の要素を $[0.01k + i] + 1$ 個以上通信を待つてメインプロセッサにスケジュールすることになり、 v のスケジュール時刻は $1.01k + i$ 以降となる。(図 3 参照)

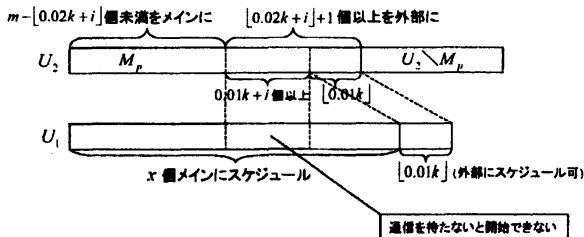


図 3: M_p のうち $[0.02k + i] + 1$ 個以上のタスクを外部プロセッサにスケジュールする場合

- M_p のうち $m - [0.02k + i]$ 個以上のタスクをメインプロセッサにスケジュールすると、メインプロセッサにスケジュールするタスクの個数が $m - [0.02k + i] + x$ 以上となり、 v のスケジュール時刻は $1.01k + i$ 以降となる。
- $m - [0.02k + i] + x < 1.01k + i$ 、即ち $m + x < 1.01k + i + [0.02k + i]$ のとき、 $2\{(\frac{2}{3} - 0.01)k + \frac{2}{3}i\} < m + x < 1.03k + 2i$ 。従って、 $i > 0.425k$ であり、 $m + y > 2\{(\frac{2}{3} - 0.01)k + \frac{2}{3}i\} + [0.01k] > 1.88k$ と $k + e(u_y) < 1.01k + i \leq 1.46k$ より、 $k + e(u_y) < m + y$ である。また、補題 1-2 より u_y 以降のタスクは時刻 $e(u_y)$ でスケジュールできるので、 u_{y+1} 以降のタスクを外部プロセッサにスケジュールし、メインプロセッサには時刻 0 から $\text{pred}(B) \cup B, (\text{pred}(C) \cup C) \setminus (\text{pred}(B) \cup B)$ を順にスケジュールし、通信を待つて N_c をスケジュールすれば、 v のスケジュール時刻の上界は $y + m + n$ となり (図 4)、

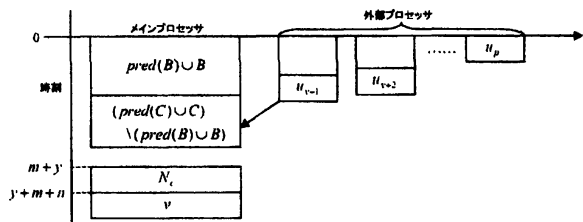


図 4: $m - [0.02k + i] + x < 1.01k + i$ の場合に可能なスケジュール

- * $n \leq (\frac{2}{3} - 0.04)k - \frac{1}{3}i$ のとき、 $m + y < 1.04k + 2i$ より、 $y + m + n \leq \frac{5}{3}(k + i) \leq \frac{5}{3}opt$ となるので、(iii) が成立する。

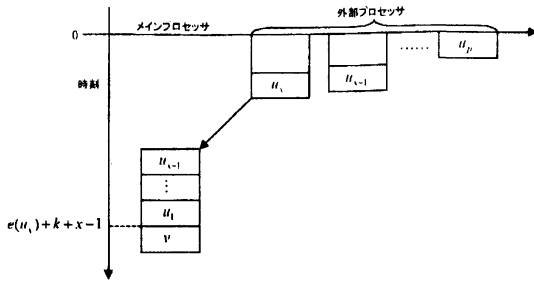
- * $n > (\frac{2}{3} - 0.04)k - \frac{1}{3}i$ のとき、
 - $m + n - 2[0.02k + i] + x < 1.01k + i$ のとき、 $y + m + n < 1.06k + 3i = \frac{5}{3}(k + i) + (1.06 - \frac{5}{3})k + \frac{4}{3}i \leq \frac{5}{3}(k + i) + (1.06 - \frac{5}{3} + \frac{4}{3} \cdot 0.45)k = \frac{5}{3}(k + i) - \frac{1}{150}k < \frac{5}{3}opt$ であり、(iii) が成立する。
 - $m + n - 2[0.02k + i] + x \geq 1.01k + i$ のとき、どのようにスケジュールしたとしても v のスケジュール時刻が $1.01k + i$ 以降になることを以下に示す。即ち (ii) が成立する。
 - $\{u_1, \dots, u_x\}$ のうち 1 個以上のタスクを外部プロセッサにスケジュールすると、 v のスケジュール時刻は $e(u_x) + k > 1.01k + i$ 以降となる。
 - $\{u_1, \dots, u_x\}$ を全てメインプロセッサにスケジュールし、
 - M_p のうち $[0.02k + i] + 1$ 個以上、または N_p のうち $[0.02k + i] + 1$ 個以上のタスクを外部プロセッサにスケジュールすると、マッチングにおいて対応している M_c または N_c の要素を $[0.01k + i] + 1$ 個以上通信を待つてメインプロセッサにスケジュールすることになり、 v のスケジュール時刻は $1.01k + i$ 以降となる。
 - M_p, N_p からそれぞれ $m - [0.02k + i]$ 個以上、 $n - [0.02k + i]$ 個以上のタスクをメインプロセッサにスケジュールすると、メインプロセッサにスケジュールするタスクの個数が $m + n - 2[0.02k + i] + x \geq 1.01k + i$ 以上となり、 v のスケジュール時刻は $1.01k + i$ 以降となる。□

補題 3 $k \geq 100$, $k \leq e(v) < 1.5k$ かつ $opt \geq k$ のとき、以下のいずれかが成立する。

- $opt \geq 1.46k$
- 近似精度 $\frac{5}{3}$ でスケジュール可能。

証明) 補題 2 より、 $i = 0$, $x = \lceil (\frac{2}{3} - 0.01)k \rceil$ のとき、

- $e(u_x) < 0.01k$ のとき、補題 1-2 より u_x 以降のタスクは時刻 $e(u_x)$ でスケジュールできるので、 u_x 以降のタスクを外部プロセッサにスケジュールし、通信を待つて残りの $x - 1$ 個のタスクをメインプロセッサにスケジュールすると v のスケジュール時刻の上界は $e(u_x) + k + x - 1 < 1.01k + (\frac{2}{3}k - 0.01)k = \frac{5}{3}k \leq \frac{5}{3}opt$ となり、近似精度 $\frac{5}{3}$ のスケジュールとなる。(図 5)
- $e(u_x) \geq 0.01k$ のとき、補題 2 の (i) ~ (iii) のいずれかが成立する。(i) が成立する場合には、 x の値を $[0.01k]$ 増やして補題 2 を繰り返し適用することにより、 $e(u_k) \geq 0.01k$, (ii), (iii) のいずれかが成立する。 $e(u_k) \geq 0.01k$ のときは、どのようにス

図 5: $e(u_x) < 0.01k + i$ の場合のスケジュール

スケジュールしたとしても v のスケジュール時刻が $1.01k$ 以降になることを以下に示す。即ち、(ii) が成立する。

- $\{u_1, \dots, u_k\}$ のうち 1 個以上のタスクを外部プロセッサにスケジュールすると、 v のスケジュール時刻は $e(u_k) + k \geq 1.01k$ 以降となる。
- $\{u_1, \dots, u_k\}$ を全てメインプロセッサにスケジュールすると、時刻 $0.01k$ 以降にメインプロセッサにスケジュールするタスクの個数が k 以上になるので、 v のスケジュール時刻は $1.01k$ 以降となる。

よって、 $e(u_x) \geq 0.01k$ かつ $opt \geq 1.01k$ のときが残る。

一般に、 $opt(v) \geq k + i$, $x = \lceil (\frac{2}{3} - 0.01)k + \frac{2}{3}i \rceil$ とする。但し、 $0.01k \leq i \leq 0.45k$ 。

- $e(u_x) \leq i + 0.01k$ のとき、補題 1-2 より u_x 以降のタスクは時刻 $e(u_x)$ でスケジュールできるので、 u_x 以降のタスクを外部プロセッサにスケジュールし、通信を待って残りの $x-1$ 個のタスクをメインプロセッサにスケジュールすると v のスケジュール時刻の上界は $k + e(u_x) + x - 1 \leq k + i + 0.01k + (\frac{2}{3} - 0.01)k + \frac{2}{3}i = \frac{5}{3}(k + i)$ となり、近似精度 $\frac{5}{3}$ のスケジュールとなる。(図 5)
- $e(u_x) > i + 0.01k$ のとき、補題 2 の (i) ~ (iii) のいずれかが成立する。(i) が成立する場合には、 x の値を $\lfloor 0.01k \rfloor$ 増やして補題 2 を繰り返し適用することにより、 $e(u_k) \geq 0.01k + i$, (ii), (iii) のいずれかが成立する。 $e(u_k) \geq 0.01k + i$ のときは、どのようにスケジュールしたとしても v のスケジュール時刻が $1.01k + i$ 以降になるので (ii) が成立する。

よって、 $i = 0.45k$ のとき、 $opt \geq 1.46k$ または 近似精度 $\frac{5}{3}$ のスケジュールが得られる。□

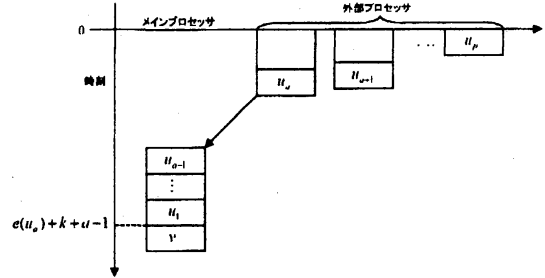
補題 4 $k \geq 100$, $k \leq e(v) < 1.5k$ かつ $opt \geq 1.46k$ のとき、以下のいずれかが成立する。

- $opt \geq 1.5k$ (このとき、 $low(v) = 1.5k$ と定義する。)
- 近似精度 $\frac{5}{3}$ でスケジュール可能。(このとき、 $low(v) = \max(e(v), 1.46k)$ と定義する。)

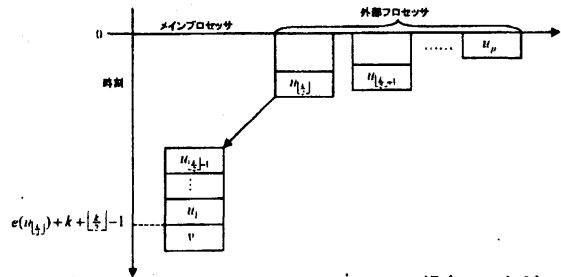
証明) $a = \lceil 0.93k \rceil$ とする。 $a + e(u_a) \leq e(v) < 1.5k$ より、 $e(u_a) < 0.56k$ である。

- $e(u_a) \leq 0.5k$ ならば、補題 1-2 より u_a 以降のタスクは時刻 $e(u_a)$ でスケジュールできるので、 u_a

以降のタスクを外部プロセッサにスケジュールし、通信を待って残りの $a-1$ 個のタスクをメインプロセッサにスケジュールすると v のスケジュール時刻の上界は $e(u_a) + k + a - 1 \leq 2.43k \leq \frac{5}{3}opt$ となり、近似精度 $\frac{5}{3}$ のスケジュールが得られる。(図 6)

図 6: $e(u_a) \leq 0.5k$ の場合のスケジュール

- $e(u_a) > 0.5k$ かつ $e(u_{\lfloor \frac{k}{2} \rfloor}) \leq 0.93k + 1$ ならば、補題 1-2 より $u_{\lfloor \frac{k}{2} \rfloor}$ 以降のタスクは時刻 $e(u_{\lfloor \frac{k}{2} \rfloor})$ でスケジュールできるので、 $u_{\lfloor \frac{k}{2} \rfloor}$ 以降のタスクを外部プロセッサにスケジュールし、通信を待って残りの $\lfloor \frac{k}{2} \rfloor - 1$ 個のタスクをメインプロセッサにスケジュールすると、 v のスケジュール時刻の上界は $e(u_{\lfloor \frac{k}{2} \rfloor}) + k + \lfloor \frac{k}{2} \rfloor - 1 \leq 2.43k \leq \frac{5}{3}opt$ となり、近似精度 $\frac{5}{3}$ のスケジュールが得られる。(図 7)

図 7: $e(u_a) > 0.5k$ かつ $e(u_{\lfloor \frac{k}{2} \rfloor}) \leq 0.93k + 1$ の場合のスケジュール

- $e(u_a) > 0.5k$ かつ $e(u_{\lfloor \frac{k}{2} \rfloor}) > 0.93k + 1$ のとき、

$$U_1 = \{u_{\lfloor \frac{k}{2} \rfloor + 1}, \dots, u_a\}, U_2 = \{u_{a+1}, \dots, u_p\}$$

M を $G' = f_G(U_2 \times U_1)$ の最大マッチングとする。(図 1 参照)

$$M_p = \{u|(u, u') \in M\}, M_c = \{u'|(u, u') \in M\}$$

$$|M| = m, B = U_1 \setminus M_c$$

すべての $u \in \text{pred}(B)$ に対して $u \notin M_c$ となるように、 B と M_c の要素を入れ換えることにより、 $\text{pred}(B) \subseteq M_p \cup B$ とする。 N を $G'' = f_G((U_2 \setminus M_p) \times M_c)$ の最大マッチングとする。(図 1 参照)

$$N_p = \{u|(u, u') \in N\}, N_c = \{u'|(u, u') \in N\}$$

$$|N| = n, C = M_c \setminus N_c$$

すべての $u \in \text{pred}(C)$ に対して $u \notin N_c$ となるように、 C と N_c の要素を入れ換えることにより、 $\text{pred}(C) \subseteq M_p \cup N_p \cup B \cup C$ とする。

$m + |B| = |U_1| = a - \lfloor \frac{k}{2} \rfloor$ である。ここで $|B| = (a - \lfloor \frac{k}{2} \rfloor) - m$, $|B| + |C| = (a - \lfloor \frac{k}{2} \rfloor) - n$ が成り立つ。

- $|B| \geq a - 0.86k$ ならば、補題 1-2 より u_a 以降のタスクは時刻 $e(u_a)$ でスケジュールできるので、 u_{a+1} 以降のタスクを外部プロセッサにスケジュールし、メインプロセッサには時刻 0 から $pred(B) \cup B$ をスケジュールし、通信を待って残り $a - |B|$ 個のタスクをスケジュールすれば、 $|pred(B) \cup B| \leq e(u_a) + k$ であることから、 v のスケジュール時刻の上界は $e(u_a) + k + a - |B| \leq 2.43k \leq \frac{5}{3}opt$ となり、近似精度 $\frac{5}{3}$ のスケジュールとなる。(図 8)

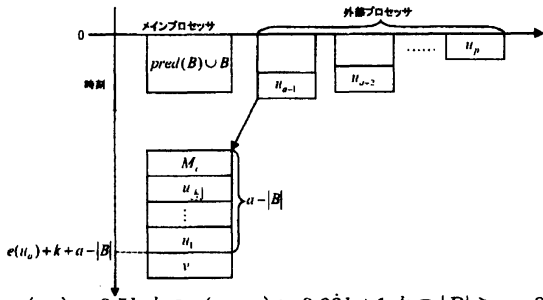


図 8: $e(u_a) > 0.5k$ かつ $e(u_{\lfloor \frac{k}{2} \rfloor}) > 0.93k + 1$ かつ $|B| \geq a - 0.86k$ の場合のスケジュール

- $|B| < a - 0.86k$ かつ $|B| + |C| \geq a - 0.86k$ ならば、補題 1-2 より u_a 以降のタスクは時刻 $e(u_a)$ でスケジュールできるので、 u_{a+1} 以降のタスクを外部プロセッサにスケジュールし、メインプロセッサには時刻 0 から $pred(B) \cup B$, $(pred(C) \cup C) \setminus (pred(B) \cup B)$ を順にスケジュールし、通信を待って残り $a - (|B| + |C|)$ 個のタスクをスケジュールすれば、 $|pred(B) \cup B| + |(pred(C) \cup C) \setminus (pred(B) \cup B)| \leq e(u_a) + k$ であることから、 v のスケジュール時刻の上界は $e(u_a) + k + a - (|B| + |C|) \leq 2.43k \leq \frac{5}{3}opt$ となり、近似精度 $\frac{5}{3}$ のスケジュールとなる。(図 9)

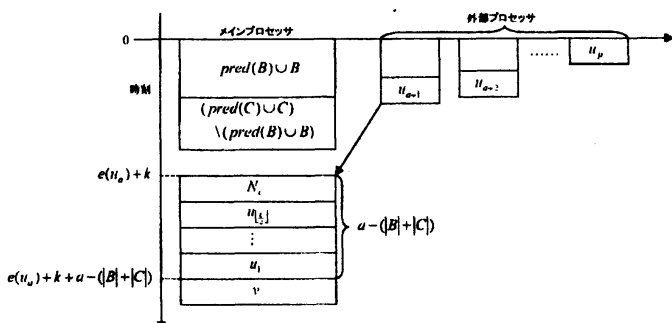


図 9: $e(u_a) > 0.5k$ かつ $e(u_{\lfloor \frac{k}{2} \rfloor}) > 0.93k + 1$ かつ $|B| < a - 0.86k$ かつ $|B| + |C| \geq a - 0.86k$ の場合のスケジュール

- $|B| < a - 0.86k$ かつ $|B| + |C| < a - 0.86k$ のとき、 $m > 0.36k, n > 0.36k$ であり、どのようにスケジュールしたとしても v のスケジュール時刻が $1.5k$ 以降になることを以下に示す。即ち $opt \geq 1.5k$ が成立する。

- * $\{u_1, \dots, u_a\}$ のうち 1 個以上のタスクを外部プロセッサにスケジュールすると、 v のスケジュール時刻は $e(u_a) + k > 1.5k$ 以降となる。

- * $\{u_1, \dots, u_a\}$ を全てメインプロセッサにスケジュールし、

M_p のうち $\lfloor 0.06k \rfloor + 1$ 個以上、または N_p のうち $\lfloor 0.06k \rfloor + 1$ 個以上のタスクを外部プロセッサにスケジュールすると、マッチングにおいて対応している M_c または N_c の要素を $\lfloor 0.06k \rfloor + 1$ 個以上通信を待ってメインプロセッサにスケジュールすることになる。 $e(u_{\lfloor \frac{k}{2} \rfloor}) > 0.93k + 1$ より、メインプロセッサに時刻 $\lfloor 0.93k \rfloor + 1$ 以降にスケジュールするタスクの個数が $0.56k - 1$ 個以上になり ($\because \lfloor 0.5k \rfloor + \lfloor 0.06k \rfloor + 1 \geq 0.56k - 1$), v のスケジュール時刻は $1.5k$ 以降となる。

M_p のうち $\lfloor 0.36k \rfloor - \lfloor 0.06k \rfloor$ 個以上のタスクと、 N_p のうち $\lfloor 0.36k \rfloor - \lfloor 0.06k \rfloor$ 個以上のタスクをメインプロセッサにスケジュールすると、メインプロセッサにスケジュールするタスクの個数が $2(\lfloor 0.36k \rfloor - \lfloor 0.06k \rfloor) + \lfloor 0.93k \rfloor \geq 2(0.36k - 0.06k) + 0.93k = 1.53k \geq 1.5k$ となり、 v のスケジュール時刻は $1.5k$ 以降となる。 □

補題 5 $1.5k \leq low(v) < 2k$ のとき、近似精度 $\frac{5}{3}$ でスケジュール可能。

証明) 定義より $k + e(u_k) \leq e(v) \leq low(v)$ なので、 $e(u_k) < k$ 。補題 1-2 より u_k 以降のタスクは時刻 $e(u_k)$ でスケジュールできるので、 u_{k+1} 以降のタスクを外部プロセッサにスケジュールし、通信を待って残りの k 個のタスクをメインプロセッサにスケジュールすると、 v のスケジュール時刻の上界は $e(u_k) + k + k \leq e(v) + \frac{2}{3}e(v) = \frac{5}{3}e(v)$ となり、近似精度 $\frac{5}{3}$ のスケジュールとなる。(図 10) □

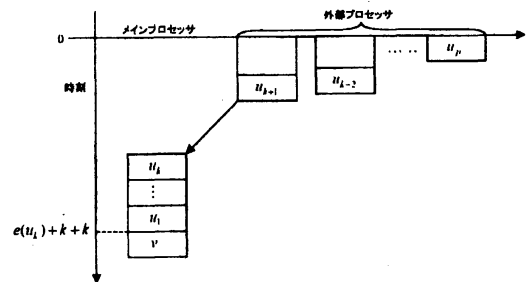


図 10: $1.5k \leq low(v) < 2k$ の場合のスケジュール

補題 1~補題 5 から、次の系が得られる。

系 1 $k \geq 100$ かつ $k \leq low(v) < 2k$ のとき、近似精度 $\frac{5}{3}$ でスケジュール可能。

定理 1 $k \geq 100$ かつ、ある自然数 c について $ck \leq low(v) < (c+1)k$, 即ち $c = \lfloor \frac{low(v)}{k} \rfloor$ のとき、近似精度 $2 - \frac{1}{3c}$ でスケジュール可能。

証明) c に関する帰納法で証明する。

- $c = 1$ のとき、即ち $k \leq \text{low}(v) < 2k$ のときについては、系 1 より成立する。 $(\frac{5}{3} = 2 - \frac{1}{3 \cdot 1})$
- $c \geq 2$ のとき、定義より $k + \text{low}(u_k) \leq \text{low}(v)$ 、即ち $\text{low}(u_k) < ck$ 。帰納法の仮定により、 u_k 以降のタスクは $2 - \frac{1}{3(c-1)} \text{low}(u_k)$ でスケジュール可能である。 u_{k+1} 以降のタスクを外部プロセッサにスケジュールし、通信を待って残りの k 個のタスクをメインプロセッサにスケジュールすると、 v のスケジュール時刻の上界は以下のようになり、成立する。(図 11)

$$\begin{aligned}
 & (2 - \frac{1}{3(c-1)})\text{low}(u_k) + k + k \\
 = & (2 - \frac{1}{3c} \cdot \frac{c}{c-1})(\text{low}(u_k) + k) + \frac{1}{3c} \cdot \frac{c}{c-1} \cdot k \\
 \leq & (2 - \frac{1}{3c} \cdot \frac{c}{c-1})\text{low}(v) + \frac{1}{3c} \cdot \frac{1}{c-1} \cdot \text{low}(v) \\
 = & (2 - \frac{1}{3c})\text{low}(v)
 \end{aligned}$$

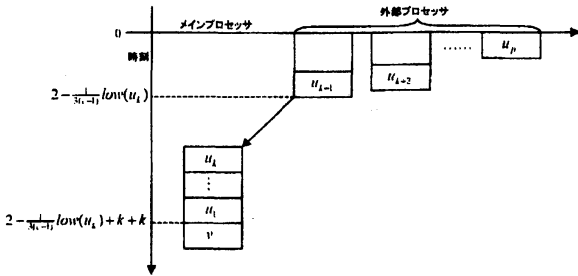


図 11: $c \geq 2$ のときのスケジュール

[4] で、近似精度 $2 - \frac{1}{k}$ のアルゴリズムが示されているが、[4] と同様の手法を用いて、定理 1 を更に改善することができる。

定理 2 $k \geq 100$ かつ、ある自然数 c について $ck \leq \text{low}(v) < (c+1)k$ のとき、即ち $c = \lfloor \frac{\text{low}(v)}{k} \rfloor$ のとき、近似精度 $2 - \frac{1}{k} - \frac{1}{ac}$ でスケジュール可能。ただし、 $a = \frac{3k}{k-3}$

証明) c に関する帰納法で証明する。

- $c = 1$ のとき、即ち $k \leq \text{low}(v) < 2k$ のときについては、系 1 より成立する。 $(\frac{5}{3} = 2 - \frac{1}{k} - \frac{1}{a \cdot 1})$
- $c \geq 2$ のとき、定義より $k + \text{low}(u_k) \leq \text{low}(v)$ 、即ち $\text{low}(u_k) < ck$ 。帰納法の仮定により、 u_k 以降のタスクは $2 - \frac{1}{k} - \frac{1}{a(c-1)} \text{low}(u_k)$ でスケジュール可能である。 u_k 以降のタスクを外部プロセッサにスケジュールし、通信を待って残りの $k-1$ 個のタスクをメインプロセッサにスケジュールすると、 v のスケジュール時刻の上界は以下のようになり、成立する。(図 12)

$$\begin{aligned}
 & (2 - \frac{1}{k} - \frac{1}{a(c-1)})\text{low}(u_k) + k + k - 1 \\
 = & (2 - \frac{1}{k} - \frac{1}{ac} \cdot \frac{c}{c-1})(\text{low}(u_k) + k) + \frac{1}{ac} \cdot \frac{c}{c-1} \cdot k \\
 \leq & (2 - \frac{1}{k} - \frac{1}{ac} \cdot \frac{c}{c-1})\text{low}(v) + \frac{1}{ac} \cdot \frac{1}{c-1} \cdot \text{low}(v) \\
 = & (2 - \frac{1}{k} - \frac{1}{ac})\text{low}(v)
 \end{aligned}$$

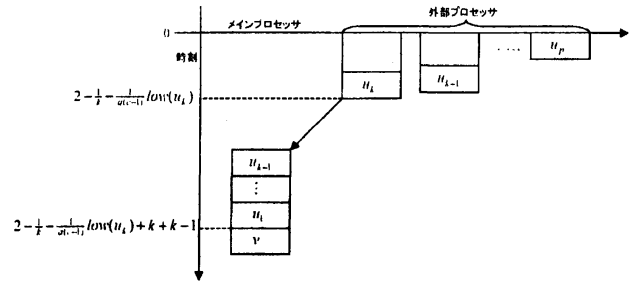


図 12: $c \geq 2$ のときのスケジュール

5 本手法の計算量

本手法の計算量について述べる。

まず、e-value の計算は $O(|V||E|)$ [3] ででき、 v の先祖を e-value の降順にソートするのは $O(|V|^2)$ ができる。タスク v をスケジュールするとき、 $e(v) < 2k$ の場合、DAG から二部グラフを構成するのは $O(|V|^2)$ ができ、二部グラフの最大マッチングを計算するのは $O(|V||E|)$ ができ、スケジュールは $O(k|V|)$ ができる。 $k \leq |V| \leq |E|$ なので、全体で $O(|V||E|)$ となる。よって、e-value が $2k$ 未満である全てのタスクのスケジュールは合計で $O(|V|^2|E|)$ となる。残りのタスクはそれぞれ $O(k)$ でスケジュールできるので、合計で $O(|V|^2)$ となる。

よって、本手法の計算量は全体で $O(|V|^2|E|)$ となる。

6 おわりに

本研究では、通信遅延を考慮したタスクスケジューリング問題に対して、新たな下界 $\text{low}(v)$ を定義し、それを用いて近似精度 $2 - \frac{1}{3c}$ のアルゴリズムを与えた。

謝辞

この研究の一部は栢森情報科学振興財団の助成を受けて遂行された。

参考文献

- [1] C.H.Papadimitriou and M.Yannakakis, "Towards an Architecture-Independent Analysis of Parallel Algorithms", SIAM J.Comput.,vol.19, no.2, pp.322-328, 1990.
- [2] 河田 俊郎, 大山口 通夫, 太田 義勝, 「通信遅延を考慮したタスクスケジューリングアルゴリズムについて」, 電子情報通信学会論文誌, Vol.J85-D-I, No.11, pp.1088-1092, 2002 年.
- [3] 坂上 知英, 大山口 通夫, 太田 義勝, 「通信遅延を考慮したタスクスケジューリングの近似アルゴリズムについて」, 2001 年度電気関係学会東海支部連合大会 講演論文集, p.339, 2001.
- [4] Michael A. Palis, Jing-Chiou Liou, and David S.L. Wei, "Task Clustering and Scheduling for Distributed Memory Parallel Architectures", IEEE Transactions on Parallel and Distributed Systems, Vol. 7, no. 1, pp.46-55, January 1996.