

# 最大マッチングを利用したタスクスケジューリング アルゴリズムの近似度の改善について

## An Improved Approximation Ratio for Task Scheduling Algorithm using Maximum Matching

新美信之助 大山口通夫 太田義勝 山本浩平  
(三重大学工学部)

Shinnosuke NIIMI, Michio OYAMAGUCHI, Yoshikatsu OHTA, Kohei YAMAMOTO  
(Faculty of Engineering, Mie University)

### 概要

Papadimitriou ら (1990) は, タスクの複製を許す通信遅延を考慮した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, 最適解の下界を与える  $e$ -value を導入して, 近似精度 2 のアルゴリズムを与えた. また, 加藤ら (2004) は  $e$ -value を改善した下界  $low$ -value を導入して,  $c = \lfloor low\text{-value}/(\text{通信遅延時間} + 1) \rfloor (\geq 1)$  のとき,  $c = 1$  で近似精度  $14/9$ , 及び  $c \geq 2$  のとき近似精度  $2 - 1/2c$  のアルゴリズムを与えており, この結果は今までに知られている最良の結果であった.

本研究ではタスクの DAG から再構成される二部グラフにおいて多対一の最大マッチングを利用することにより, 加藤らの結果を改善し,  $c = 1$  においては近似精度  $26/17$ ,  $c \geq 2$  においては近似精度  $2 - 1/1.67c$  であるアルゴリズムを示す. ここで, DAG とはタスク間の依存関係を表した有向非循環グラフである. なお, 加藤らの研究では各タスクの実行時間を一定と仮定していたが, 本研究ではその制限を除いて, 実行時間を任意の自然数に一般化した場合の近似アルゴリズムを示している.

### 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, 異なるプロセッサに割り当てられたタスク間の通信のため通信遅延が発生する. システムの性能を最大限に発揮し, 全タスクの処理時間を短縮するためには, 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] であるので, 効率的な近似アルゴリズムの研究が行われている.

文献 [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 近似解/最適解 と定義する. また, 文献 [2] では 文献 [1] 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スケジューリング法を提案した. この手法を用いて文献 [2] では, 通信遅延時間  $\tau$  ( $\tau \geq 99$ ) において, 従来の下界  $e$ -value を改善した最適解の下界  $low$ -value を新たに定義し, 近似精度  $2 - 1/3c$  を持つ近似アルゴリズムを示した. ここで,  $c = \lfloor low\text{-value}/(\tau + 1) \rfloor (\geq 1)$  である.

文献 [3] では文献 [2] の手法を拡張し, 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から独立したタスク集合を発見する Zigzag 法を提案した. さらに, 文献 [3] は Zigzag 法を用いることにより, 通信遅延時間の制限の無い, 文献 [2] の結果を改善した,  $c = 1$  のとき近似精度  $14/9$ , 及び  $c \geq 2$  のとき近似精度  $2 - 1/2c$  の近似アルゴリズムを示した.

本研究では文献 [1, 2, 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において多対一の最大マッチングを利用し,  $c = 1$  においては近似精度  $26/17$ ,  $c \geq 2$  においては近似精度  $2 - 1/1.67c$  の近似アルゴリズムへ改善を行う. さらに, 従来の前提条件を緩和し, 各タスクの実行時間を任意の自然数としても, 同じ近似精度のスケジューリングアルゴリズムが得られることを示す. なお, 紙面の制約のため, 本論文の全ての補題と定理の証明を割愛する.

### 2 準備

通信遅延を考慮したタスクスケジューリング問題とは, 各タスクを頂点とし各タスク間の依存関係を有向辺とした非循環有向グラフ (DAG) と, 各タスクの実行時間と, 各プロセッサ間の通信遅延時間が与えられたときに, 全タスクの実行を最短に終了するスケジュール, 最適スケジュールを求める問題である. ここでスケジュールとは, 各タスクの, 開始時刻と処理を行うプロセッサを決定することである.

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. 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正の整数  $\tau$
- (4) タスクの複製を許す

という文献 [1], [2], [3] と同じ条件を前提としている. この前提の下で, 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.  $G$  の根へのパス長が最大となるタスク  $v \in V$  について, 前提 (1), (4) より, ただ一つである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])

$v \in V$  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 文献 [1] の証明と同様に証明できる.

**定義 3** DAG の頂点  $u$  の先祖の集合  $pred(u)$  を次のように定義する.

$$pred(u) = \bigcup_{(u', u) \in E} (pred(u') \cup \{u'\})$$

特に,  $u$  がグラフの根の場合は  $pred(u) = \emptyset$  となる. また, 頂点の集合  $V' \subseteq V$  に対し,  $pred(V') = \bigcup_{u \in V'} pred(u)$  とする.

**定義 4** 以下の条件を満たすように, DAG  $G = (V, E)$  から構成する二部グラフ  $(U_1, U_2, E')$  を定義する.

$U_1, U_2 \subseteq V \setminus \{v^*\}$ ,  $U_1 \cap U_2 = \emptyset$ , かつ  $E' \subseteq E \cap (U_2 \times U_1)$

**定義 5** 二部グラフ  $(U_1, U_2, E')$  の部分二部グラフ  $(V_1, V_2, E'')$  が  $(U_1, U_2, E')$  のマッチングであるとは,  $(V_1, V_2, E'')$  について  $v, v' \in V_1, v \neq v'$  ならば  $pred(v) \cap pred(v') = \emptyset$  であり, かつ  $|pred(v)| = 1$  を満たすことである. そのとき,  $|E''|$  が最大のものを二部グラフ  $(U_1, U_2, E')$  の最大マッチングと定義する.

また,  $k \leq e(v) \leq 2k$  の範囲で達成する近似精度を  $r_1 > 3/2, 2k \leq e(v) \leq 3k$  の範囲における近似精度を  $r_2 = 1.7$  とする.

### 3 下界 $low(v)$

$v \in V$  に対して, 下界  $low$ -value  $low(v)$  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,
  - $k(v) \leq e(v) < 2k(v)$  について
    - \*  $k(v) \leq e(v) < \frac{1}{r_1-1}k(v)$  のとき,  $low(v)$  は 7 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,  $e(v) \leq low(v) \leq e(v) + (\frac{1}{r_1-1} - 1)k(v)$  を満たす.
    - \*  $\frac{1}{r_1-1}k(v) \leq e(v) < 2k(v)$  のとき,  $low(v) = e(v)$
  - $2k(v) \leq e(v) < 3k(v)$  について
    - \*  $2k(v) \leq e(v) \leq \frac{3r_1+2}{3(r_1-r_2+1)}k(v)$  のとき,  $low(v)$  は 7 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,  $2e(v) \leq low(v) \leq e(v) + \frac{3r_1+2}{3(r_1-r_2-2)}k(v)$  を満たす.
    - \*  $\frac{3r_1+2}{3(r_1-r_2+1)}k(v) \leq e(v) < 3k(v)$  のとき,  $low(v) = e(v)$
  - $e(v) \geq 3k(v)$  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,  $low(v) = \max_{1 \leq i \leq k(v)} (low(u_i) + i)$  とする.

$k(v) \leq e(v) < \frac{1}{r_1-1}k(v), 2k(v) \leq e(v) \leq \frac{3r_1+2}{3(r_1-r_2+1)}k(v)$  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 7 節で示す. 残りの範囲について,  $low(v) \leq opt(v)$  であることは 文献 [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.

### 4 $s$ 対 1 の最大マッチング

通常の 1 対 1 の最大マッチング (定義 6) を拡張した  $s$  対 1 ( $s \geq 1$ ) の最大マッチングを定義する.

二部グラフ  $(U_1, U_2, E')$  の部分二部グラフ  $(V_1, V_2, E'')$  が  $s$  対 1 のマッチングであるとは,  $(V_1, V_2, E'')$  において  $v, v' \in V_1, v \neq v'$  ならば  $pred(v) \cap pred(v') = \emptyset$  かつ  $1 \leq |pred(v)| \leq s$  を満たすことである. そのとき,  $|E''|$  が最大のものを  $s$  対 1 の最大マッチングと定義する.

さらに,  $1 \leq i \leq s$  のとき,  $(V_1, V_2, E'')$  に対して, ちょうど  $i$  個の  $V_2$  の頂点とマッチングしている  $V_1$  の頂点集合  $P_i$  を

$$P_i = \{v \in V_1 \mid |pred(v)| = i\}$$

と定める.  $i$  個以上の  $V_2$  の頂点とマッチングしている  $V_1$  の頂点集合である  $M'_i$  と, その頂点数  $m_i$  を

$$M'_i = \bigcup_{i \leq x \leq s} P_x, \quad m_i = |M'_i|$$

と定める. なお,  $s$  対 1 の最大マッチングと  $s'$  対 1 の最大マッチングに対する,  $m_j$  の値は等しい. ここで,  $j \leq s$  かつ  $j \leq s'$  とする.

さらに,  $\sigma: V_2 \rightarrow \{1, 2, \dots, s\}$  を各  $u' \in V_1$  に対して,  $\{\sigma(v') \mid v' \in \text{pred}(u)\} = \{1, \dots, |\text{pred}(u)|\}$  としたとき,

$$M_i = \{u \mid \sigma(u) = i, u \in V_2\} \text{ とする.}$$

$M_i$  は関数  $\sigma$  によって番号  $i$  が割り振られた  $V_2$  中の頂点の集合であり,  $V_2 = \bigcup_{1 \leq i \leq s} M_i$  である.

マッチする頂点が  $i-1$  個である頂点集合  $N_i$  を  $i=1$  のとき

$$N_1 = U_1 \setminus M'_1$$

,  $i \geq 2$  のとき

$$N_i = P_{i-1}$$

と定義する. ここで, DAG  $G = (V, E)$  に対する  $s$  対 1 の最大マッチング  $(V_1, V_2, E'')$  を考える.  $u \in N_s$  について  $u' \in \text{pred}(u)$  かつ  $u' \in M'_s$  となるような  $u'$  が存在したとき,  $\text{pred}(N_s) \subseteq V_2 \cup (\bigcup_{1 \leq i \leq s} N_i)$  が成立するように,  $s$  対 1 の最大マッチング  $(V'_1, V_2, E''')$  を次のように再構成する.  $(V_1, V_2, E'')$  において,  $(t, u') \in E''$ ,  $(t, u) \in E$  となるタスク  $t$  が存在するので,  $E''' = E'' \setminus (t, u') \cup (t, u)$ ,  $V'_1 = V_1 \setminus u' \cup u$  とする. これにより,  $\text{pred}(N_s) \subseteq V_2 \cup (\bigcup_{1 \leq i \leq s} N_i)$  が成立する. (図 1)

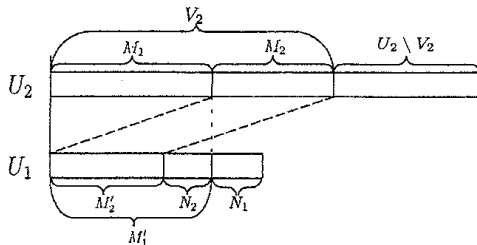


図 1:  $s=2$  の場合の  $s$  対 1 の最大マッチング

### 5 Alternating Path

二部グラフ  $(U_1, U_2, E')$  において  $s$  対 1 の最大マッチングをとったとき, 以下の条件を満たす路  $v_1, v'_1, \dots, v_t, v'_t, \dots, v_q, v'_q$  を Alternating Path<sup>1</sup> と定義する.

$1 \leq t \leq q$  に対して,

- $v_1 \in N_s \wedge v'_q \in U_2 \setminus V_2 \wedge v_t \in V_1 \wedge v'_t \in U_2$
- $(v'_{t-1}, v_t) \in E'' \wedge (v'_t, v_t) \in E' \setminus E''$

Alternating Path 上の辺においてマッチングに使用されていない辺はマッチングに使用されている辺よりも 1 本多い.

補題 1  $N_s$  の任意の頂点  $v_1$  を始点とする Alternating Path  $v_1, v'_1, \dots, v_t, v'_t, \dots, v_q, v'_q$  は存在しない.

<sup>1</sup>Augmenting Path [5]

### 6 Zigzag 法

Zigzag 法は  $s$  対 1 の最大マッチング  $(V_1, V_2, E'')$  において,  $N_s \neq \emptyset$  である場合に以下の処理により全体から独立してスケジュールできる頂点を求める方法である. Zigzag 法は次のように定義される.

```

L0 := ∅, L'0 := Ns, q := 0
repeat
  q := q + 1
  Lq := {u | u ∈ pred(L'q-1) ∩ U2} - ∪_{j ≤ q-1} Lj
  L'q := {u' | (u, u') ∈ E'' ∧ u ∈ Lq}
until Lq = ∅
L := ∪_{j ≤ q} Lj, L' := ∪_{j ≤ q} L'j
    
```

ここで  $L_q \subseteq V$  より, この処理は高々  $|V|$  回の繰り返しで停止する.

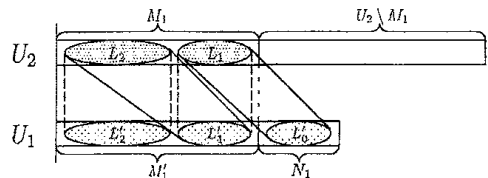


図 2:  $s$  対 1 の最大マッチングにおける Zigzag 法 ( $s=1$ )

性質 1 任意の  $v \in L$  に対してある  $v' \in N_s$  が存在して,  $v$  と  $v'$  を結ぶマッチングに含まれない辺と含まれる辺を交互に辿る路が存在する.

また, このとき Zigzag 法で求めた頂点が  $U_2 \setminus V_2$  に存在しない事を証明する.

補題 2  $s$  対 1 の最大マッチングにおいて,  $N_s \neq \emptyset$  のとき Zigzag 法により求められた集合  $L$  に対し  $L \cap (U_2 \setminus V_2) = \emptyset$  である.

### 7 近似アルゴリズム

本節では近似精度  $2-1/1.67c$  の近似アルゴリズムを示す.

タスク  $v^*$  をスケジュール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呼ぶ. 以下では, タスクの集合  $X$  をプロセッサにスケジュールするとは,  $X$  の全ての要素を下界  $e$ -value の値の小さい順にプロセッサにスケジュールすることを表す.

補題 3 ([1])

- (1)  $p(v^*) \leq \tau + 1$  ならば,  $v^*$  は時刻  $p(v^*)$  でスケジュールでき, 最適スケジュールとなる.
- (2)  $e(v^*) < \tau + 1$  ならば,  $v^*$  は時刻  $e(v^*)$  でスケジュールでき, 最適スケジュールとなる.

従って、以下では  $p(v^*) > \tau + 1$  の場合について考える。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 4, 5, 6, 8, 9, 10 では  $opt$  が下界  $low(v)$  以上である事を利用し段階的に  $opt$  の範囲を解析する。

補題 4  $k \leq e(v) < \frac{1}{r_1-1}k$  のとき、すべての  $i$  ( $0 \leq i < \frac{r_1-1}{3-r_1}k \wedge i \in N$ ) とすべての  $x$  ( $(r_1-1)(k+i) \leq x \leq k-1 \wedge x \in N$ ) に対して、 $opt \geq k+i$  かつ  $e(u_x) \geq i+1$  ならば、次のいずれかが成立する。

(i).  $e(u_{x+1}) \geq i+1$

(ii).  $opt \geq k+i+1$

(iii). 近似精度  $r_1$  でスケジュール可能。(このとき、 $low(v) = \max(e(v), k+i)$  と定義する。)

補題 5  $k \leq e(v) < \frac{1}{r_1-1}k$  のとき、すべての  $i$  ( $\frac{r_1-1}{3-r_1}k \leq i < \frac{2-r_1}{r_1-1}k \wedge i \in N$ ) とすべての  $x$  ( $(r_1-1)(k+i) \leq x \leq k-1 \wedge x \in N$ ) に対して、 $opt \geq k+i$  かつ  $e(u_x) \geq i+1$  ならば、次のいずれかが成立する。

(i).  $e(u_{x+1}) \geq i+1$

(ii).  $opt \geq k+i+1$

(iii).  $0 \leq i < \frac{5}{12}k$  ならば  $i \leq \frac{3r_1-4}{6-3r_1}k$  を満たせば近似精度  $r_1$  でスケジュール可能。 $\frac{5}{12}k \leq i < \frac{2-r_1}{r_1-1}k$  ならば  $j$  対 1 の最大マッチング ( $j \geq 3$ ) をとり Zigzag 法を用いることにより、 $i \geq \frac{1-r_1+(2-r_1)}{(r_1-1)(j+1)}k$  かつ  $r_1 \geq \frac{3j^2-1}{2j^2-1}$  を満たせば近似精度  $r_1$  でスケジュール可能。(このとき  $low(v) = \max(e(v), k+i)$  と定義する。)

補題 6 は 補題 4, 5 を使用して証明する。

補題 6  $k \leq e(v) < \frac{1}{r_1-1}k$  かつ  $opt \geq k$  のとき、以下のいずれかが成立する。

•  $opt \geq \frac{1}{r_1-1}k$

•  $k \leq low(v) < zk$  ( $1 \leq z < \frac{5}{12}$ ) の範囲では  $i \leq \frac{3r_1-4}{6-3r_1}k$  を満たせば近似精度  $r_1$  でスケジュール可能。 $zk \leq low(v) < 2k$  ( $\frac{5}{12} \leq z < 2$ ) ならば  $i \geq \frac{1-r_1+(2-r_1)}{(r_1-1)(j+1)}k$  かつ  $r_1 \geq \frac{3j^2-1}{2j^2-1}$  ( $j \geq 3$ ) を満たせば近似精度  $r_1$  でスケジュール可能。

補題 7  $\frac{1}{r_1-1}k \leq e(v) < 2k$  のとき、近似精度  $r_1$  でスケジュール可能。

定理 1 補題 4, 5, 6, 7 より、 $k \leq low(v) < zk$  ( $1 \leq z < \frac{5}{12}$ ) の範囲では  $i \leq \frac{3r_1-4}{6-3r_1}k$  を満たせば近似精度  $r_1$  でスケジュール可能。また、 $zk \leq low(v) < 2k$  ( $\frac{5}{12} \leq z < 2$ ) の範囲では  $i \geq \frac{1-r_1+(2-r_1)}{(r_1-1)(j+1)}k$  かつ  $r_1 \geq \frac{3j^2-1}{2j^2-1}$  ( $j \geq 3$ ) を満たせば近似精度  $r_1$  でスケジュール可能。

系 1 定理 1 より、 $k \leq e(v) < 2k$  のとき、近似精度  $r_1 = \frac{26}{17}$  が本研究で提案するアルゴリズムで導くことのできる最良の近似精度である。

補題 8  $2k \leq e(v) < 2k + \frac{4r_2-2r_1-3}{-2r_2+2r_1+2}k$  のとき

すべての  $i$  ( $0 \leq i < \frac{4r_2-2r_1-3}{-2r_2+2r_1+2}k$ ) と

すべての  $x$  ( $(2r_2-r_1-1)k + (r_2-r_1)i \leq x \leq k-1 \wedge x \in N$ ) ( $N$ : 自然数の集合) に対して、 $opt \geq 2k+i$  かつ  $e(u_x) \geq k+i+1$  ならば、次の (i) ~ (iii) のいずれかが成立する。

(i).  $e(u_{x+1}) \geq k+i+1$

(ii).  $opt \geq 2k+i+1$

(iii). 近似精度  $r_2$  ( $= 1.7$ ) でスケジュール可能。(このとき、 $low(v) = \max(e(v), 2k+i)$  と定義。)

補題 9  $2k + \frac{4r_2-2r_1-3}{-2r_2+2r_1+2}k \leq e(v) < 2k + \frac{6r_2-3r_1-4}{3(-r_2+r_1+1)}k$  のときすべての  $i$  ( $\frac{-6r_2+4r_1+5}{3(r_2-1)}k \leq i < \frac{6r_2-3r_1-4}{3(-r_2+r_1+1)}k$ ) とすべての  $x$  ( $(2r_2-r_1-1)k + (r_2-r_1)i \leq x \leq k-1 \wedge x \in N$ ) ( $N$ : 自然数の集合) に対して、

$opt \geq 2k+i$  かつ  $e(u_x) \geq k+i+1$  ならば、次の (i) ~ (iii) のいずれかが成立する。

(i).  $e(u_{x+1}) \geq k+i+1$

(ii).  $opt \geq 2k+i+1$

(iii). 近似精度  $r_2$  ( $= 1.7$ ) でスケジュール可能。(このとき、 $low(v) = \max(e(v), 2k+i)$  と定義。)

補題 10  $2k \leq e(v) < 2k + \frac{6r_2-3r_1-4}{3(-r_2+r_1+1)}k$  かつ  $opt \geq 2k$  のとき、以下のいずれかが成立する。

•  $opt \geq 2k + \frac{6r_2-3r_1-4}{3(-r_2+r_1+1)}k$

• 近似精度  $r_2$  でスケジュール可能

補題 11  $2k + \frac{6r_2-3r_1-4}{3(-r_2+r_1+1)}k \leq e(v) < 3k$  のとき、近似精度  $r_2$  ( $= 1.7$ ) でスケジュール可能。

系 2 補題 8, 9, 10, 11 より、 $2k \leq e(v) < 3k$  のとき、近似精度  $r_2$  でスケジュール可能。

定理 2 すべての自然数  $c \geq 2$  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  $2 - \frac{1}{1.67c}$  でスケジュール可能。

## 8 本研究結果の拡張

本研究の前提条件「各タスクの実行時間は 1」は近似精度を保ったまま「各タスクの実行時間は任意の自然数」と拡張可能である。

アルゴリズムの変更点は、まずタスクの実行時間  $b$  ( $\geq 2$ ) のものを実行時間 1 の  $b$  個のタスク  $T_1, \dots, T_b$  に分割し、 $T_i \rightarrow T_{i+1}$  ( $1 \leq i < b$ ) の依存関係を導入する。分割して得られたタスクの集合に対して、本研究で提案したスケジューリングアルゴリズムを適用する。異なる点は、スケジュールの際に分割した

タスクを同一プロセッサに割り当てるように再スケジューリングすることをアルゴリズムの最後に追加するだけであり、この変更したアルゴリズムにおいても同じ近似精度が達成される。

## 9 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、 $c=1$ において、より詳細に解析することにより近似精度を従来の  $14/9$  から  $26/17$  に改善した。また、 $c=2$ において、今まで用いることの出来なかった二部グラフの最大マッチングを利用したスケジューリング方法を用いることにより、近似精度を改善するとともに、 $c \geq 2$ において近似精度  $2 - 1/1.67c$  のアルゴリズムを与えた。本手法の計算量は  $O(k^2|V|^4)$  となり、文献 [3] と等しい。また、前提条件を緩和し各タスクの実行時間を任意の自然数としても同じ近似精度のアルゴリズムが得られることを示した。

## 謝辞

日頃、御指導・御鞭撻いただきました山田俊行助手に深く感謝致します。また、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ました。

## 参考文献

- [1] C.H.Papadimitriou, et al., "Towards an Architecture-Independent Analysis of Parallel Algorithms", SIAM J.Comput., vol.19, No.2, pp.322-328, 1990.
- [2] 片柳賢二ら, "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", 京大数解研講究録, No.1325, pp.185-190, 2003.
- [3] 加藤雅之, "スケジューリング問題の近似アルゴリズムに関する研究", 2003年度三重大学大学院工学研究科 修士論文, 2004.
- [4] D. C. Kozen, "The Design and Analysis of Algorithms", Springer-Verlag, New York, 1992.
- [5] J. E. Hopcroft, et al., "An  $n^{\frac{5}{2}}$  Algorithm for Maximum Matching in Bipartite Graphs", SIAM J.Comput., vol.2, No.4, pp.225-231, December, 1973.