数理解析研究所講究録 1375 巻 2004 年 61-67

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

加藤雅之　大山口通夫　太田義勝　新美信之助　山本浩平
(三重大学工学部)
Masayuki KATO, Michio OYAMAGUCHI, Yoshikatsu OHTA,
Shinnosuke NIIMI and Kouhei YAMAMOTO
(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 $2-\frac{1}{3\cdot 2^{L-3}}$ を持つ近似アルゴリズムを示した．また，片柳ら (2003) は e-value による下界よりも正確な下界 low-value を定義すると共に，通信遅延時間 τ $(\tau \geq 99)$ において近似精度を $2-\frac{1}{3c}$ に改善したアルゴリズムが存在することを示した．ここで，$c=\lfloor \frac{low-value}{\tau+1} \rfloor$ であり，$c \leq L$ を満たす．

本研究では通信遅延時間 $\tau \geq 1$ において片柳らの結果を改善し，$c=1$ においては近似精度 $\frac{14}{9}$，$c \geq 2$ においては近似精度 $2-\frac{1}{2c}$ であるアルゴリズムを示す．

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，異なるプロセッサに割り当てられたタスク間の通信において通信遅延が発生する．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 $\frac{\text{近似解}}{\text{最適解}}$ と定義する．[2] 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 $2-\frac{1}{3\cdot 2^{L-3}}$ を持つ近似アルゴリズムを示している．また，[3] では [1], [2] 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-value を新たに定義すると共に通信遅延時間 τ $(\tau \geq 99)$ において，[2] 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した．ここで，$c=\lfloor \frac{low-value}{\tau+1} \rfloor$ であり，$c \leq L$ を満たす．

本研究では [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，通信遅延時間 $\tau \geq 1$ において近似精度 $2-\frac{1}{2c}(c \geq 2)$ の近似アルゴリズムへ改善した．

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定数 $\tau(\tau \geq 1)$
(4) タスクの複製を許す

という [1], [2], [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])*
$v \in V$ 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, $e(u_1) \geq e(u_2) \geq \cdots \geq e(u_{p(v)})$ 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]* 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 $U \subseteq V$ に対し, $pred(U) = \bigcup_{u \in U} pred(u)$ とする.

定義 4 $V_1, V_2 \subseteq V \backslash \{v^*\}$ かつ $V_1 \cap V_2 = \phi$ とする. そのとき, 二部グラフ $f_G(V_2 \times V_1) = (V', E')$ を次のように定義する.
$V' = V_1 \cup V_2$, $E' = \{(u, u') | (u, u') \in V_2 \times V_1 \wedge u \in pred(u')\}$

3 下界 $low(v)$

$v \in V$ に対して, 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,
  - $k(v) \leq e(v) < 1.8k(v)$ のとき, $low(v)$ は 8 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, $e(v) \leq low(v) \leq e(v) + 0.8k(v)$ を満たす.
  - $1.8k(v) \leq e(v) < 2k(v)$ 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, $low(u_1) \geq low(u_2) \geq \cdots \geq low(u_p(v))$ として, $low(v) = \max_{1 \leq i \leq k(v)} (low(u_i) + i)$ とする.

$k(v) \leq e(v) < 1.8k(v)$ 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 8 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 ([2] 参照) が, この下界を用いることで高さ 2 の DAG に対し $\frac{4}{3} + \varepsilon$(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 多対一の最大マッチング

二部グラフ $f_G(V_2 \times V_1)$ において, V_1 中の各頂点が高々 s 回, V_2 中の各頂点が高々 1 回マッチングに使用され, かつ V_2 中でマッチングに使用される頂点の数が最大となるマッチングを多対一 $(s:1)$ の最大マッチング $\mathcal{M}_s \subseteq V_2 \times V_1$ という.

以降, 特に明記しない場合, U_1, U_2 を以下のものとして扱う. v^* のすべての先祖を e-value の降順にソートしたものを $u_1, u_2, \cdots, u_{p(v^*)}$ とする. すべての i $(\frac{5}{13}k \leq i < 0.8k \wedge i \in N)$ (N : 自然数の集合) とすべての x $(\frac{5}{9}(k+i) \leq x \leq k-1 \wedge x \in N)$ に対して $a = \lceil \frac{14}{9}(k+i) - 2k \rceil$ とし, U_1, U_2 を以下のものとする. (補題 6, 7)

$$U_1 = \{u_1, \cdots, u_x\},\ U_2 = \{u_{x+1}, \cdots, u_{p(v^*)}\}$$

また $s \geq 1$ とし, $s:1$ の最大マッチングにおいて関数 $mpred_s$ を以下のように定義する.

$$mpred_s(u) = \{u' | (u', u) \in \mathcal{M}_s\}$$

さらに $\mathcal{M}_s$ を二部グラフ $f_G(U_2 \times (U_1 \setminus \{u_1, \cdots, u_a\}))$ における $s:1$ の最大マッチングとする.

$$M_s = \{u | (u, u') \in \mathcal{M}_s\}$$

$$M'_s = \{u' | u' \in U_1 \wedge |mpred_s(u')| = s\}$$

$$m_s = |M_s|,\ N_s = U_1 \setminus (\{u_1, \cdots, u_a\} \cup M'_s)$$

このとき, m_s はマッチング数である. そして, すべての $u \in pred(N_s)$ に対して $u \notin M'_s$ となるように, 必要なら N_s と M'_s の要素を入れ換えることにより, $pred(N_s) \subseteq M_s \cup N_s$ とする.

また, $s:1$ の最大マッチングにおける $1 \leq i \leq s$ に対して集合 B'_i を以下のように定義する.

$$B'_i = \{u | u \in U_1 \wedge |mpred_s(u)| = i\}$$

ここで, B'_i は i 個のマッチング相手を持つ U_1 中の頂点の集合である.

さらに, $\sigma : M_s \longrightarrow \{1, 2, \cdots, s\}$ をすべての $u' \in U_1 \setminus \{u_1, \cdots, u_a\}$ に対して, $mpred_s(u')$ と $\{1, 2, \cdots, |mpred_s(u')|\}$ を $1:1$ に対応付ける関数とする. そして, $1 \leq j \leq s$ に対して

$$B_j = \{u | \sigma(u) = j \wedge u \in M_s\}$$

とする. ここで, B_j は関数 σ によって番号 j が割り振られた U_2 中の頂点の集合であり, $M_s = \bigcup_{1 \leq j \leq s} B_j$ である. (図 1)

5 最大フロー

二部グラフ $f_G(V_2 \times V_1)$ に対応するフローグラフ G_s を以下のように定義する.

- G_s の頂点集合は $V_1 \cup V_2 \cup \{source, sink\}$.

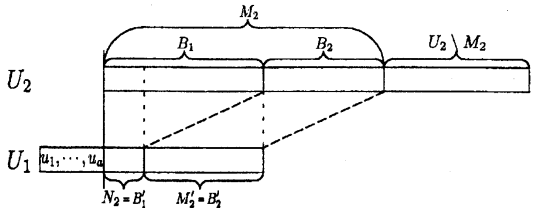

図 1: $s = 2$ の場合の $s:1$ の最大マッチング

- G_s の辺の集合は source から V_1 の各頂点への辺 (容量 s)，二部グラフの辺 (容量 1)，V_2 の各頂点から sink への辺 (容量 1) からなる．

また，本研究における各フロー F は非負整数とする．ここで，フローグラフ G_s における最大フローを $MF(G_s)$ とする．

補題 1 二部グラフ $f_G(U_2 \times (U_1 \setminus (\{u_1, \cdots, u_a\}))$ に対する $s:1$ の最大マッチングにおけるマッチング数 m_s は，対応するフローグラフ G_s における最大フロー $MF(G_s)$ と同じ値となる．

証明)

- $MF(G_s) \geq m_s$ の証明
  $s:1$ の最大マッチングにおいて，U_1 の各頂点は高々 s 回，U_2 の各頂点は高々 1 回しかマッチングに使用されないので，$s:1$ の最大マッチングがとれれば，マッチングに用いた U_1, U_2 間の辺にフローを流すことにより $MF(G_s) \geq m_s$ が得られる．
- $MF(G_s) \leq m_s$ の証明
  U_1, U_2 間の辺のフローは 0 または 1 であり，U_1 中の頂点からフロー 1 が流れ U_2 中の頂点は高々 s 個である．したがって，最大フロー $MF(G_s)$ がとれれば，最大フローにおいて使用された U_1, U_2 間の辺をマッチングとすることで，マッチング数 $MF(G_s) \leq m_s$ が得られる．

よって補題は成立する． □

系 1 $s:1$ の最大マッチングの時間計算量は $O(|V|^3)$ である．

証明) 補題 *1* より，カラザノフのアルゴリズムを利用する．*[5]*

6 Alternating Path

二部グラフ $f_G(U_2 \times (U_1 \setminus \{u_1, \cdots, u_a\}))$ 上の以下の条件を満たす路 $v_1, v'_1, \cdots, v_r, v'_r, \cdots, v_q, v'_q$ を Alternating Path[1] と定義する．

$1 \leq r \leq q$ に対して，

- $v_1 \in N_s \wedge v'_q \in U_2 \setminus M_s \wedge v_r \in M'_s \wedge v'_r \in U_2$
- $(v'_{r-1}, v_r) \in \mathcal{M}_s \wedge (v'_r, v_r) \in E' - \mathcal{M}_s$

[1]Augmenting Path [6]

ここで，E' は二部グラフにおける辺の集合であり，$\mathcal{M}_s$ は $s:1$ の最大マッチングである．また，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．

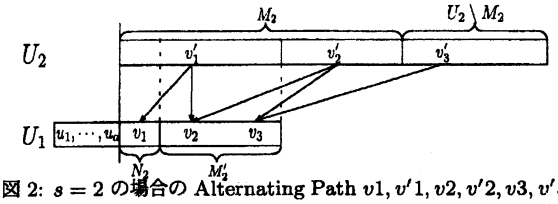

図 2: $s = 2$ の場合の Alternating Path $v1, v'1, v2, v'2, v3, v'3$

補題 2 N_s の任意の頂点 v_1 を始点とする *Alternating Path* $v_1, v'_1, \cdots, v_r, v'_r, \cdots, v_q, v'_q$ は存在しない．

証明) q の場合分けにより証明する．

- $q = 1$ のとき，*Alternating Path* v_1, v'_1 が存在すると仮定すると，マッチング (v'_1, v_1) が存在し，$s:1$ の最大マッチングに矛盾する．
- $q \geq 2$ のとき，*Alternating Path* $v_1, v'_1, \cdots, v_r, v'_r, \cdots, v_q, v'_q$ が存在すると仮定すると，$2 \leq r \leq q$ に対しマッチングに使用されている辺 (v'_{r-1}, v_r) とマッチングに使用されていない辺 (v'_r, v_r) とを交換することで v_r は $s:1$ のマッチングを保存している．さらに，マッチング (v'_1, v_1) が存在することになりマッチング数が 1 増えるので，$s:1$ の最大マッチングに矛盾する．

よって補題は成立する． □

7 Zigzag 法

Zigzag 法は $s:1$ の最大マッチング $\mathcal{M}_s$ において，$N_s \neq \phi$ 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．

$L_0 := \phi$, $L'_0 := N_s$, $q := 0$
repeat
  $q := q + 1$
  $L_q := \{u | u \in pred(L'_{q-1}) \cap U_2\} - \bigcup_{j \leq q-1} L_j$
  $L'_q := \{u' | (u, u') \in \mathcal{M}_s \wedge u \in L_q\}$
until $L_q = \phi$
$L := \bigcup_{j \leq q} L_j$, $L' := \bigcup_{j \leq q} L'_j$

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．

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．

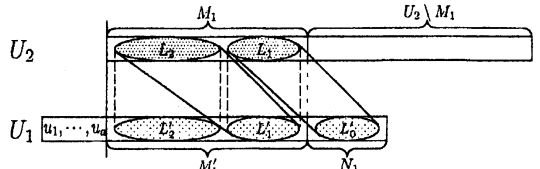

図 3: $s:1$ の最大マッチングにおける Zigzag 法 ($s=1$)

証明) 任意の $v_r \in L$ に対して $v_r \in L_r$ となる L_r が存在する．v_r に対し L_r の定義からある $v'_{r-1} \in L'_{r-1}$ が存在し，$(v_r, v'_{r-1}) \in E' - \mathcal{M}_s$ である．さらに，v'_{r-1} に対し L'_{r-1} の定義からある $v_{r-1} \in L_{r-1}$ が存在し，$(v_{r-1}, v'_{r-1}) \in \mathcal{M}_s$ である．同様の議論を繰り返すことにより，v_r から N_s に含まれる頂点への条件を満たす路が存在する．よって，補題は成立する． □

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

補題 3 $s:1$ の最大マッチングにおいて，$N_s \neq \phi$ のとき *Zigzag* 法により求められた集合 L に対し $L \cap (U_2 \setminus M_s) = \phi$ である．

証明) $L \cap (U_2 \setminus M_s) \neq \phi$ とすると，ある $v \in L \cap (U_2 \setminus M_s)$ が存在する．このとき，$v \in L$ より性質 *1* が成立し，かつ $v \in U_2 \setminus M_s$ であるので，N_s から v への *Alternating Path* が存在する．
よって，補題 *2* より $s:1$ の最大マッチングに矛盾するので，補題は成立する． □

8 近似アルゴリズム

本節では近似精度 $2 - \frac{1}{2c}$ 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 low-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 4 *([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 $u_1, u_2, \cdots, u_p$ とする．
補題 5, 6, 7 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 5 $k \leq e(v) < 1.8k$ のとき，すべての i $(0 \leq i < \frac{5}{13}k \wedge i \in N)$ とすべての x $(\frac{5}{9}(k+i) \leq x \leq k-1 \wedge x \in N)$ 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 $\frac{14}{9}$ 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．*)*

証明) *[3]* とほぼ同様の手法により証明可能．

補題 6 $k \leq e(v) < 1.8k$ のとき，すべての i $(\frac{5}{13}k \leq i < \frac{1}{2}k \wedge i \in N)$ とすべての x $(\frac{5}{9}(k+i) \leq x \leq k-1 \wedge x \in N)$ 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 $\frac{14}{9}$ 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．*)*

証明) *(i)* が成立しない場合，*(ii)* または *(iii)* が成立することを示す．$y = x+1 \leq k, a = \lceil \frac{14}{9}(k+i) - 2k \rceil < k$ とし，

- $e(u_{a+1}) < k$ のとき，u_{a+1} 以降を外部プロセッサでスケジュールし，通信後残りの a 個をメインプロセッサにスケジュールすることで，*(iii)* が成立する．
- $e(u_{a+1}) \geq k$ のときの証明を以下に示す．

1 : 1 の最大マッチングを $\mathcal{M}_1$ とする．

- $m_1 \leq k-i-1$ のとき，以下のように *[3]* と同様の手法でスケジュールを行う．u_y 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 $pred(N_1) \cup N_1$ をスケジュールし，通信を待って $M'_1, \{u_1, \cdots, u_a\}$ をスケジュールすれば，v のスケジュール時刻の上界は $e(u_y) + k + m_1 + a < \frac{14}{9} opt$ となり，*(iii)* が成立する．
- $m_1 > k-i-1$ のとき，
  - $N_1 \neq \phi$ のとき，N_1 を始点とし *Zigzag* 法で L, L' を求める．
    - $|L \cup L'| > k+i$ ならば，$U_2 \setminus L$ を外部プロセッサにスケジュールし時刻 $k+i$ までに通信する．L, U_1 はメインプロセッサに時刻 0 からスケジュールする．$L < x - a$ なので，v のスケジュールの上界は $|L \cup U_1| = (x-a) + x < \frac{14}{9} opt$ より *(iii)* が成立する．*(*図 *4)*
    - $|L \cup L'| \leq k+i$ ならば，$U_2 \setminus L$ を外部プロセッサにスケジュールし時刻 $k+i$ までに通信する．L, L' はメインプロセッサに時刻 0 からスケジュールし，$U_1 \setminus L'$ は時刻 $k+i$ 以

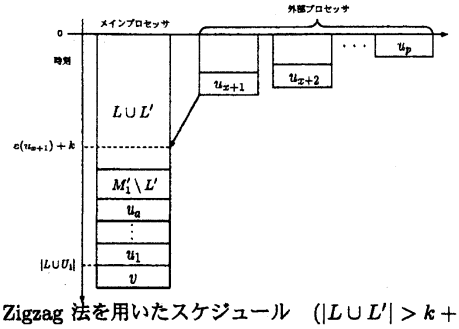

図 4: Zigzag 法を用いたスケジュール　($|L \cup L'| > k+i$)

降にスケジュールする．$e(u_x) \geq i+1$ より $|L| > i$ なので $|L'| > i$ であり，v のスケジュール時刻の上界は $k+i+(m_1-|L'|)+a < k+i+((k-1-a)-i)+a \leq \frac{14}{9}opt$ より *(iii)* が成立する．*(図 5)*

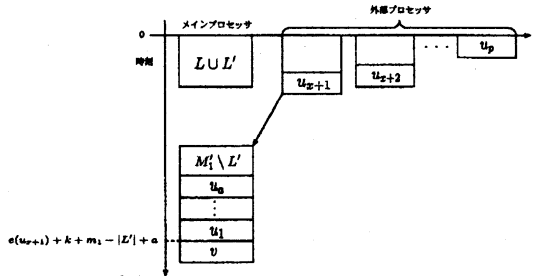

図 5: Zigzag 法を用いたスケジュール　($|L \cup L'| \leq k+i$)

- $N_1 = \phi$ のとき，
  - $x > \frac{k+2i}{2}$ ならば，
    - $\{u_1, \cdots, u_x\}$ 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 $e(u_x) \geq i+1$ より $e(u_x)+k \geq k+i+1$ 以降となる．
    - $\{u_1, \cdots, u_x\}$ を全てメインプロセッサにスケジュールする場合を考える．
      - B_1 のうち $|B_1|-(i-a)$ 個未満のタスクをメインプロセッサにスケジュールすると，通信を待って B_1' の要素 $(i-a)+1$ 個以上をメインプロセッサにスケジュールすることになるので，v のスケジュール時刻は $k+(i-a)+1+a = k+i+1$ 以降となる．
      - B_1 のうち $|B_1|-(i-a)$ 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 $|B_1|-(i-a)+x$ 以上となり，$|B_1|-(i-a)+x = (x-a)-i+a+x > k+i$ より v のスケジュール時刻は $k+i+1$ 以降となる．
  - $x \leq \frac{k+2i}{2}$ のときの証明を以下に示す．

2 : 1 の最大マッチングを $\mathcal{M}_2$ とする．

- $m_2 > -\frac{1}{9}k + \frac{17}{9}i - 2a + x$ ならば，
  - $\{u_1, \cdots, u_x\}$ 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 $e(u_x) \geq i+1$ より $e(u_x)+k \geq k+i+1$ 以降となる
  - $\{u_1, \cdots, u_x\}$ を全てメインプロセッサにスケジュールすると，
    - B_1 のうち $|B_1|-(i-a)$ 個未満または B_2 のうち $|B_2|-(i-a)$ 個未満のタスクをメインプロセッサにスケジュールすると，通信を待って $\bigcup_{j\geq 1} B_j'$ の要素 $(i-a)+1$ 個以上または $\bigcup_{j\geq 2} B_j'$ の要素 $(i-a)+1$ 個以上をメインプロセッサにスケジュールすることになるので，v のスケジュール時刻は $k+(i-a)+1+a = k+i+1$ 以降となる．
    - B_1 のうち $|B_1|-(i-a)$ 個以上かつ B_2 のうち $|B_2|-(i-a)$ 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 $|B_1|+|B_2|-2(i-a)+x$ 以上となり $|B_1|+|B_2|-2(i-a)+x = m_2 - 2(i-a)+x > k+i$ より v のスケジュール時刻は $k+i+1$ 以降となる．
- $m_2 \leq -\frac{1}{9}k + \frac{17}{9}i - 2a + x$ ならば，N_2 を始点とし *Zigzag* 法で L, L' を求める．$e(u_x) \geq i+1$ より $|L| > i, |L'| > \frac{1}{2}i$ である．
  但し，$m_2 \leq -\frac{1}{9}k + \frac{17}{9}i - 2a + x$ のとき $m_2 < 2(x-a)$ より，$\frac{13}{5}k \leq i < \frac{1}{2}k$ のとき N_2 は常に存在する．
  - $|L| > \frac{4}{9}k + \frac{22}{9}i - 2a$ のとき
    - $\{u_1, \cdots, u_x\}$ 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 $e(u_x) \geq i+1$ より $e(u_x)+k \geq k+i+1$ 以降となる．
    - $\{u_1, \cdots, u_x\}$ を全てメインプロセッサにスケジュールし，
      - L のうち $|L|-2(i-a)$ 個未満のタスクをメインプロセッサにスケジュールすると，L のうち $2(i-a)+1$ 個以上のタスクを外部プロセッサのみにスケジュールすることになり，通信を待って L' の要素 $(i-a)+1$ 個以上をメインプロセッサにスケジュールすることになるので，v のスケジュール時刻は $k+(i-a)+1+a = k+i+1$ 以降となる．
      - L のうち $|L|-2(i-a)$ 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 $|L|-2(i-a)+x$ 以上となり，$|L|-2(i-a)+x > \frac{4}{9}k + \frac{22}{9}i - 2a - 2i + 2a + \frac{5}{9}(k+i) = k+i$ より v のスケジュール時刻は $k+i+1$ 以降となる．
  - $|L| \leq \frac{4}{9}k + \frac{22}{9}i - 2a$ のとき，
    - $|L|+|L'| > k+i$ のとき，$U_2 \setminus L$ を外部プロ

セッサにスケジュールし時刻 $k+i$ までに通信する．L, U_1 はメインプロセッサに時刻 0 からスケジュールする．v のスケジュールの上界は $|L \cup U_1| \leq \frac{4}{9}k + \frac{22}{9}i - 2a + x \leq \frac{14}{9}opt$ となり，(iii) が成立する．

* $|L| + |L'| \leq k+i$ のとき，$U_2 \setminus L$ を外部プロセッサにスケジュールし時刻 $k+i$ までに通信する．L, L' はメインプロセッサに時刻 0 からスケジュールし，$U_1 \setminus L'$ は時刻 $k+i$ 以降にスケジュールすると v のスケジュールの上界は $k+i+|U_1 \setminus L'| \leq k+i+\frac{k+2i}{2} - \frac{1}{2}i < \frac{14}{9}opt$ となり，(iii) が成立する．

よって補題は成立する． □

補題 7 は $\frac{1}{2}k \leq i < \frac{11}{16}k$ と $\frac{11}{16}k \leq i < 0.8k$ の 2 部で構成されている．ただし，証明方法が同じなので 1 つの補題としてまとめた．

補題 7 $k \leq e(v) < 1.8k$ のとき，すべての i $(\frac{1}{2}k \leq i < 0.8k \wedge i \in N)$ とすべての x $(\frac{5}{9}(k+i) \leq x \leq k-1 \wedge x \in N)$ 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 $\frac{14}{9}$ 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．*)*

証明) 補題 *6* と同様の手法で証明可能．ただし，$\frac{1}{2}k \leq i < \frac{11}{16}k$ に関しては 3 : 1 の最大マッチング，$\frac{11}{16}k \leq i < 0.8k$ に関しては 4 : 1 の最大マッチングまで利用する．

補題 8 は 補題 5, 6, 7 を使用して証明する．

補題 8 $k \leq e(v) < 1.8k$ かつ $opt \geq k$ のとき，以下のいずれかが成立する．

- $opt \geq 1.8k$
- 近似精度 $\frac{14}{9}$ でスケジュール可能．

証明) $x = \lceil \frac{5}{9}k \rceil$ とする．

- $e(u_x) = 0$ のとき，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ことで近似精度 $\frac{14}{9}$ のスケジュールとなる．

$$\begin{aligned} e(u_x) + k + (x-1) &\leq \lceil \frac{14}{9}k \rceil - 1 \\ &< \frac{14}{9}opt \end{aligned}$$

- $e(u_x) \geq 1$ のとき，補題 *5* が適用できるので $(i=0)$，補題 *5* の $(i) \sim (iii)$ のいずれかが成立する．(i) が成立する場合には，x の値を 1 増やして補題 *5* を繰り返し適用することにより，$e(u_k) \geq 1, (ii), (iii)$ のいずれかが成立する．

$e(u_k) \geq 1$ 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 $k+1$ 以降になることを以下に示す．即ち，$opt > k+1$ が成立する．

- $\{u_1, \cdots, u_k\}$ 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 $k+1$ 以降となる．
- $\{u_1, \cdots, u_k\}$ を全てメインプロセッサにスケジュールすると，時刻 1 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 $k+1$ 以降となる．

一般に $opt \geq k+i, x = \lceil \frac{5}{9}(k+i) \rceil$ の場合を考える．ただし，$1 \leq i < 0.8k$．

- $e(u_x) \leq i$ のとき，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ことで近似精度 $\frac{14}{9}$ のスケジュールとなる．

$$\begin{aligned} e(u_x) + k + (x-1) &\leq \lceil \frac{14}{9}(k+i) \rceil - 1 \\ &< \frac{14}{9}opt \end{aligned}$$

- $e(u_x) \geq i+1$ のとき，補題 *5, 6, 7* の $(i) \sim (iii)$ のいずれかが成立する．(i) が成立する場合には，x の値を 1 増やして補題 *5, 6, 7* を繰り返し適用することにより，$e(u_k) \geq i+1, (ii), (iii)$ のいずれかが成立する．$e(u_k) \geq i+1$ 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 $k+i+1$ 以降になるので (ii) が成立する．

よって，$i = \lceil 0.8k \rceil$ のとき，$opt \geq 1.8k$ または近似精度 $\frac{14}{9}$ のスケジュールが得られる． □

補題 9 $1.8k \leq low(v) < 2k$ のとき，近似精度 $\frac{14}{9}$ でスケジュール可能．

証明) *[3]* とほぼ同様の手法により証明可能．

系 2 $k \leq low(v) < 2k$ のとき，近似精度 $\frac{14}{9}$ でスケジュール可能．

補題 10 $2k \leq low(v) < \frac{7}{3}k$ のとき，近似精度 $\frac{7}{4}$ でスケジュール可能．

証明) 定義より $low(u_k) + k \leq low(v)$ なので $low(u_k) < \frac{4}{3}k$ である．$k \leq low(v_k) < \frac{4}{3}k$ について補題 *5* と同様の手法で近似精度 $\frac{3}{2}$ で証明可能．

外部プロセッサに u_{k+1} 以降をスケジュールした後通信し，残りの k 個のタスクをメインプロセッサ上にスケジュールするとことで，近似精度 $\frac{7}{4}$ のスケジューリングが可能．

$$\begin{aligned} \frac{3}{2}low(u_k) + k + k &\leq \frac{3}{2}(low(u_k) + k) + \frac{1}{2}k \\ &< \frac{7}{4}low(v) \end{aligned}$$

□

補題 11 $\frac{7}{3}k \leq low(v) < 3k$ のとき，近似精度 $\frac{7}{4}$ でスケジュール可能.

証明) $k \leq low(v_k) < 2k$ について系 *2* から近似精度 $\frac{14}{9}$ でスケジュール可能であり，補題 *10* と同様の手法で証明可能. □

系 3 $2k \leq low(v) < 3k$ のとき，近似精度 $\frac{7}{4}$ でスケジュール可能.

定理 1 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 $2 - \frac{1}{2c}$ でスケジュール可能.

証明) c に関する帰納法で証明する.

- $c = 2$ のとき，即ち $2k \leq low(v) < 3k$ のときについては，系 *3* より成立する. $(\frac{7}{4} = 2 - \frac{1}{2 \cdot 2})$
- $c \geq 3$ のとき，定義より $k + low(u_k) \leq low(v)$，即ち $low(u_k) < ck$. 帰納法の仮定により，u_k 以降のタスクは $2 - \frac{1}{2(c-1)} low(u_k)$ 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，定理 *1* が成立する. *(図 6)*

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

□

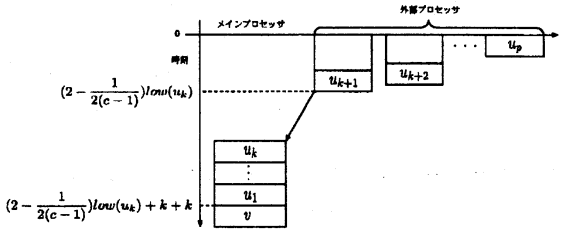

図 6: $c \geq 3$ の場合のスケジュール

9 本手法の計算量

本手法の計算量について述べる.

e-value の計算は $O(|V||E|)$ [2] ででき，v の先祖を e-value の降順にソートするのは $O(|V|^2)$ でできる. DAG から二部グラフを構成するのは $O(|V|^2)$ ででき，二部グラフの多対一の最大マッチングの計算は最大フローを求めるアルゴリズムを用いることにより $O(|V|^3)$ でできる. Zigzag 法によるタスクの集合 L, L' は高々 $O(|V|^3)$ で構成できる.

本手法では e-value を計算し，e-value の降順にソートした後，タスク v をスケジュールする. $e(v) < k$ の場合，各タスクは e-value の時刻にスケジュールできるので，合計で $O(|V|)$ となる. $k \leq e(v) < 1.8k$ の場合，各タスクのスケジュールでは二部グラフの構成・最大マッチング・Zigzag 法 を高々 k^2 回行うので $k \leq e(v) < 1.8k$ のすべてのタスクをスケジュールする計算量は高々 $O(k^2|V|^4)$ となる. $1.8k \leq e(v)$ の場合，すべてのタスクを low-value の降順にソートした後，タスク v をスケジュールする. $1.8k \leq e(v)$ の各タスクは $O(k)$ でスケジュールできるので，合計で $O(k|V|)$ となる.

よって，本手法の計算量は全体で $O(k^2|V|^4)$ となる.

10 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，多対一の最大マッチングを利用した Zigzag 法 による新たな手法を提案し，$c = 1$ における近似精度を改善するとともに，$c \geq 2$ において近似精度 $2 - \frac{1}{2c}$ のアルゴリズムを与えた. また，通信遅延による制約を除去した.

なお，証明等の詳細は [4] を参照されたい.

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 Journal on Computing, vol.19, no.2, pp.322–328, 1990.

[2] 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.

[3] 片柳賢二 砂坂明宏 大山口通夫 太田義勝, "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.

[4] 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.

[5] Dexter C. Kozen, "The Design and Analysis of Algorithms", Springer-Verlag, New York, 1992.

[6] John E. Hopcroft and Richard M. Karp, "$AN^{\frac{5}{2}}$ Algorithm for Maximum Matching in Bipartite Graphs", SIAM Journal on Computing, vol.2, No.4, pp.225–231, December, 1973.