数理解析研究所講究録
第1554巻 2007年 210-216

DAGの高さを4以下に制限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

An Approximation Algorithm for Task Scheduling of DAGs of Height not Exceeding Four

清水豪樹 大山口通夫 山田俊行 柳本貴之 小松健悟
(三重大学工学部)
Kouki SHIMIZU, Michio OYAMAGUCHI, Toshiyuki YAMADA,
Takayuki YANAGIMOTO, Kengo KOMATSU
(Faculty of Engineering, Mie University)

概要

Papadimitriouら (1990) はタスクの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. また, 小松ら (2006) によるDAGの高さを2に制限した問題に対する近似精度1.4のアルゴリズム, 清水ら (2006) によるDAGの高さを3に制限した問題に対する近似精度 $\frac{5}{3}(=1.66)$ のアルゴリズムがPapadimitriouらの結果を改善した結果である.

本研究ではDAGの高さを4に制限した問題に対して近似精度 $\frac{49}{26}(=1.88)$ 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のために通信遅延が発生する. タスク全体の処理時間を短縮するには, 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] であるため, 実行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. [2] では,DAGの高さが2, 各タスクの処理時間が1, 通信遅延時間が一定という条件下でも, この問題はNP完全であることを示した. [3] では, 各タスクの処理時間が1, 通信遅延時間が一定の場合に, 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, タスク間の依存関係をより詳細に解析し, 近似精度 $2-\frac{1}{1.67c}$ のアルゴリズムを示した. ここで c とはDAGの高さが大きくなるに従って大きくなる値であり, 詳細については [3] を参照されたい.

[4] では, タスクの処理時間, 通信遅延時間が共に可変のとき, 高さが2のDAGに対して近似精度1.4のアルゴリズムを示している. [5] では, 高さ3のDAGに対して, 最大マッチングを繰り返し適用することで近似精度 $\frac{5}{3}(=1.66)$ のアルゴリズムを示した.

本研究では, 高さ4のDAGに対して,[5] と同様に最大マッチングを繰り返し適用する方法をとるが, レベル3のタスクのスケジュール時刻を精密に計算することにより, 近似精度 $\frac{49}{26}(=1.88)$ のアルゴリズムを示す.

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, 複数のタスクと, 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, すべてのタスクの実行を最短時間内に終了する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に対する, 開始時刻と処理を行うプロセッサの割り当てである. なお, スケジュールは各タスクの開始時刻に非負整数を割り当てるものと仮定する.

非循環有向グラフ (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 (又はソース) 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. 本研究で用いる前提条件を以下に示す.

(1) 等しい機能を持つプロセッサを無制限に使用できる

(2) 各タスクの処理時間は正整数

(3) 各タスクの通信遅延時間は正整数

(4) タスクの複製を許す

この前提の下で, 高さ $L(L \leq 4)$ 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^* と表す. また, 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. $v \in V$ には処理時間 $x(v)$ と通信遅延時間 $\tau(v)$ があるが, v を $x(v)$ 個複製し, 通信遅延時間を $\tau(v) + x(v) - 1$ とすることで一般性を失うことなく処理時間を 1 と仮定できる (付録 A を参照). 従って, 以後各タスクの処理時間を 1 とする. 以下では, 集合 A の要素数を $|A|$ で表す.

定義 1 *([1])*
$v \in V$ のスケジュール時刻の下界 *e-value* $e(v)$ を以下で定める. また,*f-value* $f(v)$ を $f(v) = e(v) + x(v) + \tau(v)$ と定義する.

- v が *DAG* の根である場合, $e(v) = 0$.
- そうでない場合, 次のように計算する. v の先祖の集合を U と仮定する. $u \in U$ の *f-value* を求め,*f-value* の降順に並べたものを $u_1, u_2, \ldots, u_{|U|}$ とする. したがって, $f(u_1) \geq f(u_2) \geq \ldots \geq f(u_{|U|})$ が成立する. ここで, 整数 $j(f(u_k) > j \geq f(u_{k+1}))$ と, u_1 から u_k までのノードのうち, それらのノードのみを含んだ v へのパスが存在するノードで構成される部分 *DAG* $N_j(v)$ を考える. つまり, $N_j(v) = \{1 \leq i \leq k, \{u_1, \ldots, u_k\}$ の頂点のみを経由して, u_iから v へのパスが存在する. $\}$ である. $v_i \in \{N_j(v) \backslash v\}$ の $e(v_i)$ が $e(v_1) \geq e(v_2) \geq \ldots \geq e(u_l)$ とソートされていると仮定すると, $N_j(v)$ に含まれる全タスクの実行終了時刻の下界 L_j は $L_j = \max_{1 \leq i \leq l}(e(v_i) + i)$ である. このとき $j \geq L_j$ となる最小の j を $e(v)$ と定義する

このように計算された *e-value* $e(v)$ がタスク v の実行開始時間の下界となる証明については文献 *[1]* を参照されたい.

定義 2 v^* のスケジュール時刻の下界を *low* とし, *low* の初期値を $e(v^*)$ とする. また, v^* の最適スケジュール時刻を *opt* と表す. $low \leq opt$ が成立する.

定義 3 互いに素な頂点集合 U , U' において辺の集合 $A = E \cap (U \times U')$ のマッチング $\mathcal{M}$ とは, A の部分集合で, すべての頂点 $v \in U \cup U'$ について v を端点とする辺が高々1 つしかないものである.

定義 4 任意のマッチング $\mathcal{M}'$ に対して $|\mathcal{M}| \geq |\mathcal{M}'|$ が成り立つようなマッチング $\mathcal{M}$ を最大マッチングと呼ぶ.

3 近似アルゴリズム

本節では,DAG の高さが 4 の場合における近似精度 $\frac{49}{26}(= 1.88)$ のアルゴリズムを示す. 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 のすべての要素を下界 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 1 入力の *DAG* の高さが *0,1* のとき, 最適スケジュールが可能.

証明) *DAG* の高さが *0,1* のそれぞれについて, 最適スケジュールが可能であることを示す.

- 高さが 0 のとき, v^* をメインプロセッサに時刻 0 からスケジュールする.
- 高さが 1 のとき,*f-value* が *low* よりも大きいタスクを時刻 0 からメインプロセッサにスケジュールし, その他のタスクは時刻 0 から外部プロセッサにスケジュールする. v^* は時刻 *low* にメインプロセッサにスケジュールする. このスケジュールが最適であることは,*e-value* の定義より明らか.

□

頂点集合 U_1 を次のように定義する.

$$U_1 = \{u \mid f(u) > low\}$$

$$l = |U_1|$$

ここで, U_1 の各頂点は U_1 の頂点のみを経由して v^* に到達可能であると仮定する. また, U_1 の要素を e-value の降順に並べたものを $u_1, u_2, \ldots, u_l$ とする. したがって, $e(u_1) \geq e(u_2) \geq \ldots \geq e(u_l)$ が成立する. このとき, 定義 1 より, v^* の下界は $\max_{1 \leq i \leq l}(e(u_i) + i)$ である. ここで,

$$\lfloor 0.4e(u_j) \rfloor + j = \max_{1 \leq i \leq l}(\lfloor 0.4e(u_i) \rfloor + i)$$

と定義する.

補題 2 $\max(low + \lfloor 0.4e(u_j) \rfloor + j, \frac{5}{3} low)$ は v^* の上界となる.

証明) U_1 の要素 u 又は v^* の親 $w(f(w) \leq low)$ について考える.

(1) $level(w) = 0, 1$ のタスクは, 補題 *1* より, 最適なスケジューリングが可能なため, これらのタスクの実行結果は時刻 *low* にはメインプロセッサへの通信が完了している.

(2) $level(w) = 2$ のタスクは, 近似精度 1.4 でのスケジューリングが可能であり *[4]*, u はメインプロセッサに時刻 $\lfloor 1.4e(w) \rfloor + \tau(w) + 1$ 以降にスケジュール可能となる. ここで, $f(w) = e(w) + \tau(w) + 1 \leq low$, $e(w) \leq e(u)$ より, $\lfloor 1.4e(w) \rfloor + \tau(w) + 1 = f(w) + \lfloor 0.4e(w) \rfloor \leq low + \lfloor 0.4e(u) \rfloor$ が成立し, u のスケジュール時刻の上界は $low + \lfloor 0.4e(u) \rfloor$ となる.

(3) レベル 3 のタスクは，近似精度 $\frac{5}{3}$ でのスケジューリングが可能であるが *[5]*，この結果を必要とするタスクは v^* のみであり，v^* のスケジュール時刻の上界は $\frac{5}{3}low$ となる．

以上より，$u_1, u_2, \ldots, u_j$ を，時刻 $low + \lfloor 0.4e(u_j) \rfloor$ 以降に $u_j, u_{j-1}, \ldots, u_1$ と連続してスケジュールする．また，$u_l, u_{l-1}, \ldots, u_{j+1}$ を，時刻 $low + \lfloor 0.4e(u_j) \rfloor$ より前に $u_{j+1}, u_{j+2}, \ldots, u_l$ と連続してスケジュールする．任意の $u_i (1 \leq i \leq l)$ に対して，$\lfloor 0.4e(u_j) \rfloor + j \geq \lfloor 0.4e(u_i) \rfloor + i$ より，u_i のスケジュール時刻 $low + \lfloor 0.4e(u_j) \rfloor + j - i \geq low + \lfloor 0.4e(u_i) \rfloor$ が成立する．よって，u_i はそれぞれ u_j の前後に連続してスケジュール可能であり，$\max(low + \lfloor 0.4e(u_j) \rfloor + j, \frac{5}{3}low)$ は v^* の上界となる． □

このときの様子を図 1 に示す．

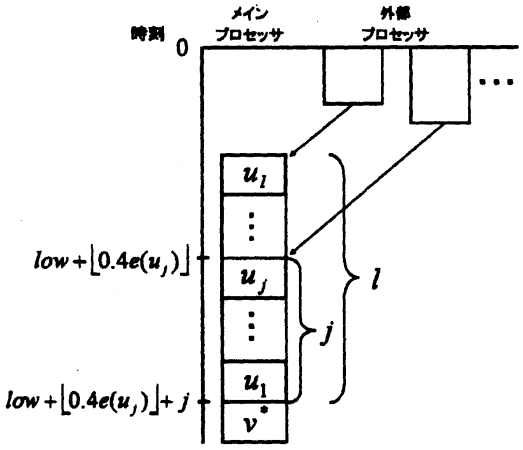

図 1: 補題 2 : v^* の上界 $low + \lfloor 0.4e(u_j) \rfloor + j$

補題 3

$$e(u_j) \geq \frac{5}{3}(low - \lfloor \alpha low \rfloor)$$
$$\alpha \geq \frac{2}{3}$$

のとき近似精度 $(1+\alpha)$ でスケジュール可能．

証明) 補題 *2* より，u_j を時刻 $low + \lfloor 0.4e(u_j) \rfloor$ にスケジュールするとき，v^* の上界は $low + \lfloor 0.4e(u_j) \rfloor + j$ となる．ここで，$\lfloor 0.4e(u_j) \rfloor + j > \lfloor \alpha low \rfloor$ と仮定すると $0.4e(u_j) + j \geq \lfloor 0.4e(u_j) \rfloor + j > \lfloor \alpha low \rfloor$，また仮定 $0.6e(u_j) \geq low - \lfloor \alpha low \rfloor$ より，$e(u_j) + j > low$ となり，$e(u_j) + j \leq low$ に矛盾する．よって，$low + \lfloor 0.4e(u_j) \rfloor + j \leq low + \lfloor \alpha low \rfloor$ となる．また，レベル 3 のタスクは近似精度 $\frac{5}{3}$ でスケジューリング可能であり *[5]*，これらのタスクの実行結果は時刻 $\frac{5}{3}low$ にはメインプロセッサへの通信が完了している．したがって，$\alpha \geq \frac{2}{3}$ ならば，v^* のスケジュール時刻の上界は $low + \lfloor \alpha low \rfloor$ となり，近似精度 $(1+\alpha)$ でスケジュール可能． □

以下では，$\alpha \geq \frac{2}{3}$ と仮定してスケジューリングを行う．

補題 4

$$e(u_j) < \frac{5}{3}(low - \lfloor \alpha low \rfloor)$$
$$l \leq \frac{5}{3}\lfloor \alpha low \rfloor - \frac{2}{3}low + 1$$

のとき近似精度 $(1+\alpha)$ でスケジュール可能．

証明) 補題 *3* と同様に，u_j を時刻 $low + \lfloor 0.4e(u_j) \rfloor$ にスケジュールするとき，v^* の上界は $low + \lfloor 0.4e(u_j) \rfloor + j$ となる．また，$j \leq l$ であるので，

$$low + \lfloor 0.4e(u_j) \rfloor + j \leq$$
$$low + \frac{2}{3}(low - \lfloor \alpha low \rfloor) + l - 1 \leq low + \lfloor \alpha low \rfloor$$

となり，近似精度 $(1+\alpha)$ でスケジュール可能． □

補題 3，4 のときの様子を図 2 に示す．

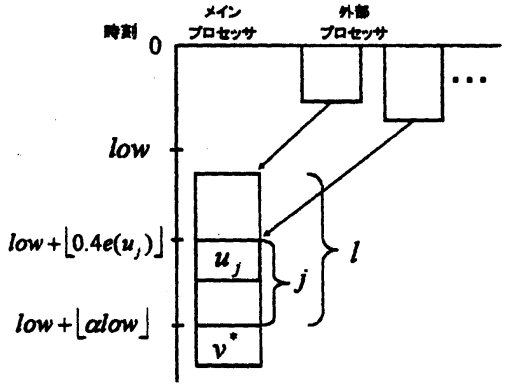

図 2: 補題 3，補題 4 : 近似精度 $(1+\alpha)$ のスケジュール

次に，整数 $t (0 \leq t < low)$ を考え，頂点集合 U_2 を以下のように定義する．

$$U_2 = \{u \mid level(u) = 0, low \geq f(u) > t\}$$

$\mathcal{M}_1$ を U_1 と U_2 の最大マッチングと定義する．

$$M_1 = \{u \mid (u, u') \in \mathcal{M}_1\},$$
$$M_1' = \{u' \mid (u, u') \in \mathcal{M}_1\}$$

$$m_1 = |M_1|, \quad N_1 = U_1 \setminus M_1'$$

このときの様子を図 3 に示す．

図 3 において，U_2 に含まれていて N_1 の先祖であるタスクは，M_1 にしか存在しない．よって，$M_1 \cup \{u | t \geq f(u), level(u) \leq 2\}$ に含まれるタスクの実行結果があれば，N_1 に含まれるタスクは実行を開始できる．

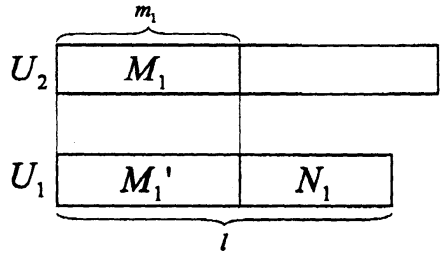

図 3: 最大マッチング $\mathcal{M}_1$

補題 5

$$l \geq \frac{5}{3}\lfloor \alpha low \rfloor - \frac{2}{3}low + 2$$
$$m_1 > 2low - t - l - 1$$

のとき，$opt \geq low + 1$ が成立する．

証明) 最適スケジュールを S とする．S が U_1 のあるタスクを外部プロセッサに割り当てると，明らかに $opt \geq low+1$．従って，U_1 のすべてのタスクをメインプロセッサに割り当てる場合について考える．次に，M_1 の幾つかのタスクをメインプロセッサにスケジュールする場合と，外部プロセッサにスケジュールする場合に分けて考える．

- $M_1'' \subseteq M_1$ を時刻 0 からメインプロセッサにスケジュールする場合，メインプロセッサには M_1'' 以外に U_1 のタスクを l 個スケジュールする必要があるので，$|M_1''| > low - l$ であれば $opt \geq low + 1$ となる．
- $M_1'' \subseteq M_1$ を外部プロセッサにスケジュールする場合，M_1'' のマッチング相手がメインプロセッサに時刻 $t+1$ 以降にスケジュールされるため，$|M_1''| > low - t - 1$ であれば $opt \geq low + 1$ となる．

以上より，$m_1 > low - l + low - t - 1 = 2low - t - l - 1$ であれば $opt \geq low + 1$ が成立する． □

さらに，頂点集合 U_3 を以下のように定義する．

$$U_3 = \{u \mid 2 \geq level(u) \geq 1, low \geq f(u) > t\}$$

また，$\mathcal{M}_2$ を U_3 と N_1 の最大マッチングと定義する．

$$M_2 = \{u \mid (u, u') \in \mathcal{M}_2\},$$
$$M_2' = \{u' \mid (u, u') \in \mathcal{M}_2\}$$

$$m_2 = |M_2|, \quad N_2 = N_1 \setminus M_2'$$

このときの様子を図 4 に示す．

図 4 において，$U_2 \cup U_3$ に含まれていて N_2 の先祖であるタスクは，$M_1 \cup M_2$ にのみ存在する．よって，$M_1 \cup M_2 \cup \{u | 2 \geq level(u) \geq 0, t \geq f(u)\}$ に含まれるタスクの実行結果があれば，N_2 に含まれるタスクは実行を開始できる．また，U_2 に含まれていて M_2 の先祖であるタスクは，M_1 にのみ存在する．よって，$M_1 \cup \{u | 1 \geq level(u) \geq 0, t \geq f(u)\}$ に含まれるタスクの実行結果があれば，M_2 に含まれるタスクは実行を開始できる．

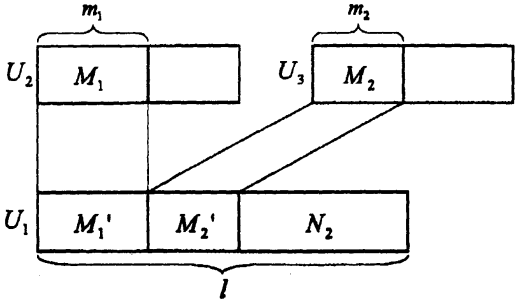

図 4: 最大マッチング $\mathcal{M}_2$

補題 6

$$l \geq \frac{5}{3}\lfloor \alpha low \rfloor - \frac{2}{3}low + 2$$
$$m_1 + m_2 > 2low - t - l - 1$$

のとき，$opt \geq low + 1$ が成立する．

証明) 補題 5 の M_1'' の定義を，$M_1'' \subseteq M_1 \cup M_2$ とすることで，補題 5 と同様に証明できる． □

補題 7

$$l \geq \frac{5}{3}\lfloor \alpha low \rfloor - \frac{2}{3}low + 2$$
$$m_1 + m_2 \leq 2low - t - l - 1$$
$$m_2 \leq \lfloor \alpha low \rfloor + low - t - l$$

のとき，以下の式が成立するならば近似精度 $(1+\alpha)$ でスケジュール可能．

$$2low - t - l - 1 \leq \lfloor \alpha low \rfloor + low - l \quad (1)$$
$$\lfloor \alpha low \rfloor + low - l > \lfloor 1.4t \rfloor \quad (2)$$
$$|N_2| \geq l - \lfloor \alpha low \rfloor + \lfloor \frac{2}{3}(low - \lfloor \alpha low \rfloor) \rfloor - 1 \quad (3)$$

証明) $M = M_1' \cup M_2'$ と定義し，M に含まれるタスクを $e\text{-}value$ の降順に並べたものを $w_1, w_2, \ldots, w_{|M|}$ とする．したがって，$e(w_1) \geq e(w_2) \geq \ldots \geq e(w_{|M|})$ が成立する．$k(1 \leq k < |M|)$ を次の式を満たすものとする．

$$\lfloor 0.4e(w_k) \rfloor + k = \max_{1 \leq i \leq |M|} (\lfloor 0.4e(w_i) \rfloor + i)$$

- M_1 を時刻 0 からメインプロセッサにスケジュールする．

- $m_1 > t$ のとき，M_2 は時刻 m_1 からメインプロセッサにスケジュールする．
- $m_1 \leq t$ のとき，M_2 は時刻 t からメインプロセッサにスケジュールする．

式 *(1)* より，$m_1 + m_2 \leq 2low - t - l - 1 \leq \lfloor \alpha low \rfloor + low - l$，かつ $m_2 + t \leq \lfloor \alpha low \rfloor + low - l$ なので，M_1 と M_2 は必ず時刻 $\lfloor \alpha low \rfloor + low - l$ 以前にメインプロセッサにスケジュールできる．また，M_2 の親は M_1 か $\{u | 1 \geq level(u) \geq 0, t \geq f(u)\}$ である外部プロセッサにスケジュールされたタスクのどちらかであり，後者のタスクの場合はその実行結果は時刻 t までにメインプロセッサへの通信が完了している．

- N_2 を時刻 $\lfloor \alpha low \rfloor + low - l$ からメインプロセッサにスケジュールする．N_2 の先祖は $M_1 \cup M_2 \cup \{u | 2 \geq level(u) \geq 0, t \geq f(u)\}$ に含まれている．このうち，外部プロセッサにスケジュールされたレベル 2 のタスクの実行結果は時刻 $\lfloor 1.4t \rfloor$ までにはメインプロセッサへの通信が完了しているので，式 *(2)* より，N_2 は必ず時刻 $\lfloor \alpha low \rfloor + low - l$ からメインプロセッサにスケジュールできる．

- M を N_2 の次にメインプロセッサにスケジュールする．式 *(3)* より，$\lfloor \alpha low \rfloor + low - l + |N_2| \geq low + \lfloor \frac{2}{3}(low - \lfloor \alpha low \rfloor) \rfloor - 1$ となり，M の開始時刻は必ず $low + \lfloor \frac{2}{3}(low - \lfloor \alpha low \rfloor) \rfloor - 1$ 以降となる．また，以下の *(1)*, *(2)* より M は必ず時刻 $low + \lfloor \alpha low \rfloor$ 以前にスケジュール可能である．

(1) $e(w_k) \geq \frac{5}{3}(low - \lfloor \alpha low \rfloor)$ のとき，M の要素 $w_i (1 \leq i \leq |M|)$ は時刻 $low + \lfloor 0.4e(w_i) \rfloor$ 以降にスケジュールする．このとき，

$$\begin{aligned} low &\geq e(w_k) + k \\ &\geq 0.6e(w_k) + 0.4e(w_i) + i \\ &\geq (low - \lfloor \alpha low \rfloor) + 0.4e(w_i) + i \\ low + \lfloor \alpha low \rfloor &\geq low + \lfloor 0.4e(w_i) \rfloor + i \end{aligned}$$

となり，M は必ず時刻 $low + \lfloor \alpha low \rfloor$ 以前にスケジュール可能．

(2) $e(w_k) < \frac{5}{3}(low - \lfloor \alpha low \rfloor)$ のとき，M の要素 $w_i (1 \leq i \leq |M|)$ は時刻 $low + \lfloor \frac{2}{3}(low - \lfloor \alpha low \rfloor) \rfloor - 1$ 以降にスケジュールする．このとき，w_i のスケジューリング時刻は

$$\begin{aligned} &low + \lfloor \frac{2}{3}(low - \lfloor \alpha low \rfloor) \rfloor - 1 + |M| - i \\ &\leq low + \lfloor \frac{2}{3}(low - \lfloor \alpha low \rfloor) \rfloor - 1 + |M| \\ &= low + \lfloor \frac{2}{3}(low - \lfloor \alpha low \rfloor) \rfloor - 1 + l - |N_2| \\ &\leq low + \lfloor \alpha low \rfloor \end{aligned}$$

となり，M は必ず時刻 $low + \lfloor \alpha low \rfloor$ 以前にスケジュール可能．

- 残りのタスクは全て外部プロセッサにスケジュールする．

このようにスケジュールすることで，v^* のスケジュール時刻の上界は $low + \lfloor \alpha low \rfloor$ となり，近似精度 $(1 + \alpha)$ でスケジュール可能．

□

このときの様子を図 5 に示す．

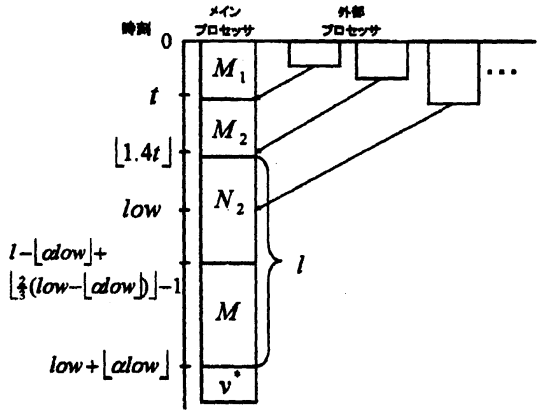

図 5: 補題 7：近似精度 $(1 + \alpha)$ のスケジュール

補題 7 の条件 (1)〜(3) を満たす α の値を後で示す．次に，$m_2 \leq \lfloor \alpha low \rfloor + low - t - l$ の場合について説明する．さらに，$\mathcal{M}_3$ を M_2 と N_2 の最大マッチングと定義する．

$$M_3 = \{u \mid (u, u') \in \mathcal{M}_3\}, \quad M_3' = \{u' \mid (u, u') \in \mathcal{M}_3\}$$

$$m_3 = |M_3|, \quad N_3 = N_2 \setminus M_3'$$

このときの様子を図 6 に示す．

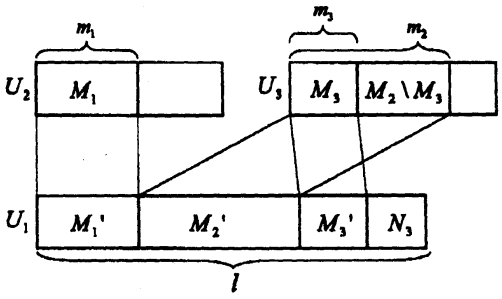

図 6: 最大マッチング $\mathcal{M}_3$

補題 8

$$l \geq \frac{5}{3}\lfloor \alpha low \rfloor - \frac{2}{3} low + 2$$
$$m_1 + m_2 \leq 2low - t - l - 1$$
$$m_2 > \lfloor \alpha low \rfloor + low - t - l$$
$$m_3 \leq \lfloor \alpha low \rfloor + low - t - l$$
$$|N_3| \geq l - \lfloor \alpha low \rfloor + \lfloor \frac{2}{3}(low - \lfloor \alpha low \rfloor) \rfloor - 1$$

のとき，近似精度 $(1+\alpha)$ でスケジュール可能.

証明) $M_2 \setminus M_3$ に含まれるタスクは N_3 と依存関係はなく，これらは外部プロセッサにスケジュールする．そして，補題 *7* の M_2 を M_3 とし，$M = M_1' \cup M_2' \cup M_3'$ とすることで，補題 *7* と同様に証明できる. □

補題 9

$$l \geq \frac{5}{3}\lfloor \alpha low \rfloor - \frac{2}{3} low + 2$$
$$m_1 + m_2 \leq 2low - t - l - 1$$
$$m_2 > \lfloor \alpha low \rfloor + low - t - l$$
$$m_3 \leq \lfloor \alpha low \rfloor + low - t - l$$
$$|N_3| < l - \lfloor \alpha low \rfloor + \lfloor \frac{2}{3}(low - \lfloor \alpha low \rfloor) \rfloor - 1$$

のとき，以下の式が成立するならば $opt \geq low + 1$ が成立する.

$$t \geq 3low - 2l - \lfloor \alpha low \rfloor + \lfloor \frac{2}{3}(low - \lfloor \alpha low \rfloor) \rfloor - 2 \quad (4)$$

証明) M_1，M_3，$M_2 \setminus M_3$ に含まれるタスクの，子の数を考える．M_3 に含まれるタスクは，M_2' と M_3' にそれぞれ 1 つ以上の子を持つ．ここで，M_3 に含まれるタスクを $low - l$ 個メインプロセッサにスケジュールし，その他のタスクを外部プロセッサに割り当てる．このとき，

$$2(m_3 - (low - l)) + m_1 + (m_2 - m_3) > low - (t+1)$$
$$m_1 + m_2 + m_3 > 3low - 2l - t - 1$$
$$t \geq 3low - 2l - \lfloor \alpha low \rfloor + \lfloor \frac{2}{3}(low - \lfloor \alpha low \rfloor) \rfloor - 2$$

が成立すれば，$opt \geq low + 1$ が成立する. □

補題 10

$$l \geq \frac{5}{3}\lfloor \alpha low \rfloor - \frac{2}{3} low + 2$$
$$m_1 + m_2 \leq 2low - t - l - 1$$
$$m_2 > \lfloor \alpha low \rfloor + low - t - l$$
$$m_3 > \lfloor \alpha low \rfloor + low - t - l$$

のとき，以下の式が成立するならば $opt \geq low + 1$ が成立する.

$$t \leq 2\lfloor \alpha low \rfloor - low + 1 \quad (5)$$

証明) 補題 *9* と同様のスケジューリングを行い，$m_1 = 0$ の場合を考えると，$m_2 + m_3 > 3low - 2l - t - 1$ であり，仮定より $m_2 + m_3 > 2(\lfloor \alpha low \rfloor + low - t - l)$ であるので，

$$2(\lfloor \alpha low \rfloor + low - t - l) \geq 3low - 2l - t - 1$$
$$1 - low + 2\lfloor \alpha low \rfloor \geq t$$

が成立すれば，$opt \geq low + 1$ が成立する. □

補題 11 $\alpha \geq \frac{23}{26}$ のとき，*(2)⇒(5)*.

証明) 式 *(2)*，式 *(5)* の左辺を比較すると，$\alpha \geq \frac{23}{26}$ のとき，

$$1.4\ (\ 2\lfloor \alpha low \rfloor - low + 1) - (\lfloor \alpha low \rfloor + low - l - 1) + 1$$
$$> \frac{2}{15} low(26\alpha - 23) + \frac{14}{15} > 0$$

となる．従って，

$$1.4(2\lfloor \alpha low \rfloor - low + 1) \geq (\lfloor \alpha low \rfloor + low - l - 1)$$
$$\geq \lfloor 1.4t \rfloor + 1 > 1.4t$$

であり，補題 *11* は成立する. □

補題 12 $\alpha \geq \frac{23}{26}$ のとき，近似精度 $(1+\alpha)$ でスケジュール可能，または $opt \geq low + 1$ が成立する.

証明) 補題 *3*，補題 *4*，補題 *10* より，補題 *7* の式 *(1)*，式 *(2)*，式 *(3)*，補題 *9* の式 *(4)*，補題 *10* の式 *(5)* が成立する場合に，近似精度 $(1+\alpha)$ でスケジュール可能，または $opt \geq low + 1$ が成立する．補題 *11* より，*(2)⇒(5)*．また，*(4)⇒(1)∧(3)* であるため，式 *(2)*，式 *(4)* について考える．この 2 式を満たす α を求めると，

$$\frac{5}{7}(\lfloor \alpha low \rfloor + low - l) - 1$$
$$> 3low - 2l - \lfloor \alpha low \rfloor + \lfloor \frac{2}{3}(low - \lfloor \alpha low \rfloor) \rfloor - 2$$
$$\alpha low > \frac{16}{19}(low - \frac{19}{20})$$
$$\alpha > \frac{16}{19}$$

となり，$\frac{23}{26} > \frac{16}{19}$ より，$\alpha \geq \frac{23}{26}$ のとき，近似精度 $(1+\alpha)$ でスケジュール可能，または $opt \geq low + 1$ が成立する. □

定理 1 入力の *DAG* の高さが 4 のとき，近似精度 $\frac{49}{26}$ のアルゴリズムが存在する.

証明) 補題 *12* より，$\alpha = \frac{23}{26}$ のとき，近似精度 $\frac{49}{26}$ でスケジュール可能，または $opt \geq low + 1$ が成立する．後者の場合は，low の値を 1 増やしたものを新しい low とし，補題 *12* を繰り返し適用することで，定理 *1* は成立する. □

4 本手法の時間計算量

本手法の時間計算量について述べる.

e-value の計算は $O(|V|^2(|E|+|V|\log|V|))$ で実行でき, DAG の変換は $O(|V|)$ でできる. 頂点集合 U_1, U_2, U_3 をとるのにそれぞれ $O(|V|)$ かかり, 二つの頂点集合の最大マッチングを計算するのは $O(|V||E|)$ でできる. 下界が改善され, 再帰的に本手法が適用される回数は高々 $O(|V|)$ 回であり, 全体として $O(|V|^2(|E|+|V|\log|V|))$ となる.

5 おわりに

本研究では, DAG の高さが 4 の場合において Papadimitriou ら [1] の近似精度 2 よりも良い近似精度 $\frac{49}{26}$ を持つ近似アルゴリズムを与えた. 本アルゴリズムでは, これまでの高さ 3 以下の DAG に対するスケジューリングでは問題とならなかった, メインプロセッサに割り当てた v^* 以外のタスクに対する, 高さ 2 のタスクからの通信を考慮したスケジューリング手法を考案した.

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] 河田俊郎, 大山口通夫, 大田義勝, "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.

[3] 新美信之助 大山口通夫 太田義勝 山本浩平, "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", 京大数解研講究録, No.1426, pp.184–188, 2005.

[4] 小松健悟, "タスクスケジューリングアルゴリズム関する研究–DAG の高さ 2 の場合–", 三重大学, 修士論文, 2006.

[5] 清水豪樹, 大山口通夫, 山田俊行, "DAG の高さを 3 に制限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", 2006 年度電気関係学会東海支部連合大会 講演論文集,O–439, 2006.

A DAG の変換

与えられた DAG において, V の要素である処理時間が $x(v)(>1)$ のタスクを, 処理時間が 1 である $x(v)$ 個のタスクに並列に分割し, 分割したそれぞれのタスクの通信遅延時間を $\tau(v)+x(v)-1$ にすることで, 全てのタスクの処理時間を 1 にする.

図 7 に DAG の変換の例を示す.

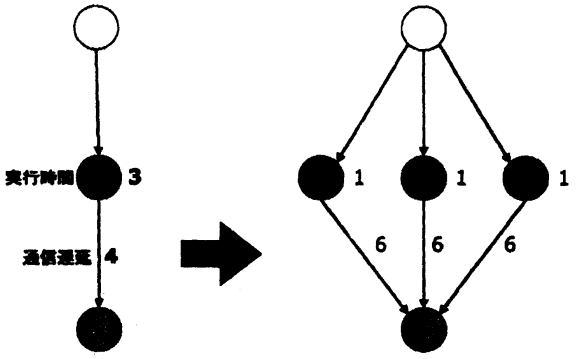

図 7: DAG の変換

図 7 は, 実行時間が 3, 通信遅延時間が 4 のタスクの変換の例である. 変換後のタスクの実行時間が 1 になり, 通信遅延時間は $3+4-1=6$ となる.

補題 13 変換後の *DAG* を入力として得られた各タスクのスケジュール時刻で, 変換前の *DAG* のスケジュールが可能である.

証明) タスクの処理時間を $x(v)(>1)$ とし, v を分割してできた処理時間が *1* のタスクを $W=\{w_1,\cdots,w_{x(v)}\}$ と定義する. $w\in W$ の *e-value* は $e(w)=e(v)$ になる.

- $w\in W$ が全てメインプロセッサにスケジュールされた場合
  最も早い時刻にスケジュールされた $w\in W$ を v のスケジュールにする. 連続していないときは,w 以外の頂点 $w'\in W$ はスケジュールから削除する. この結果, W に属さない頂点で w 以降に割り当てられたものは, 元のスケジュール時刻以降にスケジュールされることが保証される.

- $w\in W$ が全て外部プロセッサにスケジュールされた場合
  任意の w を選び, それを v スケジュールにする. w と v の *f-value* は同じなので, v の子のスケジュールには影響を与えない.

- $w\in W$ がメインプロセッサと外部プロセッサににスケジュールされた場合
  外部プロセッサにスケジュールされたタスクの中から任意の w を選び, それを v のスケジュールにする.

□

よって, 各タスクの処理時間を 1 と仮定した場合におけるタスクスケジューリングアルゴリズムは, 各タスクの処理時間を正整数とした場合に拡張できる.