A-016

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* 紅林清志*
Kouki Shimizu , Michio Oyamaguchi , Toshiyuki Yamada , Kiyoshi Kurebayashi

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（非循環有向グラフ）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 の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．[1] と同じ前提の下で，[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．

また，[3] では，タスクの処理時間，通信遅延時間に関する制約条件が無い場合に，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，[4] では，高さ 3 の DAG に対して，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，近似精度 $\frac{49}{26}(=1.88)$ のアルゴリズムを与えることにより，[1] の結果を改善する．このアルゴリズムで用いる手法の主なものは [4] 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，(高さ 3 のときには考慮する必要がなかった) レベル 3 のタスク集合に対する新しいスケジューリング手法を発見することにより，本研究の結果を得ることができた．

*三重大学工学部，Faculty of Engineering，Mie University

2 準備

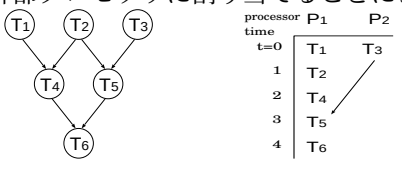
本研究では，高さ $L(L \leq 4)$ の DAG $G=(V,E)$ をスケジューリングの対象とする．ここで，V はタスクの集合，E はタスク間の依存関係を表す有向辺の集合である．DAG G，各タスクの処理時間，及び各プロセッサ間の通信遅延時間が与えられたときに，すべてのタスクの処理を最短時間内に終了するスケジュールを求める問題について考察する．ここでスケジュールとは各タスクに対する，開始時刻と処理を行うプロセッサの割り当てである．スケジュールは各タスクの開始時刻に非負整数を割り当てるものとし，各 $(u,v) \in E$ に対して，タスク v の割り当て時刻はタスク u の終了時刻 (+ 通信遅延) 以降でなければならない．

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．

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 のソース（又は根）という．$v \in V$ のレベルとは G のソースから 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．

前提条件 (1)，(4) より，最大レベル 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．

例. 下図の DAG に対し，各タスク $T_1 \sim T_6$ の実行時間を 1，各タスク間の通信遅延を 2 とする．この最適スケジューリングの一つは，メインプロセッサに時刻 0 から順に $T_1T_2T_4T_5T_6$ を割り当て，T_3 を外部プロセッサに割り当てることにより得られる．



DAG　　最適スケジューリング

各タスク $v \in V$ の処理時間と通信遅延時間をそれぞれ $x(v)$ と $\tau(v)$ と書く．

定義 1 $v \in V$ のスケジュール時刻の下界 e-値 $e(v)$ を以下で定める．また，f-値 $f(v)$ 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-値を求め，f-値の降順に並べたものを $u_1, u_2, \cdots, u_{|U|}$ とする．したがって，$f(u_1) \geq f(u_2) \geq \cdots \geq f(u_{|U|})$ 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，$N_j(v) = \{1 \leq i \leq k, \{u_1, \cdots, u_k\}$ 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 $e(v_1) \geq e(v_2) \geq \cdots \geq e(v_l)$ 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 $L_j = \max_{1 \leq i \leq l}(e(v_i) + \sum_{q=1}^{i} x(v_q))$ 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-値 $e(v)$ 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．

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[1][3]．定義 1 の e-値と f-値 も同じである．従って，以後各タスクの処理時間を 1 とする．

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．

以下では，集合 A の要素数を $|A|$ で表す．

定義 3 M がグラフ $G(V, E)$ のマッチングとは，$M \subseteq E$ で，M のどの 2 本の辺も共通の端点を持たないときである．$|M|$ が最大であるマッチング M を G の最大マッチングと呼ぶ．V が明らかなときには M を E の最大マッチングと呼ぶ．

以後，辺集合 E は次の条件をみたすとする．

$$\forall (u, v), (v, w) \in E \Rightarrow (u, w) \in E$$

この条件は 3 章の結果を示すために必要である．

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．高さ 2 および 3 の DAG に対するアルゴリズムについては紙面の都合上省略する．詳細については，それぞれ文献 [3]，[4] を参照されたい．

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-値[2] 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．また，近似精度を $(1 + \alpha)$ と仮定してスケジューリングを行い，$\alpha \geq \frac{23}{26}$ ならば常にスケジュール可能であることを示す．

[1]変換前と変換後のタスクの最適開始時刻は一致する．したがって，一方の最適開始時刻の下界は他方の最適開始時刻の下界にもなる．

[2]正確には，[3], [4] で定義される下界値であるが，議論の簡単化のため，e-値と同じものとする．

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) 明らか (文献 [3] 参照) □

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-値の降順に $e(u_1) \geq e(u_2) \geq \ldots \geq e(u_l)$ と並べる．このとき，$j(1 \leq j \leq l)$ は次の式をみたすものとする．

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

なお，$\max_{1 \leq i \leq l}(e(u_i) + i) \leq low$ である．(そうでないならば，v^* の下界を改善できる．) 従って，$l \leq low$.

補題 2 $\max(low + \lfloor 0.4e(u_j) \rfloor + j, \lfloor \frac{5}{3} low \rfloor)$ は v^* のスケジュール時刻の上界となる．

証明) 文献 [1] の手法とほぼ同様の考え方を用いて，この補題 *2* を示すことができる（付録 A 参照）．□

補題 3 $low - e(u_j) + \lfloor 0.4e(u_j) \rfloor \leq \lfloor \alpha low \rfloor$ 及び $\alpha \geq \frac{2}{3}$ のとき近似精度 $(1 + \alpha)$ でスケジュール可能．

証明) $\alpha \geq \frac{2}{3}$ と補題 *2* より，$low + \lfloor 0.4e(u_j) \rfloor + j \leq low + \lfloor \alpha low \rfloor$ を示せば十分である．この不等式は $e(u_j) + j \leq low$ と補題の前提条件より明らか． □

以下では，次の条件を仮定する．

$$1 > \alpha \geq \frac{2}{3} \quad \text{かつ}$$

$$low - e(u_j) + \lfloor 0.4e(u_j) \rfloor \geq \lfloor \alpha low \rfloor + 1 \qquad (*1)$$

補題 4 $3l < 5\lfloor \alpha low \rfloor - 2low + 5$ のとき近似精度 $(1 + \alpha)$ でスケジュール可能．

証明) $(*1)$ より，$\frac{2}{3}(low - \lfloor \alpha low \rfloor - 1) \geq \frac{2}{3}(e(u_j) - \lfloor 0.4e(u_j) \rfloor) \geq \lfloor 0.4e(u_j) \rfloor$．$j \leq l$ より，$\lfloor 0.4e(u_j) \rfloor + j < \lfloor \alpha low \rfloor + 1$．従って，$low + \lfloor 0.4e(u_j) \rfloor + j \leq low + \lfloor \alpha low \rfloor$ □

以下では，次の条件を仮定する．

$$3l \geq 5\lfloor \alpha low \rfloor - 2low + 5 \qquad (*2)$$

次に，整数 $t(0 \leq t < low)$ を考え，U_2，U_3 を以下のように定義する．$U_2 = \{u \mid level(u) = 0, low \geq f(u) > t\}$，$U_3 = \{u \mid 2 \geq level(u) \geq 1, low \geq f(u) > t\}$.

$\mathcal{M}_1$ を $E \cap (U_2 \times U_1)$ の最大マッチングと定義する．$M_1 = \{u \mid (u, u') \in \mathcal{M}_1\}$ ， $M_1' = \{u' \mid (u, u') \in \mathcal{M}_1\}$ ， $m_1 = |M_1|$ ， $N_1 = U_1 \setminus M_1'$.

N_1 に含まれるタスクは $M_1 \cup \{u \mid t \geq f(u)$ 又は $low \geq f(u)$ かつ $1 \leq level(u) \leq 2\}$ に含まれるタスクの実行結果があれば，実行を開始できる．[3]

[3]正確には，そのように最大マッチング $\mathcal{M}_1$ を選ぶことができる．(即ち，N_1 中のあるタスク v が M_1' 中に親 u を持つならば交換することができる．)

補題 5 $m_1 > 2low-t-l-1$ 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．$M_1'' \subseteq M_1$ 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．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，$\mathcal{M}_2$ を $E \cap (U_3 \times N_1)$ の最大マッチングと定義する．$M_2 = \{u \mid (u,u') \in \mathcal{M}_2\}$，$M_2' = \{u' \mid (u,u') \in \mathcal{M}_2\}$，$m_2 = |M_2|$，$N_2 = N_1 \setminus M_2'$．

N_2 に含まれるタスクは，$M_1 \cup M_2 \cup \{u \mid 2 \geq level(u) \geq 0, t \geq f(u)\}$ に含まれるタスクの実行結果があれば，実行を開始できる．また，M_2 に含まれるタスクは $M_1 \cup \{u \mid 1 \geq level(u) \geq 0, t \geq f(u)\}$ に含まれるタスクの実行結果があれば，実行を開始できる．[4]

補題 6 $m_1+m_2 > 2low-t-l-1$ 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．□

以下では，次の条件を仮定する．

$$m_1 + m_2 \leq 2low - t - l - 1 \qquad (*3)$$

補題 7 $m_2 \leq \lfloor \alpha low \rfloor + low - t - l$ 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．

$$2low - t - l - 1 \leq \lfloor \alpha low \rfloor + low - l \qquad (1)$$

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

$$m_1 + m_2 \leq \frac{5}{3}\lfloor \alpha low \rfloor - \frac{2}{3} low + 1 \qquad (3)$$

証明) 付録 B 参照．□

補題 7 の条件 (1)〜(3) をみたす α の値を後で示す．次に，$m_2 \geq \lfloor \alpha low \rfloor + low - t - l + 1$ の場合について説明する．さらに，$\mathcal{M}_3$ を $E \cap (M_2 \times N_2)$ の最大マッチングと定義する．$M_3 = \{u \mid (u,u') \in \mathcal{M}_3\}$，$M_3' = \{u' \mid (u,u') \in \mathcal{M}_3\}$，$m_3 = |M_3|$，$N_3 = N_2 \setminus M_3'$．

補題 8 $m_2 \geq \lfloor \alpha low \rfloor + low - t - l + 1$

$$m_3 \leq \lfloor \alpha low \rfloor + low - t - l$$

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

のとき，近似精度 $(1+\alpha)$ でスケジュール可能．

証明) 補題 *7* の証明の M_2 と N_2 をそれぞれ M_3 と N_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 $m_2 \geq \lfloor \alpha low \rfloor + low - t - l + 1$

$$m_3 \leq \lfloor \alpha low \rfloor + low - t - l$$

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

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

$$t \geq \frac{11}{3} low - 2l - \frac{5}{3}\lfloor \alpha low \rfloor - 2 \qquad (4)$$

証明) 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，$2(m_3-(low-l))+m_1+(m_2-m_3) > low-t-1$，即ち，式 *(4)* が成立すれば，$opt \geq low+1$ が成立する．□

補題 10 $m_2 \geq \lfloor \alpha low \rfloor + low - t - l + 1$

$$m_3 \geq \lfloor \alpha low \rfloor + low - t - l + 1$$

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

$$t < 2\lfloor \alpha low \rfloor - low + 3 \qquad (5)$$

証明) 補題 *9* と同様に証明できる．□

次の 3 つの補題は簡単な算術計算によって得ることができる．

補題 11 (4) $\Rightarrow$ (1) $\wedge$ (3)

補題 12 $\alpha \geq \frac{23}{26}$ のとき，(2) $\Rightarrow$ (5)

補題 13 $\alpha = \frac{23}{26}$ かつ $low \geq 4$ ならば (2)$\wedge$ (4) をみたす整数 $t(0 \leq t < low)$ が存在する．[5]

補題 14 $\alpha = \frac{23}{26}$ かつ $low \geq 4$ ならば，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*，*4*，*6* より，条件 $(*1) \sim (*3)$ が成立するときの証明が残る．補題 *11* $\sim$ *13* より，条件 (1) $\sim$ (5) が成立する．従って，補題 *7* $\sim$ *10* より，m_2 と m_3 の取りうる値の全ての場合において，この補題が成立する．□

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．但し，レベル 4 のタスクの下界は 4 以上とする．

証明) 補題 *14* 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 *14* を繰り返し適用することで，この定理が成立する．□

[4] 正確には，そのように最大マッチングを $\mathcal{M}_2$ を選ぶことができる．(即ち，M_2 中のあるタスク v が $U_3 \setminus M_2$ 中に親 u を持つならば交換することができる．)

[5] 具体的には，t として (2) 式をみたす最大の整数をとればよい．

4 本手法の時間計算量

e-値 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 を求めるのにそれぞれ $O(|V|)$ 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．

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] 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.

[4] 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 補題 2 の証明

証明) U_1 の要素 u 又は v^* の親 w(但し，$f(w) \leq low$) 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 [3]，u はメインプロセッサに時刻 $\lfloor 1.4e(w)\rfloor + \tau(w) + 1 = f(w) + \lfloor 0.4e(w)\rfloor \leq low + \lfloor 0.4e(u)\rfloor$ 以降にスケジュール可能となる．よって，U_1 中の u のスケジュール時刻は $low + \lfloor 0.4e(u)\rfloor$ の小さい順とすることができる．

(3) レベル 3 のタスクの結果を必要とするタスクは v^* のみであり，w の到達時刻は $\lfloor \frac{5}{3}f(w)\rfloor \leq \lfloor \frac{5}{3}low\rfloor$ であるから，v^* のスケジュール時刻を $\frac{5}{3}low$ 以降とすればよい．

以上より，$u_j, u_{j-1}, \ldots, u_1$ を，時刻 $low + \lfloor 0.4e(u_j)\rfloor$ 以降に連続してスケジュールする．また，$u_{j+1}, u_{j+2}, \ldots, u_l$ を，時刻 $low + \lfloor 0.4e(u_j)\rfloor$ 以前に連続してスケジュールする．即ち，任意の $u_i (1 \leq i \leq l)$ に対して，u_i のスケジュール時刻を $low + \lfloor 0.4e(u_j)\rfloor + j - i$ とする．j の選び方から，$low + \lfloor 0.4e(u_j)\rfloor + j - i \geq low + \lfloor 0.4e(u_i)\rfloor$ が成立し，従って，(2) より，各 u_i はその時刻で開始可能である． □

B 補題 7 の証明

証明) M_1，M_2，N_2，M_1'，M_2' をメインプロセッサに，その他を外部プロセッサにスケジュールする．$M = M_1' \cup M_2'$ と定義し，M に含まれるタスクを e-値 の降順に並べたものを $w_1, w_2, \ldots, w_{|M|}$ とする．従って，$e(w_1) \geq e(w_2) \geq \ldots \geq e(w_{|M|})$ が成立する．$k (1 \leq k \leq |M|)$ は次の式をみたすものとする．$\lfloor 0.4e(w_k)\rfloor + k = \max_{1\leq i\leq |M|} (\lfloor 0.4e(w_i)\rfloor + i)$ と定義する．

- 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．($*3$) と式 (1) より，$m_1 + m_2 \leq \lfloor \alpha low\rfloor + low - l$ ，かつ 補題の前提より，$m_2 + t \leq \lfloor \alpha low\rfloor + low - l$ 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 $\{u \mid 1 \geq level(u) \geq 0, t \geq f(u)\}$ 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 $M_1 \cup M_2 \cup \{u \mid 2 \geq level(u) \geq 0, t \geq f(u)\}$ 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．各 $w_i \in M$ を時刻 $low + \lfloor \alpha low\rfloor - i$ に割り当てる $(1 \leq i \leq |M|)$ ($|M| + |N_2| = l$ より，時刻 $low + \lfloor \alpha low\rfloor - l + |N_2|$ から M の要素を順に割り当てたことになる．) このスケジュールが可能であるためには，補題 *2* の証明と同様に，次の条件をみたせばよい．各 $w_i \in M$ に対して $low + \lfloor \alpha low\rfloor - i \geq low + \lfloor 0.4e(u_i)\rfloor$ (即ち，$\lfloor \alpha low\rfloor \geq \lfloor 0.4e(w_i)\rfloor + i$)．$k$ の選び方より，次の式をみたせば十分である．

$$\lfloor \alpha low\rfloor \geq \lfloor 0.4e(w_k)\rfloor + k \qquad (7.1)$$

  (a) $e(w_k) \geq \frac{5}{3}(low - \lfloor \alpha low\rfloor)$ (即ち，$0.6e(w_k) \geq low - \lfloor \alpha low\rfloor$) のとき，$low \geq e(w_k) + k$ より，$low - 0.6e(w_k) \geq 0.4e(w_k) + k$．従って，$\lfloor \alpha low\rfloor \geq low - 0.6e(w_k) \geq 0.4e(w_k) + k$．ゆえに，(7.1) 式が成立する．

  (b) $e(w_k) < \frac{5}{3}(low - \lfloor \alpha low\rfloor)$ (即ち，$0.4e(w_k) < \frac{2}{3}(low - \lfloor \alpha low\rfloor)$) のとき，$k \leq m_1 + m_2 \leq \frac{5}{3}\lfloor \alpha low\rfloor - \frac{2}{3}low + 1$ ($\because$ (3)) より，$0.4e(w_k) + k < \lfloor \alpha low\rfloor + 1$．ゆえに，(7.1) 式が成立する．

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． □