Vol.2011-SLDM-149 No.5
Vol.2011-EMB-20 No.5
2011/3/18

DVSを用いたマルチプロセッサ・システムのための低消費電力量タスク割当て手法

白石 多一郎[†1] 坂主 圭史[†1]
武内 良典[†1] 今井 正治[†1]

近年，アプリケーションの大規模化に伴う消費電力量の増加が著しい．一方，大規模化するアプリケーションを並列に実行するマルチプロセッサ・システムが台頭している．そのため，マルチプロセッサ・システムにおいても消費電力量の削減が課題である．マルチプロセッサ・システムの消費電力量を削減するには，実行時間制約の下，タスクを割当てるPE，実行する順序，DVSを用いることによるPEの動作電圧と動作周波数を適切に決定する必要がある．本稿では，SA法を基にした低消費電力量タスク割当て手法を提案する．実験の結果，提案手法により非常に良い解が得られることを確認し，また従来手法に対しては，約7%〜約30%消費電力量を削減でき，提案手法の有効性を確認できた．

Low Energy Task Assignment Method for DVS based Multi-Processor System

Taichiro Shiraishi,[†1] Keishi Sakanushi,[†1]
Yoshinori Takeuchi[†1] and Masaharu Imai [†1]

Recently, an increase energy consumption is remarkable because the scale of applications become huge. On the other hand, multi-processor systems that execute huge applications parallel become more popular. Therefore, it is still an important problem to reduce energy consumption in multi-processor systems. To reduce energy consumption, it is necessary to optimize assignments of tasks to PEs, the order of executing tasks and the operating voltage and frequency for each task by DVS under execution time constraint. This paper proposes a low energy task assignment method based on simulated annealing method. Experimental results show that the proposed method can obtain optimal or near optimal solutions. The proposed method can obtain about from 7% to 30% less energy solution than that of the conventional method.

1. はじめに

近年，アプリケーションの大規模化により，システムの消費電力量が増加している．一方，大規模化するアプリケーションに対して，複数のPEを用いて並列処理を行うことでアプリケーションを高速に実行し，マルチプロセッサ・システムが台頭している．そのため，マルチプロセッサ・システムにおいても消費電力量の削減が課題となっている．

消費電力量の増加の主な要因として，処理するアプリケーションの大規模化による，PEの処理量の増加に伴う消費電力量の増加，処理データ量の増大に伴うPE間の相互接続上で消費される消費電力量の増加が挙げられる．PEの消費電力量の削減には，PEの動作電圧と動作周波数を動的に変更するDynamic Voltage Scaling (DVS)[1)]の適用，相互接続上の消費電力量の削減には相互接続上で発生するデータ通信処理を考慮したタスク割当てが考えられる．実行時間制約を超えない範囲で，システムの消費電力量を削減するためには，タスクを割当てるPE，実行する順序，実行するときのPEの動作電圧と動作周波数を適切に決定する必要がある．文献[4)]では，各PEに割当てる処理および処理の実行順序を決定し，その後にPEの動作電圧と動作周波数を決定することで消費電力量の削減を行っている．しかし，PEに割当てる処理，処理の実行順序，PEの動作電圧と動作周波数を段階的に決定しており，行うことは必ずしもシステム全体の消費電力量を最小化しているとは限らない．そこで本稿では，各PEに割当てる処理，処理の実行順序，各PEの動作電圧と動作周波数を同時に考慮した低消費電力量タスク割当て手法を提案する．以下，本稿の節構成は次の通りである．第2節では対象とするシステムモデルについて説明する．第3節ではタスク割当て最適化問題について説明する．第4節で低消費電力タスク割当て手法について説明する．第5節で評価実験を行い，その結果と考察を述べる．第6節ではまとめと今後の課題と展望について述べる．

2. DVSを用いたマルチプロセッサ・システムモデル

本稿では，複数の動作電圧と動作周波数が動的に変更できるPEから構成されるマルチプロセッサ・システムを対象とする．また，対象とするシステムにおいて，PE内部の演算処理とPE間のデータ通信処理は同時に実行が可能であり，複数のデータ通信処理も同時に実

†1 大阪大学 大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University

Vol.2011-SLDM-149 No.5
Vol.2011-EMB-20 No.5
2011/3/18

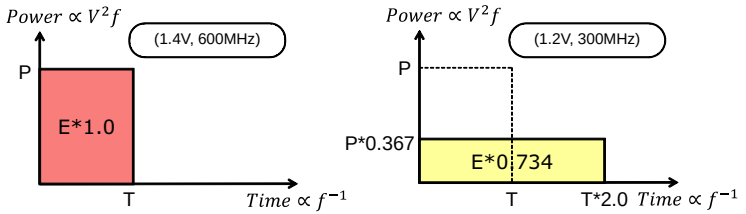

図 1 動作電圧と動作周波数の変更に伴う消費電力量と実行時間の変化

行できると仮定する．以下，DVS の特徴と，相互接続上におけるデータ通信処理について説明する．

2.1 Dynamic Voltage Scaling (DVS)

Dynamic Voltage Scaling (DVS)[1] とは PE の動作電圧と動作周波数の値をアプリケーション実行中に動的に変更する技術である．PE の消費電力量が動作電圧の二乗に比例し，実行時間が動作周波数の逆数に比例するため，動作電圧と動作周波数を変更することにより，アプリケーション実行時の PE の消費電力量を変化させることが可能である．図 1 に PE の動作電圧と動作周波数を変更したときの消費電力および実行時間の変化の例を示す．縦軸が消費電力，横軸が実行時間を示し，グラフ中の矩形の面積が消費電力量を表す．動作電圧と動作周波数がそれぞれ 1.4V，600MHz のときの消費電力量を E，実行時間を T とすると，動作電圧と動作周波数をそれぞれ 1.2V，300MHz に変更すると，のときの消費電力量は動作電圧の二乗に比例して $E \times 0.734$，実行時間は動作周波数の逆数に比例して $T \times 2.0$ となる．

一般的に，PE の動作電圧と動作周波数の値は，PE 毎に設定された複数の動作電圧と動作周波数の値の組から選択される．

2.2 相互接続上でのデータ通信処理

アプリケーションをタスクグラフとよばれる，タスクを点，タスク間のデータ依存関係を辺とした有向無閉路グラフで表すとき，相互接続上でのデータ通信処理は，データ依存関係にある 2 つのタスクを異なる PE に割当てることにより発生する．

相互接続上で発生するデータ通信処理はタスクの割当て方によって変化するため，システムの消費電力量および実行時間に影響する．図 2 右上，右下の図は，図 2 左のタスクグラフを PE に割当てたときのタスクの実行および相互接続上で発生するデータ通信処理を示している．図 2 右上では t_2 が PE2 に割当てられているため，t_1，t_2 間および t_2，t_5 間のデータ通信処理が相互接続上で発生する．図 2 右下では t_3，t_4 が PE2 に割当てられているため，t_1，t_3 間および t_4，t_5 間のデータ通信処理が相互接続上で発生する．図 2 右上，右下で発生するデータ通信処理が異なるため，それに伴って消費電力量と実行時間が変化する．図 2 右下は図 2 右上に比べて実行時間が長く消費電力量が小さくなっている．

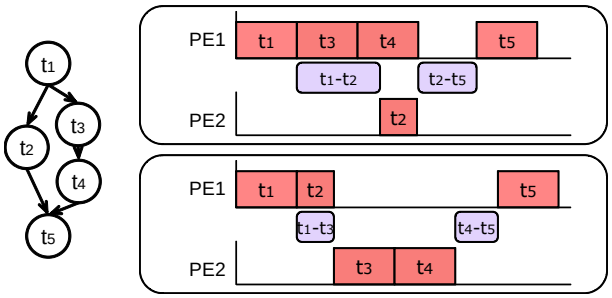

図 2 タスク割当てにより発生するデータ通信処理の様子

3. タスク割当て最適化問題

本稿が対象とするタスク割当て最適化問題における入力，制約条件，目的関数および出力を以下に示す．

- 入力
  - PE の個数，各 PE について，DVS で使用可能な動作電圧と動作周波数の値の組
  - タスクグラフ
  - 各 PE において，各タスクの実行に必要な実行時間，消費電力量
  - タスク間のデータ通信を相互接続上で処理する際に必要な通信時間，消費電力量
- 制約条件
  - 実行時間制約
- 目的関数
  - 消費電力量
- 出力
  - スケジュール

本稿におけるスケジューリングとは，タスクを割当てる PE，タスクの実行順序，タスク実行時の PE の動作電圧と動作周波数を決定する操作である．

Vol.2011-SLDM-149 No.5
Vol.2011-EMB-20 No.5
2011/3/18

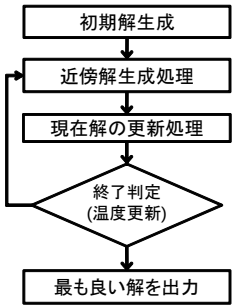

図 3 低消費電力量タスク割当て手法の流れ

3.1 関 連 研 究

本稿が対象とするタスク割当て最適化問題に対して，短時間に，最適解または最適解に近い解を求める手法が考案されている．文献[3] では発見的手法によって DVS を適用することで，消費電力量を最小化する解を求めているが，データ通信によるシステムの消費電力量および実行時間への影響を考えていない．また，文献[4] では，まず，データ通信量を削減するためにタスクを割当てる PE とタスクの実行順序を ETF スケジューリング[5] によって決定する．その後，各 PE で利用可能な余裕時間を計算し，最も消費電力量を削減できるタスクと PE に DVS を適用することを繰り返して最適解を求めている．しかし，PE における余裕時間はタスク割当ておよびタスクの実行順序によって変化するため，文献[4] はタスク割当てとタスクの実行順序，タスク実行時の動作電圧と動作周波数を別々に行っており，必ずしもシステム全体の消費電力量を最小化していない．

4. 低消費電力量タスク割当て手法

タスク割当て最適化問題に対して，本稿は Simulated Annealing 法 (SA 法) を利用した低消費電力量タスク割当て手法を提案する．提案手法の流れを図 3 に示す．また，提案手法の制約として，図 3 の初期解生成において与えられる初期解は実行時間制約を満たす必要がある．以下，近傍解生成処理，現在解の更新処理について説明する．

4.1 近傍解生成処理

本稿では，現在解から近傍解を生成する変更操作として 5 種類の操作を提案する．近傍解生成処理では，5 種類の操作からランダムに 1 つ選択して近傍解を生成する．以下，提案する 5 種類の操作について説明し，各操作によって生成される近傍解を，図 4 に示す入力タスクグラフ，DVS の情報，PE 数および現在解を用いて説明する．入力タスクグラフは 5 つのタスクで構成され，t_1 と t_2，t_2 と t_5，t_1 と t_3，t_3 と t_4，t_4 と t_5 にそれぞれデータ依存関係が存在する．また，PE で使用可能な動作電圧と動作周波数の値の対は (1.0V, 150MHz), (1.2V, 300MHz)，(1.4V, 600MHz) の 3 対とする．現在解は，PE1 に順に t_1，t_2，t_4，t_5 が割り当てられ，PE2 に t_3 が割り当てられており，各タスクを実行するときの PE の動作電圧と動作周波数の値が，t_1，t_2，t_3，t_4，t_5 の順に，(1.4V, 600MHz)，(1.0V, 150MHz), (1.4V, 600MHz)，(1.2V, 300MHz)，(1.4V, 600MHz) となっている．

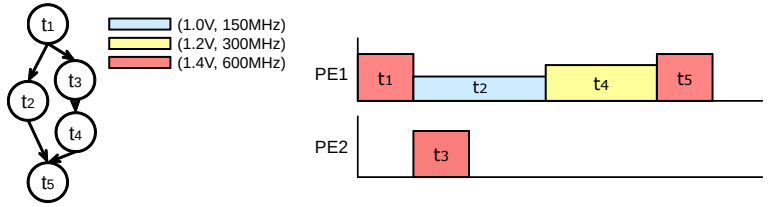

図 4 入力および現在解

4.1.1 操作 1:動作電圧と動作周波数の減少

操作 1 によって近傍解を生成する手順を説明する．

手順 1

タスク実行時の PE の動作電圧と動作周波数が最小値でないタスクの中からランダムに 1 つタスクを選択する．

手順 2

選択したタスクを実行する PE の動作電圧と動作周波数の値を減少させる．

図 4 に示す解に対して操作 1 を行って生成される近傍解を図 5 に示す．手順 1 で t_1，t_3，t_4，t_5 からランダムにタスクが選択される．t_3 が選択された場合，手順 2 で t_3 を実行する際の動作電圧と動作周波数が (1.4V, 600MHz) から (1.2V, 300MHz) に減少する．

4.1.2 操作 2:動作電圧と動作周波数の増加

操作 2 によって近傍解を生成する手順を説明する．

手順 1

タスク実行時の PE の動作電圧と動作周波数が最大値でないタスクの中からランダムに 1 つタスクを選択する．

手順 2

Vol.2011-SLDM-149 No.5
Vol.2011-EMB-20 No.5
2011/3/18

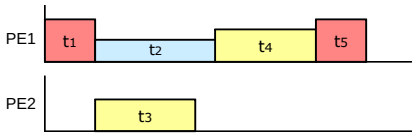

図 5 操作 1 で生成される近傍解

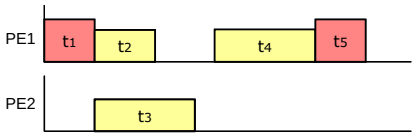

図 6 操作 2 で生成される近傍解

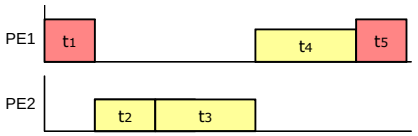

図 7 操作 3 で生成される近傍解

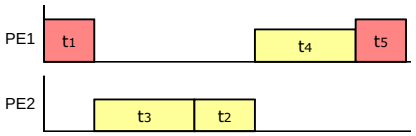

図 8 操作 4 で生成される近傍解

選択したタスクを実行する PE の動作電圧と動作周波数の値を増加させる．

図 5 に示す解に対して操作 2 を行って生成される近傍解を図 6 に示す．手順 1 で t_2，t_3，t_4 からランダムにタスクが選択される．t_2 が選択された場合，手順 2 で t_2 を実行する際の動作電圧と動作周波数が (1.0V, 150MHz) から (1.2V, 300MHz) に増加する．

4.1.3 操作 3:割当て先 PE 変更

操作 3 によって近傍解を生成する手順を説明する．

手順 1

現在解において，ランダムにタスクを 1 つ選択し，選択したタスクが割当てられていない PE をランダムに選択する．

手順 2

選択したタスクを選択した PE の中で最も早く実行可能な位置に挿入する．

手順 3

選択した PE での選択したタスクを実行するときの動作電圧と動作周波数の値の対について，変更前後の PE が同種であれば値は保持され，異なる場合は変更先の PE で使用可能な対からランダムに選択する．

図 6 に示す解に操作 3 を行って生成される近傍解を図 7 に示す．手順 1 で t_2 が選択された場合，割当て先の PE は t_2 が割当てられている PE1 以外の PE2 が選択される．手順 2 で t_2 は PE2 内で最も早く実行できる位置に挿入されるため，t_2 は t_3 の前に一意に割当てられる．手順 3 で，PE1 と PE2 は同種であるため，t_2 を実行するときの動作電圧と動作周波数は操作 3 の適用の前後で変化しない．

4.1.4 操作 4:実行順序入れ替え

操作 4 によって近傍解を生成する手順を説明する．

手順

同一 PE 内の連続に実行されていて並列実行可能な関係にある 2 つタスクを選択し，それらの実行順序を入れ替える．

ここで，2 つのタスクが並列実行可能な関係にあるとは，タスクグラフにおいて順方向のみまたは逆方向のみに有向辺を辿ったときに，一方のタスクからもう一方のタスクへと至る経路が存在しないことを意味する．

図 7 に示す解に対して操作 4 を行って生成される近傍解を図 8 に示す．連続に実行されていて並列実行可能な関係にあるタスクは t_2 と t_3 のみであるため，t_2 と t_3 の実行順序を入れ替える．

4.1.5 操作 5:割当て箇所の交換

操作 5 によって近傍解を生成する手順を説明する．

手順 1

現在解において互いに異なる PE に割当てられている並列実行可能な 2 つのタスクを選択し，選択した 2 つのタスクの割当て箇所を入れ替える．

手順 2

交換の前後で 2 つのタスクを実行するときの動作電圧と動作周波数の値の対は，交換前後で PE が同種であれば値は保持され，異なる場合は交換先の PE で使用可能な対からランダムに選択する．

図 8 に示す解に対して操作 5 を行って生成される近傍解を図 9 に示す．手順 1 において，互いに異なる PE に割当てられている並列実行可能な関係にあるタスクは t_2 と t_4 であるため，図 9 では t_2 と t_4 の割当て箇所が交換されている．交換の前後で PE の種類は変わらないため，t_2，t_4 を実行する PE の動作電圧と動作周波数の値は保持される．

4.2 現在解の更新処理

解 S の評価関数 $cost(S)$ を式 (1) で定義する．

$$cost(S) = \sqrt{\left(\frac{E(S) - E_{min}}{E_{min}}\right)^2 + \left(\frac{L(S) - DL}{DL}\right)^2} \quad (1)$$

Vol.2011-SLDM-149 No.5
Vol.2011-EMB-20 No.5
2011/3/18

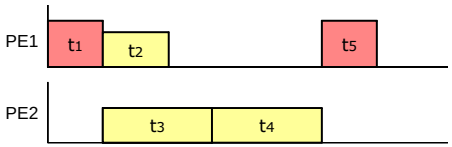

図 9 操作 5 によって生成される近傍解

式 1 において，$L(S)$，$E(S)$ はそれぞれ解 S における実行時間，消費電力量を表しており，DL は実行時間制約の値，E_{min} は SA による探索中で発見された最小の消費電力量を表している．$cost(S)$ は 2 点 $(L(S), E(S))$，(DL, E_{min}) の正規化距離を表し，$cost(S)$ を 0 に近づけていくことにより，解 S の実行時間を実行時間制約付近に収束させつつ消費電力量を低下するように探索する．

5. 評 価 実 験

本節では提案手法の有効性を示すため，提案手法を計算機プログラム上に実装し実験を行った

Simulated Annealing 法におけるパラメータは，初期温度 100 度，終了温度 0.0001 度，各温度における探索回数 1000 回，温度更新時の倍率 0.99 とした．また，1 つのタスクグラフに対して提案手法を 10 回適用し，得られた 10 個の解の消費電力量の平均値，最小値，最大値を評価対象とした．初期解は ETF スケジューリング[5] によって与え，実行時間制約は初期解の実行時間とする．入力となるタスクグラフは TGFF[7] を用いて自動生成した．

また，PE において DVS で使用可能な動作電圧と動作周波数の値の対は (1.4V, 600MHz)，(1.2V, 300MHz)，(1.0V, 150MHz) の 3 対とする．

以下，評価実験の詳細と実験結果について述べる．なお，本実験の実験環境は Intel Core2Quad 3.00GHz, メモリ 4GB である．

5.1 最適解との比較

まず，全探索で得られる最適解における消費電力量との比較評価実験を行う．本実験では，PE の数は 3 つとし，タスクグラフはタスクの実行時間とタスク間のデータ転送時間がほぼ同じであり，タスク数が 6〜16 個のグラフを 100 個用意した．提案手法で得られる 10 個の解の消費電力量のの平均値と全探索で得られる最適解の消費電力量との差を表 1 に示す．

実験の結果，99 個中 89 個の入力に対して平均値と最適解の消費電力量との差が 0 となり，10 個の解がすべて全探索で得られた最適解と一致した．また，平均値と最適解の消費電力量との差は 0.8%未満であり，提案手法により非常に良い解が得られることが確認できた．なお，提案手法を 10 回適用して得られた 10 個の解の消費電力量の平均値と最適解の消費電力量に差があったグラフ A〜J について，10 個の解の消費電力量の最小値，最大値と最適解の消費電力量の差を表 2 に示す．最小値と最適解の消費電力量との差は 0 であり，最小値については最適解の消費電力量と一致していることを確認した．また，最大値と最適解の消費電力量との差は 2.3%未満であり，提案手法で得られる解のバラつきが小さいことを確認した．

また，タスク数が 15 個を超える入力については解の総数が 100 兆程度存在するが，提案手法では約 120 万回程度の探索で最適解を求めることができており，提案手法が最適解に対して非常に良い収束性を持つことを確認できた．

表 1 消費電力量の差の区分

消費電力量の差	タスクグラフの数
0% (最適解と一致)	89 個
0.01%未満	3 個
0.01%以上 0.1%未満	3 個
0.1%以上 0.8%未満	4 個

表 2 各グラフにおける最小値，平均値，最大値での消費電力量の差

入力グラフの記号	A	B	C	D	E
最小値における差	0%	0%	0%	0%	0%
平均値における差	0.00216%	0.00577%	0.03002%	0.00822%	0.04864%
最大値における差	0.01078%	0.05772%	0.07866%	0.08225%	0.46487%
入力グラフの番号	F	G	H	I	J
最小値における差	0%	0%	0%	0%	0%
平均値における差	0.09326%	0.34213%	0.54576%	0.37287%	0.77575%
最大値における差	0.46629%	1.25192%	1.37989%	1.85409%	2.21642%

5.2 従来手法との比較評価

次に，従来手法[4] と提案手法を比較した．本実験では，提案手法の評価対象である平均値について，従来手法で得られる解の消費電力量からの削減率を調査する．

本実験では，タスク数について約 50 個または約 100 個，PE 数について 4 個または 8 個，タスクの実行時間とタスク間のデータ転送時間の比がおよそ 10:1 または 1:1 を考え，計 8

Vol.2011-SLDM-149 No.5
Vol.2011-EMB-20 No.5
2011/3/18

表 3 従来手法で得られた解の消費電力量からの削減率の平均値

(タスクの実行時間):(データ通信時間) = 10:1	PE 数:4	PE 数:8
タスク数:50	6.40%	6.65%
タスク数:100	6.30%	6.93%
(タスクの実行時間):(データ通信時間) = 1:1	PE 数:4	PE 数:8
タスク数:50	30.13%	22.71%
タスク数:100	33.67%	30.22%

表 4 探索時間の推移

タスク数	7	10	12	46	53	57	95	98	108
探索時間 (秒)	0.795	0.936	1.045	3.291	4.057	4.415	8.518	9.735	10.592

通りの条件のそれぞれに対してタスクグラフ 180 個を用いた．入力した 180 個のタスクグラフに対して得られた消費電力量の削減率の平均値を表 3 に示す．

実験の結果，提案手法で得られた解の消費電力量は，表 3 の上段では約 7%程度，表 3 の下段では約 30%前後の削減を確認でき，提案手法で得られる解の消費電力量が従来手法で得られた解の消費電力量よりも小さいことが確認できた．

5.3 探索時間の評価

約 120 万回探索を行った時の探索時間を表 4 に示す．表 4 から，タスク数が約 100 個になっても探索時間は 10 秒程度であり，実用的な時間で探索が可能であることを確認した．

6. ま と め

本論文では，DVS を用いたマルチプロセッサ・システムのための低消費電力量タスク割当て手法を提案した．実験の結果，タスク数 6〜15 個のタスクグラフに対しては，全探索で得られた最適解と一致または最適解に非常に非常に近い解を得られることを確認し，タスク数が 50 や 100 の規模の大きいグラフの入力に対しては，既存手法と比較して非常に良い解を得られることを確認した．また，探索に要する時間について，約 120 万回の探索で，タスク数が約 100 個となっても 10 秒程度で探索が終了し，解を得るための探索時間が非常に短いことを確認し，静的なスケジューリングであれば，探索時間は十分に許容範囲内であることが確認できた．

今後の課題としては，データ通信処理の競合を考慮することによる対象とするシステムモデルの拡大が考えられる．

謝辞 本研究の一部は科学研究費補助金基盤研究 (B)(課題番号 20300017) の助成によるものである．

参 考 文 献

1) 水野 弘之, ”DVS/DVFS 技術による低電力化の今後” 電子情報通信学会技報 SDM2007-148 ICD2007-76 pp.41-46.
2) SupueH RISC engine ファミリ http://documentation.renesas.com/jpn/products/-mpumcu/doc/superh/r01cl0001jj0100_superh.pdf
3) G.Varatkar and R.Marculescu: ”Communication-Aware Task Scheduling and Voltage Selection for Total Systems Energy Minimization”: Proc. of 2003 Int’l. Conf.on Computer-aided design (ICCAD), pp.510-517(2003).
4) Yuichiro MORI, Koichi ASAKURA, Toyohide WATANABE,”A Task Selection Based Power-aware Scheduling Algorithm for Applying DVS” International Conference on Parallel and Distributed Computing, Application and Technologies, pp.518-523
5) Ahmad, Y. K. Kwok, and M. Y. Wu, ”Analysisi, evaluation, and comparison of algorithms for scheduling task graphs on parallel procesors,” in Proceedings of the 1996 International Symposium on Parallel Architectures, Algorithm and Networks. Washinton, DC, USA: IEEE Computer Society, 1996, pp.207
6) J. Hromkovic 著，和田 幸一，増澤 利光，元木 光雄 訳，”計算困難問題に対するアルゴリズム理論,” シュプリンガー・フェアラーク東京，pp.469-477 2002.
7) TGFF(Task Graph For Free) http://ziyang.eecs.umich.edu/dickrp/tgff/