Vol.2016-ARC-219 No.33
Vol.2016-SLDM-175 No.33
Vol.2016-EMB-40 No.33
2016/3/25

タスクの実行時間とPEの消費電力のばらつきを考慮したスケジューリング手法

野村 孔命[1,a)]　高島 康裕[1,b)]　中村 祐一[2,c)]

概要：本稿では，タスクの実行時間と Processing Element(PE) の消費電力がばらつくシステムにおいてのスケジューリング手法を提案する．近年，実行時間と消費電力のばらつきを考慮したタスクの割り当てとスケジューリング (TAS) は，マルチプロセッサシステムが正しく動作するために必要である．従来手法では、TAS を行った後、実行時間と電力の歩留まりを評価していた．しかし、この方法では目標となる歩留まりを得るまでに，何度も TAS を実行しなければならず，設計時間が膨大になってしまう．そこで，本稿では，両ばらつきが正規分布に従うという仮定に基づいて，まず，電力制約をある確率で満たす PE の組み合わせを計算し，その組み合わせだけを利用した上で実行時間最小となる Power and Execution Variation-Aware Scheduling(PEVaS) を提案する．実験により，従来手法と比較し，性能が向上していることを確認した．

キーワード：リストスケジューリング，MPSoC，TAS，電力制限，正規分布，PEVaS

Task execution time and PE power consumption variation-aware scheduling method

Komei NOMURA[1,a)]　Yasuhiro TAKASHIMA[1,b)]　Yuichi NAKAMURA[2,c)]

Abstract: We propose a scheduling method with considering the task execution time variation and the Processing Element(PE) power consumption variation. Task allocation and scheduling (TAS) with the execution time and power consumption variation is necessary to ensure the correction of the Multiprocessor System-on-Chip(MPSoC). For this problem, the previous method evaluates the yield of the execution time and power consumption after TAS. This method, however, may need large design time to satisfy the yield constraint due to too many iterations of TAS. We propose a novel method, called Power and Execution Variation-Aware Scheduling(PEVaS), to achieve the minimum execution time with the combination of PE's that satisfies the power constraint with some probability. We confirm the efficiency of the proposed method compared with that of the previous method, empirically.

Keywords: List scheduling, MPSoC, TAS, power constraint, normal distribution, PEVaS

1. 序論

マルチプロセッサシステムにおいてタスクの割り当てとスケジューリング (TAS) は重要である．特に近年では，実行時間と消費電力のばらつきを考慮することが，正しくシステムが動作するために必要である．この問題に対し，従来手法では、TAS を行った後、実行時間と消費電力の歩留まりを評価し，もし制約を満たしていなかった場合は，入力の電力制約パラメータや MPSoC の構成を変更し，TAS の再計算後，再度評価を行なうという繰返しにより実現する方法が取られている [3]．この手法は実行時間と消費電力の分布に依存せずに歩留まりの評価が可能な為，汎用性が高い，しかし，スケジューリングを行う際に消費電力を

[1] 北九州市立大学 国際環境工学部　〒 808–0135 福岡県北九州市若松区ひびきの 1–1
[2] NEC グリーンプラットフォーム研究所 〒 211–8666 神奈川県川崎市中原区下沼部 1753
[a)] w5mcb006@eng.kitakyu-u.ac.jp
[b)] takasima@kitakyu-u.ac.jp
[c)] yuichi@az.jp.nec.com

Vol.2016-ARC-219 No.33
Vol.2016-SLDM-175 No.33
Vol.2016-EMB-40 No.33
2016/3/25

考慮していないため，生成される TAS は電力制約を違反する確率が高くなり，結果として，目標となる歩留まりを満たすことが難しくなっている．そのため，何度も再スケジューリングと評価を行うので設計時間が膨大となってしまう．

そこで，本稿では，実行時間と消費電力のばらつきが正規分布に従うと仮定し，まず，電力制約をある確率で満たす PE の組み合わせを求め，その組み合わせだけを利用して，実行時間最小となるようなスケジューリング手法を提案する．また，実行時間の最小化には，従来の静的スケジューリングに比べて実行時間性能が向上する動的スケジューリング [4] を利用する．ここで，歩留まりは電力制約と実行時間制約を考慮するものとし，PE はタスクとタイプが一致した場合割り当て可能と仮定した．そして，提案手法の有用性をを確認するため，従来手法との比較実験を行った．その結果，提案手法によって，デザイン制約 (電力制約，実行時間制約) を満たすことができ，かつ，TAS の効率が改善されていることを確認した．

本稿は以降，以下のように構成される．第 2 章において，タスクを統計的に評価する残実行時間の計算方法と残実行時間を用いた動的統計的リストスケジューリングについて説明する．第 3 章で電力テーブル，電力制限スケジューリングについて説明する．第 4 章で実験結果と結果に対する考察を述べる．最後に，第 5 章で結論を述べる．

2. 準備

2.1 正規分布に対する基本演算

本稿では，タスクの実行時間が正規分布に従うとしたときのスケジューリングについて検討する．そこで，その準備として，正規分布に対する基本的な演算を紹介する [1]．本節の議論では，2 数 x_1，x_2 がそれぞれ正規分布に従う，すなわち，$x_1 \sim N(\mu_1, \sigma_1^2)$，$x_2 \sim N(\mu_2, \sigma_2^2)$ とする．ここで，$N(\mu, \sigma^2)$ は，平均 μ，分散 σ^2 の正規分布を表わす．

まず，2 数の和 x_1+x_2 の分布は，$x_1+x_2 \sim N(\mu_1+\mu_2, \sigma_1^2+\sigma_2^2)$ である．

また，2 数の最大値 $x = \max\{x_1, x_2\}$ については，Clark の手法 [2] を用いる．今，$x_1 \sim N(\mu_1, \sigma_1^2)$，$x_2 \sim N(\mu_2, \sigma_2^2)$，かつ，これらの共分散を c_{12} としたとき，$x = \max\{x_1, x_2\} \sim N(\mu, \sigma^2)$ は，式 (1) のように近似される．

$$\begin{aligned}\mu =& \mu_1\Phi(\alpha) + \mu_2\Phi(-\alpha) + \beta\phi(\alpha)\\ \sigma^2 =& \left(\mu_1^2+\sigma_1^2\right)\Phi(\alpha) + \left(\mu_2^2+\sigma_2^2\right)\Phi(-\alpha) +\\ & (\mu_1+\mu_2)\beta\phi(\alpha) - \mu^2\\ & \text{ただし}, \beta = \sqrt{\sigma_1^2+\sigma_2^2-2\sigma_1\sigma_2 c_{12}},\ \alpha = \frac{\mu_1-\mu_2}{\beta}\end{aligned} \tag{1}$$

また，x_1，x_2 との共分散 c_{1y}，c_{2y} を持つ正規分布 y と，$x = \max\{x_1, x_2\}$ との共分散 c_{xy} は，式 (2) で表される．

$$c_{xy} = \frac{\sigma_1 c_{1y}\Phi(\alpha) + \sigma_2 c_{2y}\Phi(-\alpha)}{\sigma^2} \tag{2}$$

ここで注意すべきは，以上の計算により得られた和及び最大値の分布もまた正規分布となっていることである．その為，以上の計算は繰り返し適用可能である．

2.2 残実行時間 [4]

本稿では，あるタスクの実行を開始してから全ての後続タスクの実行が終わるのに必要な時間を残実行時間として定義する．残実行時間の計算の処理は，ALAP スケジューリングの流れに沿ったものとなっている．違いは，通常の ALAP は，各タスクの実行時間がスカラー値となっているのに対し，ここでは，分布として表現され，統計的処理を用いて，計算されるところである．入力として，タスク間の依存関係を表わすデータフローグラフ (DFG) と各タスクの実行時間の分布として平均と分散が定められた正規分布が与えられる．

(1) DFG において，トポロジカルオーダの逆順に節点 x を取り出す．
(2) x の残実行時間を直接後続節点の残実行時間と x の実行時間から計算する．
  (a) 直接後続節点の残実行時間の最大値を Clark の手法を用いて計算する．もし，後続節点が無い場合は，この処理はスキップし，平均 0，分散 0 として次に進む．
  (b) x の残実行時間を直接後続節点の残実行時間の最大値に x の実行時間を足すことで得る．

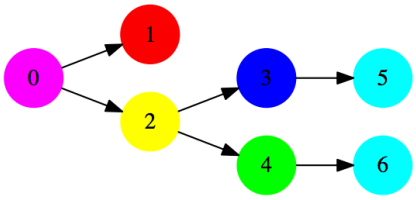

図 1 DFG の例

図 1 に示した DFG にこのアルゴリズムを適用する．図 1 の場合，節点番号がそのままトポロジカルオーダ順になっている．そこで，節点 6，節点 5 が順に取り出され，それぞれの残実行時間が計算される．ここで，節点 6，節点 5 は後続節点を持っていないので，それぞれの残実行時間は実行時間そのものである．次に，節点 4 が取り出される．節点 4 の直接後続節点は，節点 6 であるので，節点 6 の残実行時間に節点 4 の実行時間を加えることにより節点 4 の残実行時間が計算できる．同様に，節点 3 に対しても，直接後続節点の節点 5 の残実行時間に節点 3 の実行時間を加

Vol.2016-ARC-219 No.33
Vol.2016-SLDM-175 No.33
Vol.2016-EMB-40 No.33
2016/3/25

えることにより残実行時間を計算する．次に，節点 2 に対し処理を行なう．節点 2 においては，直接後続節点が 3 と 4 であるため，これらの残実行時間の最大値を Clark の手法を利用して求める．そして，その結果と節点 2 の実行時間との和により，節点 2 の残実行時間を求める．以下，同様にして，全節点の残実行時間を求めることができる．

2.3 動的統計的リストスケジューリング

本節では，動的統計的リストスケジューリングの手順を説明する．まず，2.2 節で述べた手法により，各タスクの残実行時間を求める．その結果，各タスクの残実行時間の分布が正規分布 $N\left(\mu_i^r, \sigma_i^{r2}\right)$ として求まる．次に，各タスクの優先度を $\mu_i^r + 3\sigma_i^r$ として，リストスケジューリングを行なう．この動的統計的リストスケジューリングでは，先行タスクの終了後にタスクを割り当てるので，各 PE の利用が効率的に行なわれ，結果として実行時間の削減が期待できる [4].

3. 提案手法

本節では，タスクの実行時間と PE の消費電力が正規分布に従うときのスケジューリング手法を提案する．まず，本稿で考える問題を定義 1 で定義する．

定義 1 タスクの実行時間と PE の消費電力が正規分布に従う場合に対するスケジューリング問題

入力 データフローグラフ $G = (V, E)$，各タスクタイプ，各タスク i の実行時間 $\left(\sim N\left(\mu_i^e, \sigma_i^{e2}\right)\right)$，PE 数，各 PE タイプ，各 PE タイプ t の電力消費 $\left(\sim N\left(\mu_t^e, \sigma_t^{e2}\right)\right)$，電力制約，実行時間制約，目標歩留まり

出力 TAS

制約 先行制約，利用 PE に重なりがない，割り当て制約，目標歩留まりの達成

本稿では，同じ PE タイプは同じ電力分布とし，歩留まりは電力制約と実行時間制約で評価する．提案手法は，以下の流れで TAS を完了する．1）電力テーブルを作成，2）電力を制限しながらスケジューリング．以下，各手法について説明をする．

3.1 電力テーブル作成

電力テーブルとは，電力制約を統計的に満たすことのできる PE の組み合わせを列挙したテーブルと定義する．電力テーブル作成処理を以下に示す．

(1) PE タイプ毎に電力制約を満たす最大の PE 数を計算する．ここで，電力分布は，正規分布の和の性質を用いて統計的に計算し，$\mu + a\sigma$ で評価する．ただし，a は範囲を決定する係数である．

(2) 各 PE タイプの最大数以下の PE の組み合わせを列挙し，電力制約を満たすものをテーブルに追加する．

例として，PE_A と PE_B の 2 種類が利用可能であり，電力制約が 45 である場合を考える．なお，各 PE の電力分布を表 1 に示す．

表 1 PE の電力ばらつき

Type	Power Variation
A	N(10, 1^2)
B	N(15, 2.25^2)

まず，PE_A について最大数を求める．以下の議論では，評価の係数として $a = 3$ を利用する．すなわち，評価式は $\mu + 3\sigma$ である．まず，PE_A が 1 個の場合は，電力分布は $N(10, 1)$ となるため，$\mu + 3\sigma = 10 + 3 \cdot 1 = 13$ より，電力制限を満たす．同様に，PE_A が 2 個の場合は，電力分布は $N(20, 2)$ となり，$\mu + 3\sigma = 24.24$，PE_A が 3 個の場合は，電力分布は $N(30, 3)$ となり，$\mu + 3\sigma = 35.19$ であり，いずれも電力制限を満たす．一方，PE_A が 4 個のときは，$\mu + 3\sigma = 46$ であり，電力制限 45 を越えてしまうので，利用不可能である．よって，PE_A は最大 3 個利用可能である．PE_B についても求めると，PE_B は最大 2 個利用可能である．

次に，各 PE タイプの最大数以下の PE の組み合わせを列挙し，電力制約を満たすものをテーブルに追加する．図 2 は，横軸に PE_A の数，縦軸に PE_B の数を取り，利用可能な PE の組み合わせの最大値を実線で示したものである．ここで，この実線を求めるに対し，明らかに点線で示された領域内だけを探索すれば良いことがわかる．具体的には，PE_B を 1 個利用したときを考えると，$(PE_A, PE_B) = (2, 1)$ では，評価が 42.97 となり，電力制約を満たすので，テーブルに追加するが，$(PE_A, PE_B) = (2, 2)$ では，評価が 53.51 となり，電力制約を違反するので，テーブルに追加しない．これらの処理をすることで図 2 の各軸と太線で囲まれたおよびその境界の組み合わせが電力テーブルに追加され，電力テーブルが完成する．完成した電力テーブルを表 2 に示す．

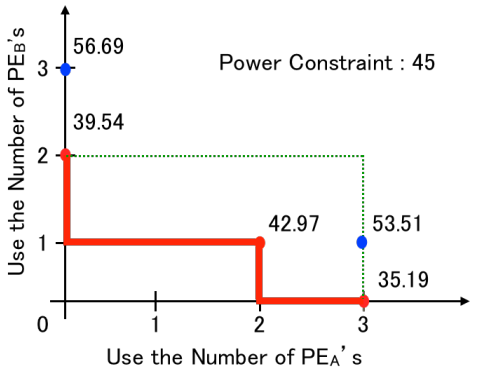

図 2 PE の組み合わせ

Vol.2016-ARC-219 No.33
Vol.2016-SLDM-175 No.33
Vol.2016-EMB-40 No.33
2016/3/25

表 2　電力テーブル

PE_A	PE_B	消費電力	PE_A	PE_B	消費電力
0	0	0	1	1	32.38
0	1	21.75	2	0	24.24
0	2	39.54	2	1	42.97
1	0	13	3	0	35.19

3.2　電力制限スケジューリング

電力制限スケジューリングとは，3.1 節で述べた電力制約で許されている PE の組み合わせのみを用いてタスクの割り当てを行うスケジューリング手法である．電力テーブルには，統計的に電力制約を満たす PE の使い方の組み合わせが列挙されている．つまり，電力テーブルに含まれる PE の使い方だけを利用すれば，スケジューリングの結果，電力制約違反での歩留まり劣化を抑制することができる．ここで，電力制約を考慮した PE の利用では，実行前のスケジューリングだけではなく，実行時に制御する必要がある．すなわち，利用可能な PE とは，単にタスクが実行されていないだけでなく，電力利用状況まで考慮しなくてはならない．

3.3　動的統計的リストスケジューリング&電力制限スケジューリング

2.3 節で述べたように動的統計的リストスケジューリングは実行時に優先度の高いタスクを随時割り当てる手法である．また，3.2 節で提案した電力制限スケジューリングは PE の利用状況を確認して，PE 利用を制御する手法である．この 2 つの手法は PE の利用状況を確認するという点において共通であり，組み合わせることが可能である．そこで，この 2 つの手法を組み合わせた手法を Power and Execution Variation-Aware Scheduling(PEVaS) と呼ぶ．PEVaS の詳細を以下に記述する．

(1)　各タスクの残実行時間の計算，及び，電力テーブルを作成する．
(2)　タスクの集合 *Tasks* から先行タスクが終了しているタスクを取り出し *Ready* に追加する．
(3)　以下 *Ready* が空になるまたは利用できる PE がなくなるまで処理を繰り返す．
　(a)　*Ready* から優先度の最も高いタスクを一つ取り出し，タスクタイプと利用可能な PE のタイプが一致しているかどうかを確認する．一致しなかった場合は *Wait* に追加する．
　(b)　タスクを PE に割り当てたときの PE の利用状況が電力テーブルに存在しているかを確認し，存在した場合，タスクを割り当てる．存在しなかった場合，タスクは割り当てずに *Wait* に追加する．
(4)　*Wait* のタスクを *Ready* に追加し，2 の処理に戻る．*Tasks* と *Ready* が空の場合処理を終了する．

4.　実験

提案手法を評価するために，従来手法 [3] との比較実験を行った．なお，本来，従来手法はタスクの実行時間と PE の消費電力の分布として任意の分布を扱えるが，今回は，正規分布に従うものだけを検討する．

4.1　性能評価

まず，提案手法の性能を評価する．実験は 1000 回のモンテカルロシミュレーションで提案手法の歩留まりを計算し，従来手法との比較を行った．

入力に用いたデータフローグラフを図 3 に示す．また，各タスクのタスクタイプと実行時間の分布を表 3 に，PE タイプによる電力消費分布を表 4 に示す．ここで，$N(\mu,\sigma)$ は，平均 μ，標準偏差 σ の正規分布を表わす．また，PE は全部で 8 個あり，タイプ A が 3 個，タイプ B が 3 個，タイプ C が 2 個で構成されているものとする．そして，デザイン制約として，電力制約を 65，実行時間制約を 190 とし，歩留まりは 90%を満たすとする．この実験においては，電力テーブルの評価値は $a = 2$，すなわち，$\mu + 2\sigma$ で考えた．

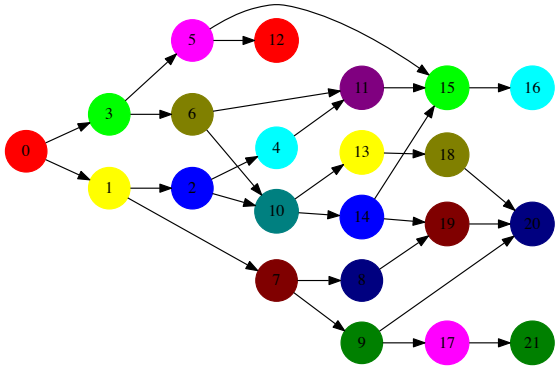

図 3　データフローグラフ

表 3　各タスクのタイプ及び実行時間分布

ID	type	Distribution	ID	type	Distribution
0	A	$N(10, 0.5^2)$	11	C	$N(25, 0.07^2)$
1	B	$N(23, 2.3^2)$	12	C	$N(2.7, 0.07^2)$
2	A	$N(12, 0.6^2)$	13	C	$N(26, 0.07^2)$
3	C	$N(30, 0.07^2)$	14	B	$N(22, 2.2^2)$
4	A	$N(17, 0.85^2)$	15	A	$N(18, 0.9^2)$
5	A	$N(14, 0.7^2)$	16	C	$N(30, 0.07^2)$
6	B	$N(16, 1.6^2)$	17	B	$N(22, 2.2^2)$
7	C	$N(29, 0.07^2)$	18	B	$N(24, 2.4^2)$
8	B	$N(21, 1.05^2)$	19	A	$N(19, 0.95^2)$
9	B	$N(23, 2.3^2)$	20	B	$N(25, 2.5^2)$
10	A	$N(11, 0.55^2)$	21	C	$N(28, 0.07^2)$

Vol.2016-ARC-219 No.33
Vol.2016-SLDM-175 No.33
Vol.2016-EMB-40 No.33
2016/3/25

表 4　PE の電力消費分布

PE type	Power variation
A	$N(10, 1.00^2)$
B	$N(15, 2.25^2)$
C	$N(20, 2.60^2)$

実験結果を図 4 に示す．実験結果によると，提案手法と従来手法のいずれもが歩留まり 90%を達成できている．これは，提案手法においては，電力制約を満たす PE の組み合わせしか利用しないことの効果であると考えられる．一方で，提案手法は従来手法より短かい実行時間で歩留まり 90%を満たしている．これは，主に動的スケジューリングにより，実行時間が短かくなることに起因していると考えられる．

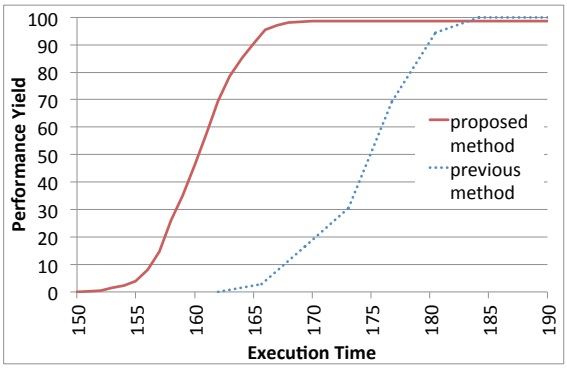

図 4　性能評価

4.2　PE の構成による性能評価

次に，PE の構成による違いを確認する実験を行なう．電力テーブルの評価値は先程と同様 $\mu + 2\sigma$ とし，デザイン制約，データフローグラフも 4.1 節と同様とする．以下，PE 構成として，タイプ A が a 個，タイプ B が b 個，タイプ C が c 個を Proc(a, b, c) と表記する．この実験においては，Proc$(1, 4, 3)$，Proc$(2, 3, 3)$，Proc$(2, 4, 2)$，Proc$(3, 3, 2)$ を利用した．それぞれの実験結果を図 5，図 6，図 7，図 8 に示す．

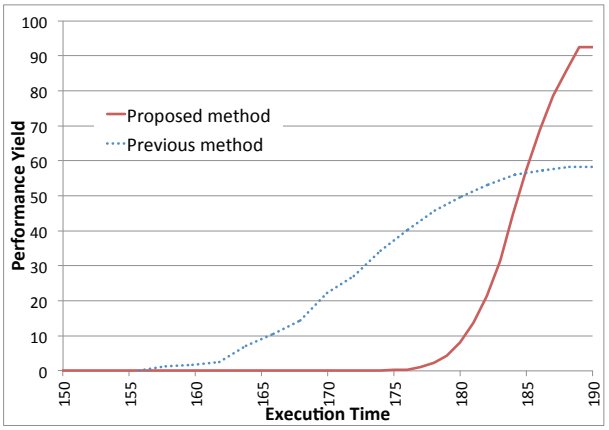

図 5　Proc$(1, 4, 3)$ における性能比較

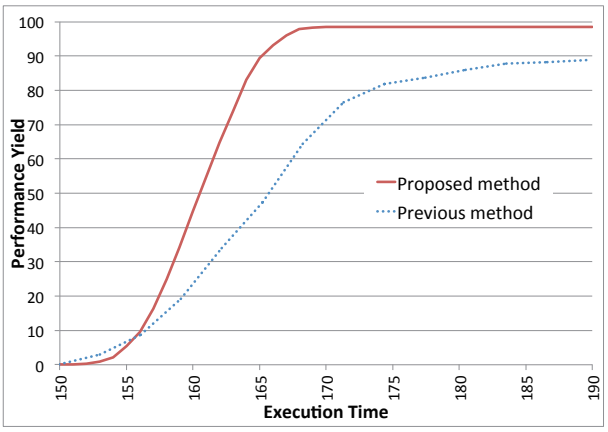

図 6　Proc$(2, 3, 3)$ における性能比較

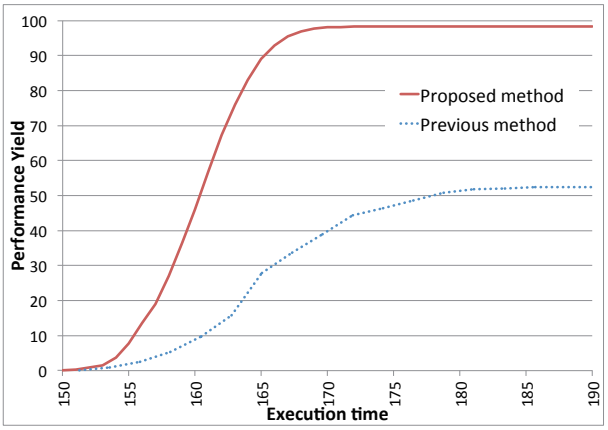

図 7　Proc$(2, 4, 2)$ における性能比較

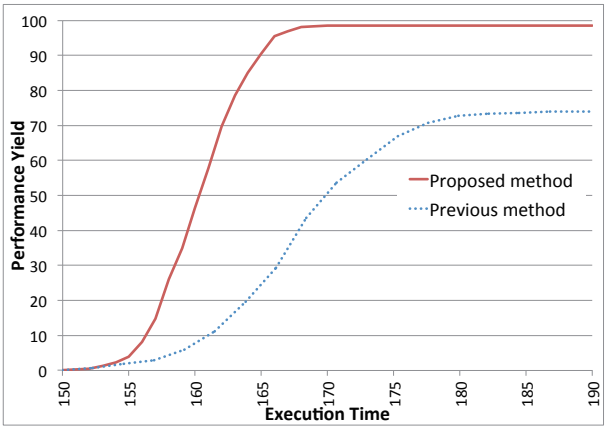

図 8　Proc$(3, 3, 2)$ における性能比較

図 5，6，7，8 の結果から，従来手法がどの PE 構成でも歩留まり制約を満たせていないのに対して，提案手法はすべての PE 構成において歩留まり制約を満たせている．これは，提案手法が与えられた構成の中で電力制約を満たす構成だけを利用して，TAS を実現していることが原因の一つと考えられる．なお，同じ Proc$(3, 3, 2)$ において，図 4 と図 8 の従来手法の結果が一致しないのは，電力制約のパラメータの設定の違いによると考えられる．

また，図 5 に注目すると，従来手法は短かい実行時間か

ら歩留まりが上がり始めるが，最終的に歩留まりは 60%までしか到達できていない．これに対して提案手法は比較的大きな実行時間から歩留まりが上がり始めるが，最終的には歩留まりは 90%まで達している．これは，従来手法では，スケジュールの段階で電力制約を考慮していないため，短かい実行時間の TAS を生成し，その中に偶然電力制約を満たす TAS が含まれることがあるからと考察する．しかし，殆どの場合電力制約に違反するため，歩留まりが目標値までは到達できない．一方，提案手法では，電力テーブルにより使える PE を制限しているため，電力制約を違反する確率の高い実行時間の短かい TAS はあらかじめ生成されないようになっており，結果的に歩留まり制約を満たすことができている．

以上より，提案手法は有効であると考えられる．

4.3 電力テーブル評価値による性能変化

提案手法では，電力テーブルを作成することで電力制約を考慮している．そのため，電力テーブルによって性能が大きく変化する．また，電力テーブルは評価値 $\mu + a\sigma$ に依存して作成される．ここで，あまり悲観的な見積もりにより，利用可能な PE を制限しすぎると，実行時間が長くなってしまう．本稿の問題では，歩留まりは電力制約と実行時間制約を考慮するとしているので，結果として歩留まりが小さくなってしまう．したがって，電力テーブル評価値 $\mu + a\sigma$ の a を適切に決めることが重要となる．

そこで，電力テーブルの評価値による歩留まりの変化を知るためにパラメータを変化させて実験を行った．実験の入力，PE 構成，デザイン制約は 4.1 節と同様とし，電力テーブル評価値 $\mu + a\sigma$ を $a = 3$，$a = 2$，$a = 1.25$，$a = 1$ と変化させた結果を図 9 に示す．

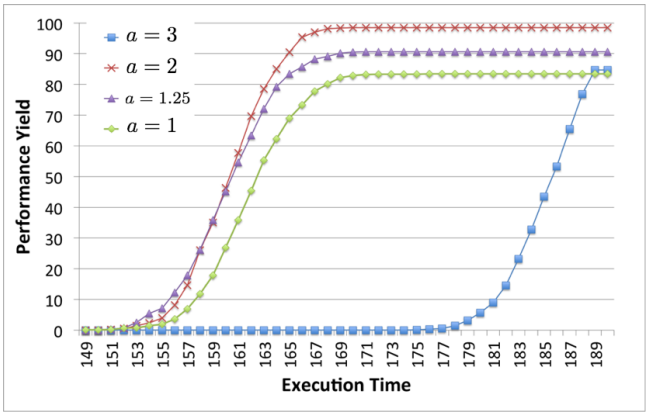

図 9 電力テーブル値による性能変化

図 9 の結果から，電力テーブル評価値によって性能が大きく変化していることがわかる．まず，$a = 3$ のとき，歩留まり制約を満たすことができていない．これは，電力テーブルで許される PE の組み合わせが制限されすぎた結果，実行時間制約を満たすことが難しくなっているからである．一方，$a = 1$ のときも，歩留まりを満たすことができていない．これは，電力テーブルが大きいため実行時間制約を満たすことができているが，電力の見積もりが甘く PE の組み合わせの制限が足りずに電力制約違反が多く起きているからである．最後に，$a = 1.25, 2$ のとき，どちらも歩留まり制約を満たすことができている．これは，電力制約と実行時間制約を共に高確率で満たすことができているからである．また，$a = 2$ と $a = 1.25$ のときを比べると，$a = 2$ のときのほうが短かい実行時間で歩留まり制約を満たすことができている．そして，最終的な歩留まりも $a = 2$ の方が高い．これは，$a = 1.5$ は，実行時間の短かい TAS を生成するが，電力見積もりにおいてマージンが小さく，消費電力が電力制約ぎりぎりとなり，違反する確率も高まるため，結果として歩留まり低下を引き起こすからと考えられる．以上より，係数 a の値の決定が重要であることがわかる．

5. 結論

本稿では，タスクの実行時間と PE の消費電力がばらつくシステムにおいて，実行時間と消費電力のデザイン制約を満たすことのできるスケジューリング手法を提案した．そして，提案手法の有用性を確認するために従来手法との比較実験を行った．その結果，提案手法は従来手法に比べ実行時間の短かい TAS が多く含まれていることから効率が向上していることを確認した．また，PE 構成を変化させて従来手法との比較実験を行った結果，提案手法はどの PE 構成においても有効であることが確認した．最後に，電力テーブルの評価値 $\mu + a\sigma$ の a を変化させて実験を行った．実験から，a を適切に決めることが提案手法の性能を向上させるために必要であることがわかった．

今後の課題として，上記した電力テーブルの評価値の設定を入力に対して適切に決めることが挙げられる．さらに，本研究は消費電力とタスクの実行時間を正規分布と仮定しているので，他の分布に対しての利用方法の検討が必要である．

参考文献

[1] Wataru Shimoyama, Shuji Tsukiyama, and Yusuke Takagi, “An Implementation of a Statistical Static Timing Analyzer Considering Path-Delay Correlation and Evaluation, ” 電子情報通信学会論文誌 A Vol.J90-A NO.11 pp.826–838, 2007.

[2] Charles E. Clark, “The Greatest of a Finite Set of Random Variables,” Operations Resarch, Vol.9, No.2, pp.145–162, 1961.

[3] Mingsong Chen, Daian Yue, Xiaoke Qin, Xin Fu, and Prabhat Mishra, “Variation-Aware Evaluation of MPSoC Task Allocation and Scheduling Strategies using Statistical Model Checking,” in Proceeding of Design Automation & Test in Europe Conference & Exhibition, pp.199–204, 2015.

[4] Komei NOMURA and Yasuhiro TAKASHIMA, “List-sheduling for tasks with execution time variation, ” 信学技報, VLD2014-184, pp.177–182, 2015.