

# 計算機の効率運用のためのジョブスケジューリング手法

4S-1

諏訪 有美 小野山 隆 中重 亮 岡本一弘  
日立ソフトウェアエンジニアリング(株)

## 1 まえがき

計算機ジョブのスケジューリング問題は、古くからある問題であり、その主な目的は、ジョブの先行/後続関係の定義の順に、ジョブを自動起動することであった。しかし近年、計算機業務の多様化により、ジョブの前後関係だけではなく、計算機リソースの複雑な制約を守ったスケジューリングを作成し、計算機リソースの有効利用を図ることが必須となってきた。

本論文は、計算機の効率運用を目的とする計算機リソース有効利用のためのスケジューリング手法の研究に関するものである。

## 2 計算機ジョブスケジューリングの従来方法における問題点

従来のスケジューリング方法では、図1の例のようにジョブ実行時にリソース待ち状態に陥るジョブが発生してしまう。

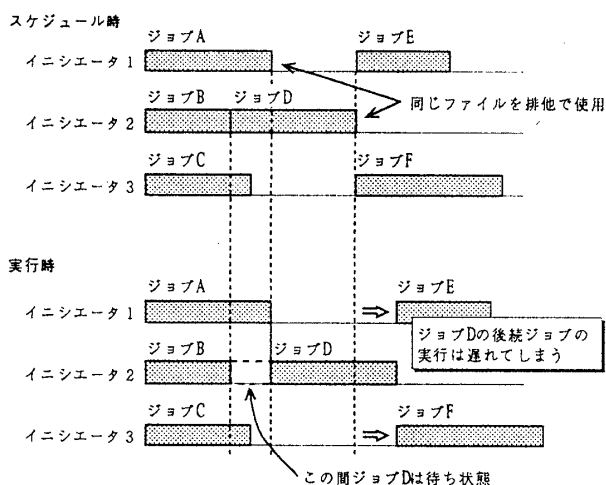


図1: リソース待ちの例

Job Scheduling Method for Efficient Computer System Operation  
Yumi SUWA, Takashi ONOYAMA, Ryo NAKASHIGE, Kazuhiro OKAMOTO  
Hitachi Software Engineering Co., Ltd.

nリソース待ちにはこの例のようなファイル待ちの他に、ディスクスペースやMTデッキなどを確保出来ずに待ち状態になる場合がある。このような待ちが発生した場合、その後のジョブ全体に影響を与えるので、結果としてスケジューリング通りの時間に作業を終了することが出来なくなる。

今回考えたスケジューリング手法の目的の1つは、スケジューリング作成時に計算機リソースを考慮し、このような不都合をなくし、計算機運用業務の作業見通しをよくすることである。しかし、計算機リソースを考慮することによって実質的な計算機の稼働時間が長くなってしまった場合は、結局計算機リソースの無駄使いになり、人手を煩わすことにもなる。この問題において重要なのは、如何にしてジョブの実行総時間を短くし、無駄なく計算機資源を使用するためのスケジューリングを作成するかである。

## 3 ジョブスケジューリング問題のヒューリスティクスによる解法

スケジューリング問題の最適解を求める手法に、Simulated Annealing法やGenetic Algorithmを適用した研究[1]をよく見かけるが、これらはいずれもスケジューリングに多くの時間を要する。その意味で、計算機ジョブスケジューリング問題にとっては、例えばジョブの実行総時間を短くしたスケジューリングを作成出来たとしても、最適とはいえない。

本研究ではジョブの割り当て順序をヒューリスティクスにより決定することで、ジョブの実行総時間をなるべく短くする方法を採用し、ジョブスケジューリングのプロトタイプを作成した。

ジョブスケジューリング問題が、単にジョブの実行を行う複数のイニシエータへのジョブの割り当てである場合、時間の長いジョブから順番に割り当てていけばジョブの総実行時間を出来るだけ短く出来るだろうことは容易に想像できる。本研究では、ジョブ同士が先行/後続関係を持つ場合のいくつかのヒューリスティクスの効果の比較と、更にジョブが複数種類のリソースを使用する場合にもその効果が表れるかどうかの評価実験を行った。

評価実験で用いたヒューリスティクスは次の5種類である。

1. processing-time 処理時間の長いジョブから順に割り当てる。
2. post-jobs  
直後のジョブ数が多いジョブから順に割り当てる。
3. later-jobs  
全後続ジョブ数が多いジョブから順に割り当てる。
4. height  
そのジョブを始点とするパス中のジョブの数が大きい順に割り当てる。
5. longest-path  
そのジョブを始点とするパスの処理時間の合計が最大のジョブから順に割り当てる。

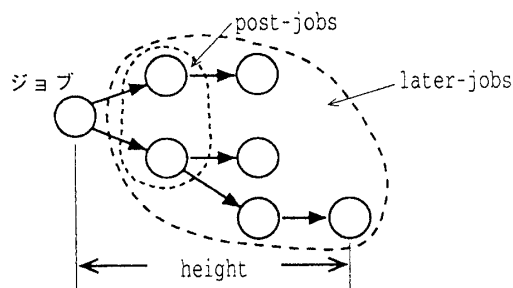


図 2: ヒューリスティクスの例

#### 4 実験結果

各ヒューリスティクスの効果を比較するために、100 ジョブのスケジューリング問題を 1000 種類ランダムに生成し、評価実験を行った。その結果を表 1~3 に示す。

表 1, 2 によると、どちらの場合も processing-time による実行総時間が一番長く、longest-path が平均が一番短い。しかし、processing-time 以外のヒューリスティクスはあまり変わらないことがわかる。

各実行総時間の平均だけでは、各問題毎の結果がどれほどヒューリスティクスによって違うかはわからない。そこで、5 種類のヒューリスティクスによる 5 つの解のうち最良解とどれだけ近いかを比較したものが、表 3 である。最良解との実行総時間の差の最良解に対する割合を表したものである。

表 3 によると、やはり processing-time 以外の 4 種類が効果的であることがわかる。中でも longest-path が有効だが、他の 3 種類に比べて際だって優れているという程ではない。

表 1: リソースを使用しない場合  
ジョブの実行総時間 (単位: 分)

ヒューリスティクス	平均	最小	最大
processing-time	99.955	66	149
post-jobs	87.498	56	127
later-jobs	86.26	58	126
height	86.128	57	126
longest-path	85.39	54	126

表 2: 複数種類のリソースを使用する場合  
ジョブの実行総時間 (単位: 分)

ヒューリスティクス	平均	最小	最大
processing-time	159.208	109	231
post-jobs	144.49	102	211
later-jobs	140.865	101	216
height	140.388	102	206
longest-path	139.039	99	199

表 3: 最良解との比較  
(単位: %)

ヒューリスティクス	(T-BT)/BT × 100	
	リソース有り	リソース無し
processing-time	19.46	20.24
post-jobs	4.07	9.01
later-jobs	2.57	6.25
height	2.42	5.91
longest-path	1.44	4.87

T: 実行総時間  
BT: 最良解の実行総時間

#### 5 まとめ

本研究では、計算機ジョブのスケジュールにリソースなどの複雑な条件を取り入れ、更に計算機の効率運用のためのスケジュールを作成するヒューリスティクスを得た。ここで得られた結果からわかることは、ジョブ間の関係を考慮したヒューリスティクスを導入することにより、ジョブの実行総時間を短くすることが出来るということである。しかし、実用化するには更に様々なデータでの評価実験により、各問題の性質毎に具体的にどのヒューリスティクスが適しているかを検証する必要がある。

#### 参考文献

- [1] Coroyer, Liu : Effectiveness of Heuristics and Simulated Annealing for the Scheduling of Concurrent Tasks - An Empirical Comparison : Rapport INRIA No.1379 (1991)