



Title	E-NETによる計算機システムのシミュレーション
Author(s)	橋津, 正晶; 阿部, 政樹; 栃内, 香次; 永田, 邦一
Citation	北海道大學工學部研究報告, 78, 59-67
Issue Date	1976-02-16
Doc URL	<a href="http://hdl.handle.net/2115/41336">http://hdl.handle.net/2115/41336</a>
Type	bulletin (article)
File Information	78_59-68.pdf



[Instructions for use](#)

## E-NET による計算機システムのシミュレーション

橋津正晶\* 阿部政樹\*\*

栃内香次\* 永田邦一\*

(昭和 50 年 6 月 30 日受理)

### A Computer System Simulation Using Evaluation Nets Method

Masaaki HASHIZU Masaki ABE  
Koji TOCHINAI Kuniichi NAGATA

(Received June 30, 1975)

#### Abstract

Modern computer systems are growing increasingly complicated and it is difficult to measure and evaluate characteristics such as throughput, turnaround time, and utilization of resources. For measuring these characteristics, various approaches have been developed and are considered to be useful in evaluating computer systems. Of these useful approaches, simulation is the most popular and effective evaluating method.

This report discusses a computer system simulation using E-Nets (or Evaluation nets) method, which has been developed for the use of the representation of computer systems. The model discussed in this report is constructed for the FACOM 230/60 system and 230/75 system of the Hokkaido University Computing Center, and its simulation is experimentally executed. Consequently, it was found that this model reflected the figure of the real system, and it was shown that E-nets method was effective for constructing simulation models.

Therefore these simulation results will be available for improving the efficiency of the computer system of the Hokkaido University Computing Center.

#### 1. 序 言

計算機システムの性能や効率の改善を目的として、システムの効率等の測定や評価がしばしば行われる。その際の評価手法として

- (1) 理論解析による方法
- (2) 実物テストによる方法
- (3) シミュレーションによる方法

の 3 つが考えられる。(1) の理論解析による方法は取り扱いの対象が限定され、複雑なシステムの解析は困難であり、(2) の実物テストによる方法は良い結果を得られるが、計算機センターなどで

\* 工学部電子工学科 電子機器工学講座

\*\* 三菱電機株式会社

\*

\*\*

は日常業務を中断させなければならないし、また時間と費用がかかり過ぎる欠点がある。これに対して、(3)のシミュレーションによる方法は結果の精度と信頼性に問題があるが、モデル化に柔軟性があり、求める結果とも関係するが、比較的、低コストで済むという利点を持つ。

このように、シミュレーションは効率改善における評価のための全ての目的に利用しうる有力な手段ではあるが、プログラミング及びモデル化の点でその作製がかなり困難であるといった問題がある。プログラミングやモデル化に関しては、普通、汎用シミュレータや専用シミュレータを用いることで比較的、楽にはなる。しかし、これらはモデルの対象が限定されていたり、一般のユーザには余り普及していないため、使用の際には様々な不都合な点が生じて来る。

このため、計算機システムのシミュレーションを行う際に上述のシミュレーション言語を使用せず、FORTRANなどの汎用言語を用いて簡単にシミュレーションプログラムを作製できるようなモデリング手法の必要性は大きい。その様なモデリング手法として、最近、E-NET手法が目目されており、これを用いて北大大型計算センターのFACOM 230/60システムをモデル化し、シミュレーションを行ってみた。その結果、システム効率改善に役立つような種々のデータが得られ、また、E-NET手法の有効性等が確かめられたので、ここに報告する。

## 2. E-NET の概説

E-NET (Evaluation net) は Petri net を母体とし、modified Petri net を経て J. D. Noe, G. J. Nutt たちによって計算機システムを記述するためのモデルを目的として開発された<sup>1)~4)</sup>。E-NET の特徴は、モデルの表現が実際のシステムのハードウェア構成に即した形で出来ること、しかも図的表現を用いるためシステムの動作、並びに構成が直観的に把握でき、またフローチャートでは表現出来ないようなデータ・フローも表現できることなどである。

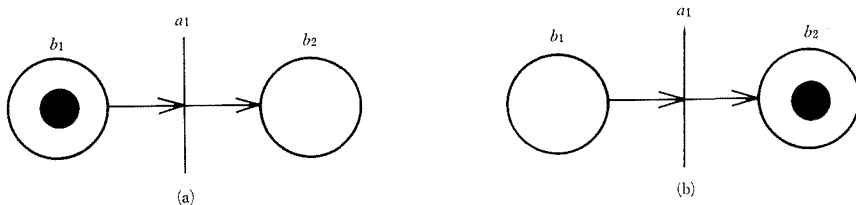


図1 トークンの移動

E-NET は、いくつかの基本遷移型の結合で表わされ、基本遷移の各々はトークン、ロケーション、遷移、矢線の要素で構成される。以下にそれぞれの構成要素の意味を示す。

- (1) トークンは計算機内部でのジョブやタスクを意味し、トランザクション (時間と共にシステム中を動いていく対象をモデル化したもの) などに相当する。(図1では黒点に相当する)
- (2) ロケーションはある期間存続する条件を表わし、円あるいは六角形で表わされる。(図1では  $b_1, b_2$  に相当する)
- (3) 遷移はトークンの流れや変更が起こる時の NET 中の節点であり、縦線で示される。(図1では  $a_1$  に相当する)
- (4) 矢線はロケーション相互を結び、それらの間の入出力関係を表わし、またトークンの移動する経路を示す。

例として、トークンの流れの具体例が図1に示してあるが、ここで(a)ではロケーション  $b_1$  にトークンが入っており、ロケーション  $b_2$  が空である。この時、遷移  $a_1$  は遷移可能であると呼

ばれ、遷移時間  $t(a_1)$  が経過した後、ロケーション  $b_1$  のトークンはロケーション  $b_2$  に移動し遷移  $a_1$  が終了する。この時、図 1(a)は図 1(b)のように変化する。これを遷移の発火と呼ぶ。

E-NET では、このようなトークンの移動に関しては図 2 に示すように、5つの基本的な遷移が考えられており、それぞれに対して遷移規則が与えられている。例えば、F型遷移であれば、ロケーション e 及び f が空でロケーション b にトークンがあればトークンの移動が可能であるといったように遷移に関していくつかの規則がある。図 2 には、5つの基本遷移型を示すと共に、各々の横に遷移規則を示してある。尚、遷移規則の左側の状態が遷移可能な状態を示している。

ところで、図 2 から分るように、5つの基本遷移型の中で、X 及び Y 型遷移は遷移規則が 1 通りではなく、他の 3 つの遷移型とは異なっている。この 2 つの遷移型は、ルートを決したり、トークンの間の競合を整理するのに使用される。例えば、X 型遷移の場合、

入力ロケーション b にトークンがあって、出力ロケーション e が空の時は、ロケーション r が 0 であればロケーション f の状態にかかわらず、必ず b から e にトークンが移動する。移動後、リゾリューションロケーションは、それが内部ロケーション（入出力矢線の両方に結ばれたロケーション）のときは“0”にされ、ペリフェラルロケーション（入出力矢線のいずれか一方に結ばれたロケーション）のときは未定義に置かれる。未定義に置かれたリゾリューションロケーションの値は、それ以外の入出力ロケーションが遷移規則のいずれかに一致した（擬似的可能段階と呼ぶ）ときに、改めて“e”の値がある手続きによって、“1”か“0”のいずれかに決定され次の遷移が起こる。この“e”の値を決定する手続きをリゾリューション手順の評価と呼ぶ。

E-NET モデルを実際に動作させる時は、それを構成している各遷移の 1 つ 1 つに対して遷移可能か否かのチェックや遷移時間のチェックを行ったり、リゾリューション手順の評価を行ったりする必要が出て来る。従って、E-NET を完全に記述するためには、図的表現だけでは不十分で、もう少し詳細な部分を記述するために、形式的な表現が用意されている。例えば、遷移  $a_i$  は次のように表現される。

$$a_i = (S, t(a_i), q) \quad S: \text{遷移型}, t(a_i): \text{遷移時間}, q: \text{遷移手順}$$

またリゾリューション手順の評価は次のように表現される。

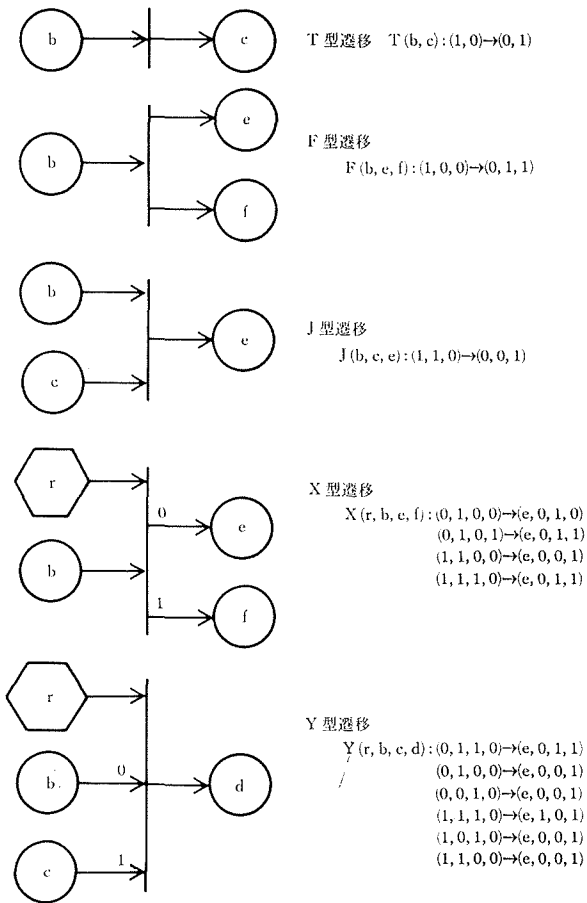


図 2 基本遷移型

$$r: [p_1 \rightarrow M(r) := i; p_2 \rightarrow M(r) := 1-i] \quad p_i: \text{述語}, M: \text{ロケーションの値}, i: 0 \text{ か } 1$$

上式の意味は  $p_1$  で示した述語 (関連ロケーションなどの条件を記述したもの) が真であるならばロケーションの値を  $i$  にし, そうでなければ  $p_2$  を評価して真であるならば  $1-i$  にせよということである。  $p_1$  及び  $p_2$  のいずれもが偽であればリゾリューションロケーションの値は未定義のままである。

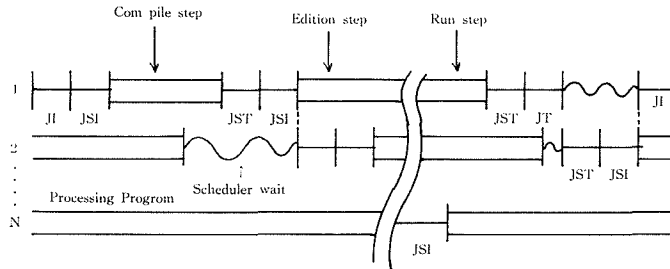
### 3. モデルの作製

#### 3.1 FACOM 230/60 システムのモデル化

モデル化の対象は前述のように, 北大大型計算機センターの FACOM 230/60 システムである。このシステムは CPU が 2 台のマルチプロセッサシステムで, 主記憶装置の容量は 160 kW (1 W=36 ビット) である。その内, ユーザ領域は 93 kW で, 残りの部分はシステム領域である。また, システムプログラムや処理されるジョブは, 各補助記憶装置 (磁気ドラム, 磁気テープ, ディスクパック) に格納されている。

尚, モデルの作製の基礎としたデータは, アカウント情報 (いくつかのジョブについて個々の CPU 使用時間や CORE 使用時間などを実測したデータの集計) であり, これを用いてジョブが SYSIN (入力装置から入力されて, CPU による処理が始められる前に, 最初に格納されるファイル) QUEUE から主記憶に読み込まれたところから, CPU による処理を完了するまでをモデル化した。

次に, モデル化した部分におけるジョブのふるまいについて概説する。ジョブは最初 SYSIN ファイルに置かれており, そのジョブはバッチジョブ・イニシエータにより実行開始が準備される。そして, ジョブ・ステップ・イニシエータ, 処理プログラム (ジョブ・ステップ), ジョブ・ステップ・ターミネータからなるサイクルをジョブ・ステップの回数だけ繰り返した後, ジョブ・ターミネータにより終了する。このようなジョブの流れが設定された多重度の数だけでき, スケジューラ (イニシエータやターミネータを指す) は同時に 2 つ以上のジョブにサービスを行うことはない。多重度  $N$  で処理されるジョブの流れを図 3 に示す。



JI: Job Initiator                      JT: Job Terminator  
 JSI: Job Step Initiator              JST: Job Step Terminator

図 3 マルチプログラミングにおける  $N$  個のジョブの流れ

次に, モデルを作製した時に用いた仮定を以下に記す。

- (1) RELBIN ファイル (リロケータブル・バイナリ形式のプログラムが格納されるファイル) 及び EXELIB ファイル (リロケータブル・バイナリ形式のプログラムが結合編集され格納されるファイル) はジョブを 30 個まで収容可能である。

- (2) スケジューラの走行時間は入出力時間のみからなる。
- (3) 制御プログラムの CPU 使用時間は充分小さいものとして無視する。
- (4) 主記憶の空き領域は連続である。
- (5) 入出力はブロック化されており、1 回の入出力量及びアクセス時間は一定とする。
- (6) ロールイン/アウト機能を省略する。

以上の仮定を前提として FACOM 230/60 システムを E-NET モデルにより記述した。その E-NET の図的表示を図 4 に示す。

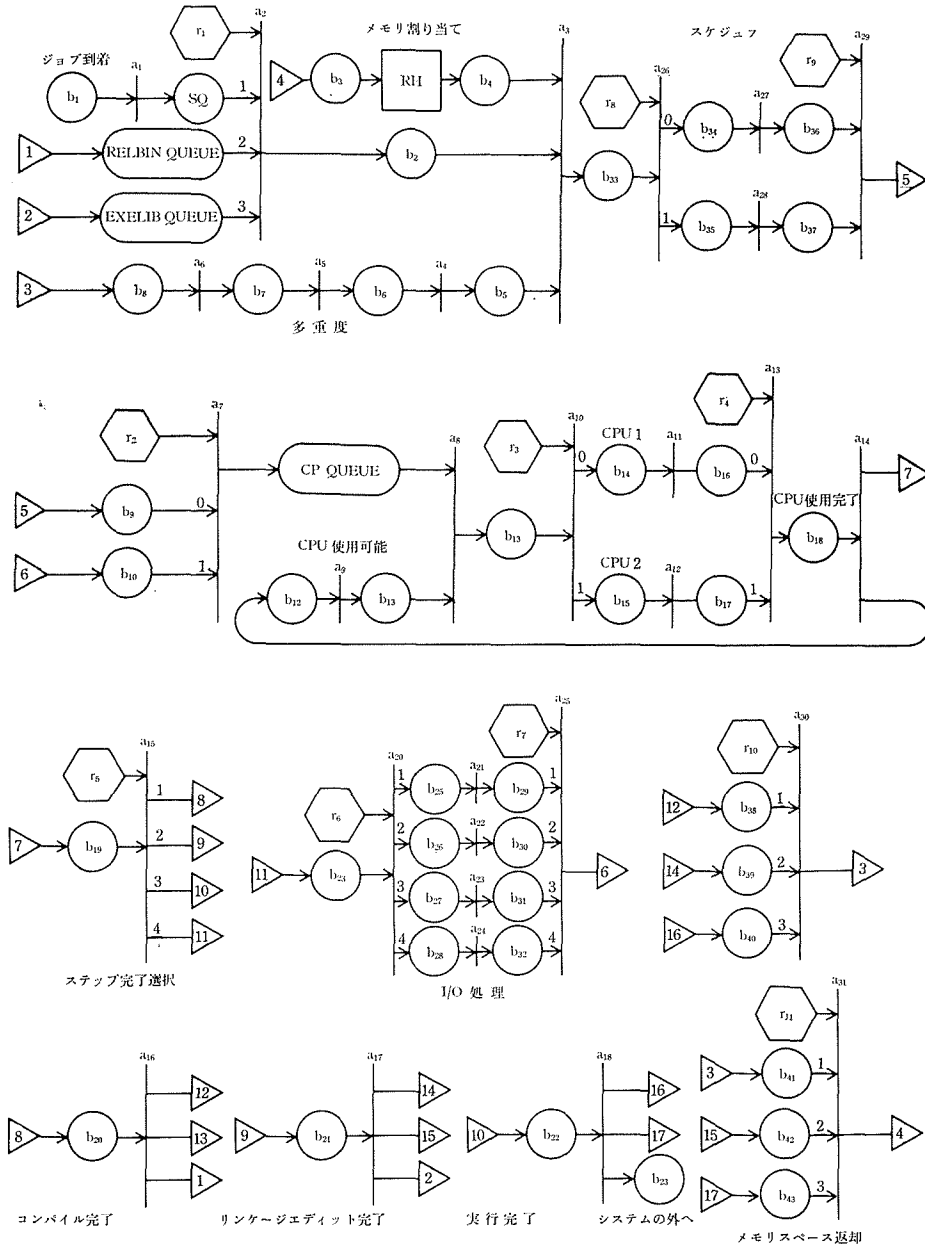


図 4 E-NET による FACOM 230/60 システムのモデル化

### 3.2 シミュレータの入力データ及びシミュレーションプログラム

アカウント情報のうち、入力データとして用いたのはジョブの優先度、CPU 使用時間、CORE 使用時間である。ところで、シミュレータを実際に動作させるためには、アカウント情報の他に各ジョブ・ステップにおける入出力回数及び入出力間隔とスケジューラ時間とが必要である。後者については京都大学において実測されたデータがあったのでそれを参考にした。前者については北大大型計算機センターの FACOM 230/60 システムの主記憶装置が小さいことによる影響の反映として各ジョブ毎におおよその入出力回数が求められることが分っている<sup>5)</sup>。

このことから、入出力時間を主メモリ占有時間と CPU 時間の差と考え、これを単位入出力時間で割ることにより、入出力回数を求めた。入出力間隔については、入出力回数で割った値を平均値とする負の指数分布に従うものと仮定した。

シミュレーションプログラムは、メインプログラムと 4 個のサブルーチンプログラムとから構成されており、各サブルーチンは E-NET の形式的表現に従ってコーディングされている。尚、出力は各ジョブの処理開始時間、処理終了時間とシステム運転時間中の各多重度毎の累積時間である。

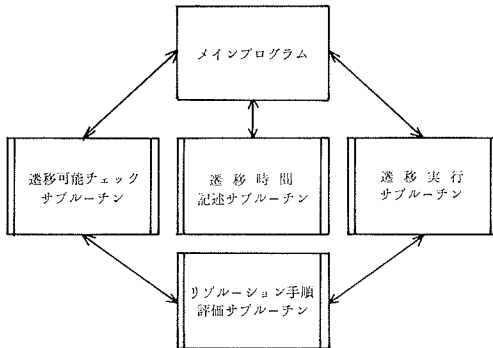


図5 各プログラム間の関係

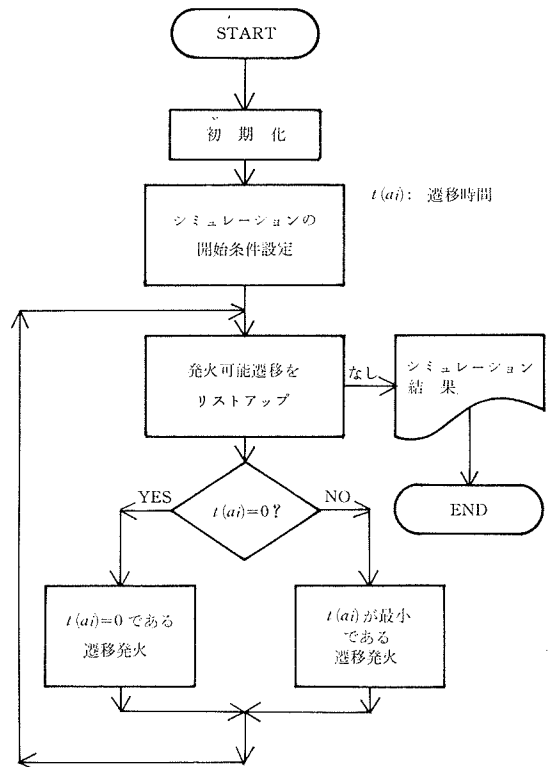


図6 メインプログラムのフロー

## 4. シミュレーション結果

シミュレーションは2種類のアカウント情報を入力データとして用い、行われた。その1つは、北大大型計算機センターでシステムの効率等を予測するのに使用しているテストプログラム群で、ジョブ数は50個である。もう1つは、システムの運転時に実際に測定したデータであり(1974年10月22日に測定)、ジョブ数は359個である。

シミュレーションに際して、単位入出力時間は、50ジョブデータの場合が300ms、3000msとし、359ジョブデータの場合は3000msとした。シミュレーション結果からみて、この値は300ms程度にとるのがよく、3000msとすると結果の誤差が大きくなる。しかし、シミュレーションに要する時間が非常に短くなるので、おおよその結果をほしいときには3000msでもよいと思われる。

このシミュレーションでは、モデルにロールイン/アウト機能を取り入れなかったので、モデルの妥当性は運転時間のアカウント値とシミュレーション値との相対誤差から判断することにした。(表1参照) その結果、単位入出力時間が300 msの場合については、相対誤差も比較的小さいので、モデルが対象システムをマクロ的に充分近似しているものと考えられる。

次に表2には単位入出力時間が3000 msの場合の多重度とそれに対応する累積時間とを示してある。この表から多重度を求めると、約2.151、アカウント情報から平均多重度を求めると約1.98となり、ほぼ一致した値と考えられる。しかも、シミュレーションでは多重度毎の累積時間を計算するのにスケジューラ時間も加えていることから、これを差し引いた平均実効多重度は2.151よりも小さくなり、モデルの妥当性を支持するものと言える。

表1 運転時間のシミュレーション値 (50 ジョブの場合)

	アカウント 情報	300 ms	300 ms	3000 ms	3000 ms	3000 ms
乱数の初期値	—	5	1234567	3	5	1234567
運転時間(秒)	4321	4219	4216	3717	3782	4071
相対誤差(%)	—	2.4	2.4	13.9	12.4	5.8

表2 多重度毎の累積時間

多重度	累積時間 (秒)	比率 (%)
4	0	0
3	812	19.2
2	3231	76.7
1	173	4.1
0	0	0

次に、ジョブ毎のターンアラウンド時間のアカウント値とシミュレーション値との偏差について調べると、図7のようになる。この図で、シミュレーション値とアカウント値の誤差の大きなものが、正負の両方に少数個見られるが、この理由としては次のことが考えられる。

(1) モデルではロールイン/アウト機能や制御プログラムのCPU使用時間を無視しているが、これらの影響が無視できないジョブがある。これらのジョブではシミュレーション値は実測値より小さくなる。

(2) I/O時間及びI/O間隔を求めるために用いた仮定が不成立なジョブがある。これらのジョブでは(1)とは逆にシミュレーション値の方が実測値よりも大きくなる。

また、359ジョブデータの場合のシミュレーションについても、同様の結果が得られた。その結果、運転時間に対する相対誤差が-3.6%で、実効多重度はアカウント値から得られた値が

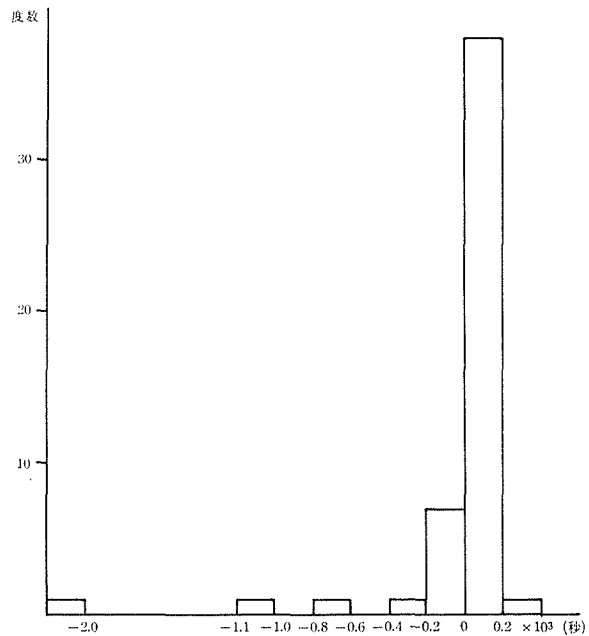


図7 ターンアラウンド時間の偏差に関する  
ジョブ毎の度数分布



1.77であるのに対し、シミュレーション値から得られた値が1.82で非常に良く一致している。

北大大型計算機センターでは、1974年秋にFACOM 230/60から同230/75へとシステムの変更が行われた。そこで、このシミュレーションの応用として、対象システムをFACOM 230/75 (CPU 1台)におきかえ、設定多重度と50個のジョブを処理するのに要する時間との関係を求めてみた。ここでは、CPU速度をFACOM 230/60システムの6倍とし、他の要素、入出力装置の構成はFACOM 230/60システムの場合と同じで、主記憶のユーザ領域は500 kWであるとしている。この結果を図8に示す。これを見ると、次のようなことが言える。即ち、比較的多重度が低い時は主メモリが充分あるので、設定多重度を上げてやることにより、CPUの遊びが減るため処理時間が急激に減少する。それに対し、多重度6~10までは入出力機品の制約と主記憶内のジョブ間の相互作用により処理時間の減少は前に比べて激しくはなく、多重度が10を超えるあたりから入出力機器とジョブ間の相互作用の影響により、処理時間が増加し始める。

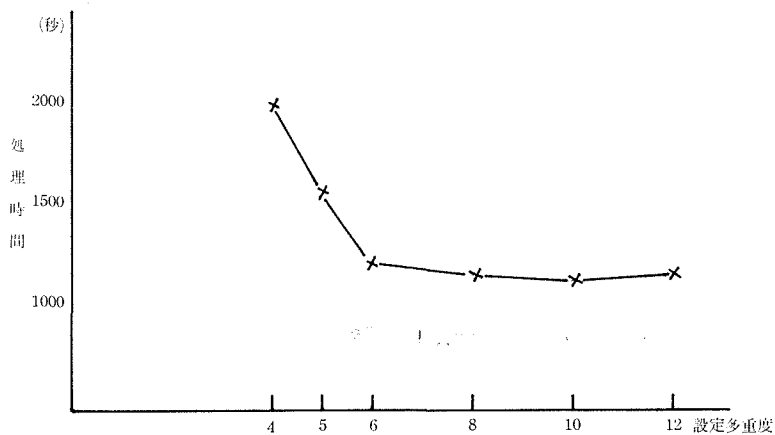


図8 設定多重度と処理時間との関係

## 5. 結 言

E-NET手法のモデル化により、計算機システムのシミュレーションを行った結果、次のようなことが確かめられた。

- (1) この実験で作製したモデルが実際のシステムの動作を良く反映していること。
- (2) モデル作製時に用いた仮定は、ほぼ妥当であること。
- (3) E-NET手法を用いたことにより、モデルの作製が非常に容易に行えたこと。しかも図的表現であるためデータ・フローやコントロール・フローを容易に把握でき、モデルの修正が容易に行えたこと。
- (4) プログラムをコーディングする際にE-NETの形式的表現が役に立ったこと。

今後の課題として、この実験で作製したモデルをいくつかの部分について詳細化すること、そしてE-NET手法における問題点を明確にしていくことが必要と思われる。

## 謝 辞

本研究を行うにあたり、アカウントデータの提供、及び実験結果の検討に御助力下さった北大大型計算機センターの方々、並びに研究室の諸氏に深く感謝いたします。

## 参 考 文 献

- 1) G. J. Nutt: "Evaluation Nets for Computer System Performance Analysis", FJCC Proceedings, 41 (1972), p. 279-286.
- 2) J. D. Noe and G. J. Nutt: "Macro E-nets for Representation of Parallel Systems", IEEE Trans on Computers, C-22 (1973), p. 718-727.
- 3) G. J. Nutt: "The formulation and application of evaluation nets", Computer Science Group, Univ. Washington, TR 72-07-02 (1972).
- 4) G. J. Nutt and J. D. Note: "Some Evaluation Net Macro Structures", Computer Science Group, Univ. Washington, TR 73-01-07 (1973).
- 5) 千葉正喜: "CORE 占有時間と CPU 時間の比による処理多重度と処理効率の予測", 北大大型計算機センター・研究開発部内部資料.
- 6) 金沢正憲, 他: "計算機システムのシミュレーションについて", 情報処理, Vol. 13, No. 10 (1972), p. 684-691.
- 7) 阿部政樹: "E-NET による計算機システムのシミュレーション", 修士論文 (昭 50).