

Line Balancing の Computer 化**

佐 伯 弘 子*

1. ま え が き

自動車とかテレビといった多くの商品は、コンベアによって大量に作り出されており、作業者は流れてくる製品に対して各々決められた仕事を行なっている。

然るに今日のように市場の変化が激しいと、生産台数はもちろん、モデルも始約変更しなければならず、それに伴って各作業員への仕事の割当も変更しなければならない。われわれの会社でも変更の度にこの割当を人間が行っているために、次第にオーバーワークとなり、又必ずしも良い割当を行なっているとはいえないと思う。

そこで、これを計算機でやれないものかと考えたのが標題の「ラインバランシングの Computer 化」である。この研究が、アメリカではじまったのはここ10年間のことで、主な研究は表1の通りである。

各人の成果もちがっているが、ここで書くのは、*印のついている Jackson—Mitchell の流れである。しかし、その前にラインバランシングの概略を簡単に説明しよう。

表1 主 な 研 究

①	1954	Bryton (ノースウエスタン大学)	
			“Convergence Procedure”
②	1955	Salveson	“The Assembly Line Balancing Problem”
③*	1956	Jackson (カリフォルニア大学)	
			“A Computer Procedure for Line Balancing Problem”
④*	1957	Mitchell (W社)	
			“A Computational Procedure for Balancing Zoned Assembly lines”
⑤*	1958	Burgeson (W社) プログラム	
		& Daum	“Production Line Balancing”
⑥*	1959	Hoffman (ウィスコンシン大学)	
	1963		Precedence Matrix による解法
⑦	1960	Bowman (M.I.T.)	L.P. による解法
⑧	1961	Helgeson & Birnie (G.E.)	位置的重みづけによる解法 & Birnie
⑨	1960	Tonge	} Heuristic Method
	1961		
⑩	1961	Kilbridge	
	1962		

* 三菱電機 1964年5月15日 第15回研究会発表 「経営科学」第8巻第2号

2. 普通 の モデル

Cycle time “Co” は1日の実働時間を1日の生産台数で割ったものである。即ち Co 時間に一台の割で生産しようとするから、Co は一人の作業者に許される最大の作業時間である。従って割当てる方からいえば、Co 時間以下でやれる仕事を各人に割当てていかねばならない。

次に、例えば“AとBをハンダ付する”といったように、適当にわけられた要素作業を“Job”と呼ぶ。

各 Job にはそれを行なうのに必要な時間値があり、これを Job time と呼ぼう、例えば“AとBをハンダ付する”という1つの Job があったとして、それに5秒かかるとすれば、この Job time は5秒である。

又 Job 同志の間には、先行関係が存在する。即ちAという Job はBという Job が終わった後で行ない、その逆は許されないという関係である。これは、例えばピンズめを考えると、“中味を入れる”という Job の後に“栓をする”という Job を行なうのであって、その逆は許されない。

このような条件、即ち cycle time Co と、各 Job 間の先行関係を満足するように割当て、しかも作業者の数を minimum にするのが Line Balancing の目的である。

それは Balance idle を minimum にすることと同じである。Balance idle は1台当りの idle time を1台の生産に必要な作業時間で割ったものである。この値は5%といったように%で表示するが、実際問題として、分子の idle time が Co より小さければ、その答は最小人数であるから、最良解といってもよい。

このようにラインバランシングの問題は、ある制限下における順列組合せの問題とみることができる。

Jackson は今まで書いてきたモデルについて、ほぼ満足できる方法を開発したが、まだ計算時間がかかりすぎるという問題を残している。この点及び実際のラインを考慮して、Jackson の方法を発展させたのが Mitchell の方法である。

表2 普通 の モデル

$\text{Cycle time } Co = \frac{\text{実働時間}}{\text{生産台数}} = \text{1人当りの最大作業時間}$
$\text{Job (要素作業)} \begin{cases} \text{Job time } Ti \\ \text{先行関係 } \textcircled{B} \longrightarrow \textcircled{A} \\ \text{中味をつめる} \quad \text{栓をする} \end{cases}$
<p>作業数Nを min にしたい。</p>
$\text{Balance idle} = \frac{Co \times N - \sum Ti}{Co \times N}$

3. Mitchell のモデル

実際に割当を行なう時、いつもライン全部の Job を考慮しなければならないわけではなく、初めの作業員への割当を考えるには、Job も比較的ラインのはじめのもののみ考えればよいわけで、最後の方の Job まで考えに入れる必要はない。

又、製品の位置や設備の制限からも Job の group わけができてくるわけで、これを Mitchell は Zone と名づけた。

実際の適用にはいろいろな使い方があるが、プログラム上 Zone を「1人の作業員に、その Zone 内のどの Job が割当てられても一応実行できる範囲」と定義する。

又、各 Job を Must-do-Job と Can-do-Job とにわけ、Must-do-Job とはその Zone でしかできない Job のことをいい、Can-do-Job は2つ以上の Zone のどこかでやってもよい Job をいう。

この Zone の導入は、機械・設備等、実際のラインの条件からくる当然の制限とも、又可能な組合せの数を減らすのが目的であるともいえる。

この方法が、理論上もほぼ非のうちようがないと思われるが、さらに IBM-650 のライブラリーを実際のラインに適用してみた、かなり効果的であることを確かめることができた。

しかし同時に、コンベア一本で、作業員は一列に並んでいるこのモデルでは、われわれのラインには窮屈であることがわかった。

そこで、各ラインの関係者から要請を出してもらい、それに基づいて IBM-7090 Fortran で PRINCE/MI という名前のプログラムを作った。これは production Line Balance for Mitsubishi の略称である。

PRINCE/MI は実際のラインの要請を実現する必要の他に計算機の機種変更に対処する必要もあったわけで、高速機種への変更により計算時間はかなり短縮され、あまり大きくない普通のラインについては、Jackson 以来の「計算時間がかかりすぎる」という問題は解決されたと思う。

表3 Mitchell のモデル

Zone (職域)	—	1人の作業員に、その Zone 内のどの Job が割当てられても、一応実行できる範囲
Must-do-Job	—	決められた Zone で必ず実行しなければならない Job
Can-do-Job	—	いくつかの Zone のうち、どこかで実行すればよい Job

4. PRINCE/MI での主な改正点

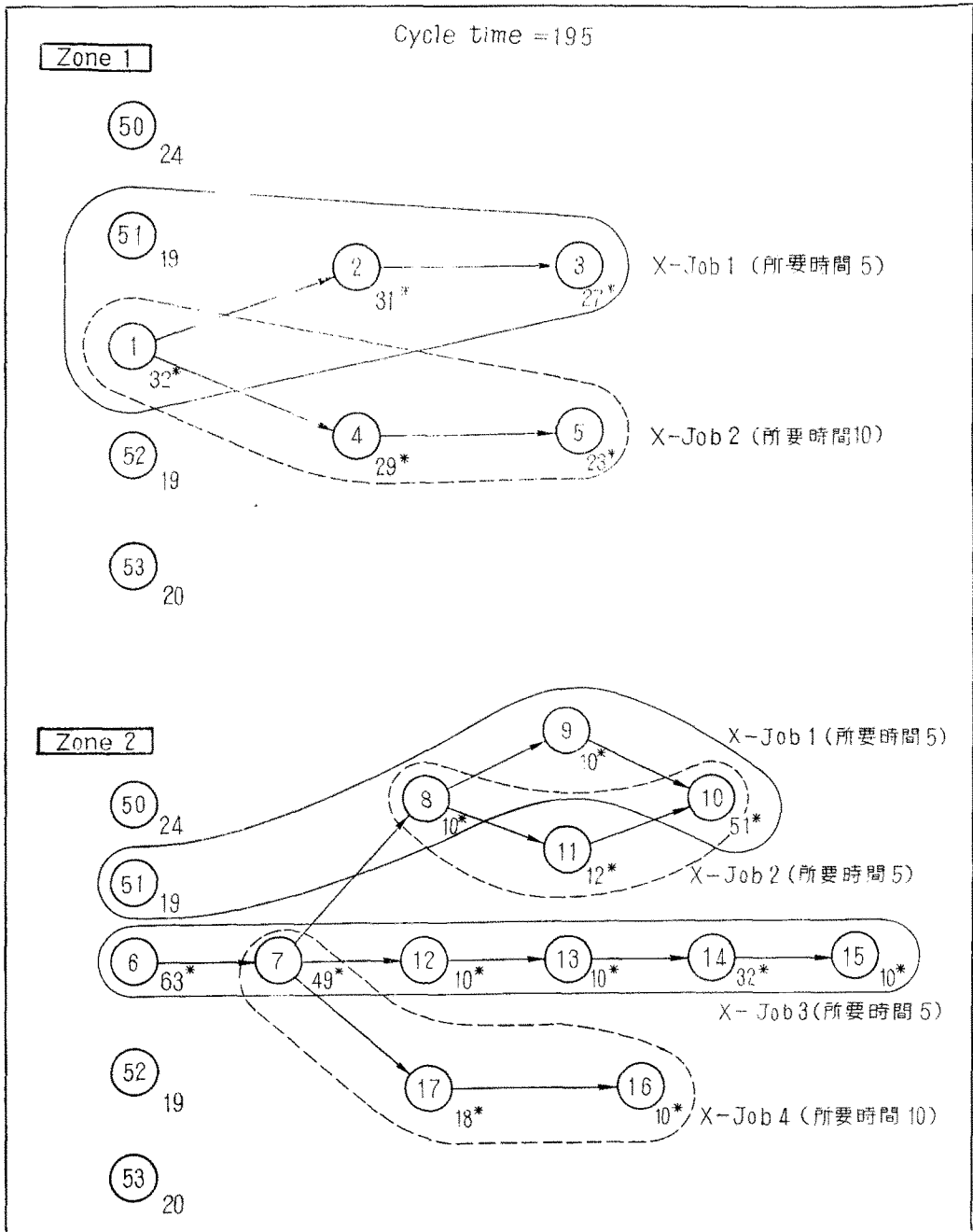
PRINCE/MI での主な改正点は、表4にあげた6つである。

5. 簡単な例

次に簡単な例を表示する。

第5がそのモデルである。丸の中側の数字が Job 番号で、外側にその time を示す。従って Job 50 は所要24時間であります。

表 5 例 題



次に矢印は先行関係を示し、Job 2 は Job 1 の後で行なうべきことを示しており、また米印のついているのが Must—do—Job であることを示している。

X—Job を示すために、曲線で囲んだ。Zone 1 では Job 1, 2, 3, 51 が同じ道具を使ってやる作業であることを示す。

この2つの Zone からなるラインの計算結果は表6の通りである。左から 作業者番号, Zone 番号, Job 番号, その所要時間, 各作業者別時間要計, ライン全体の時間累計を示している。

Cは、実質上の Can—do—Job を示す。

尚, Job No の左にその Job 内容を簡単に示すアルファベットを出すこともできる。表6の後に各人の割当状況をグラフにしたものも, out put される。

表6 計算結果

TEST MODEL		To=195			
OP.No.	Zone No.	Job No.	TIME.	SUM/MAN	TOTAL
1	1	1	32	47	47
1	1	2	31	78	78
1	1	3	27	105	105
1	1	4	29	134	134
1	1	5	23	157	157
1	1	50	24	181	181C
<hr/>					
2	2	52	19	19	200
2	2	6	63	87	268
2	2	7	49	146	327
2	2	17	18	164	345
2	2	8	10	184	365
2	2	16	10	194	375
<hr/>					
3	2	53	20	20	395
3	2	11	10	42	417
3	2	12	12	57	432
3	2	13	10	67	442
3	2	14	32	99	474
3	2	9	10	114	489
3	2	10	51	170	545
3	2	51	19	189	564
<hr/>					
4	2	51	50	55	619
<hr/>					
IDLE=171		IDLE PERCENT=21.92			

6. 最 後 に

実際のあるラインについて、PRINCE/MI の結果と、人間が試みた結果を比較したものが、次のグラフ1である。これをみても、かなり良い結果を得ていると思う。ラインバランスの方法は、これ以外にも多いが、現在までの論文はすべて先に書いた Jackson のモデルである。

これらが、ラインのバランスをとるのに有益なことはいうまでもないが、他にまだ多くの問題があることも事実である。ラインバランスそのものの問題の他に、ライン前後との関連、ラインへの材料補給の問題もあらためてクローズアップされてくるであろう。

ラインバラシングのプログラムも、これらの問題を総合した広い視野の中で使用されると、一層真価を発揮すると思われる。これらの要求から PRINCE/MI も今後幾度かの修正を余儀なくされるであろう。しかし又逆に PRINCE/MI がこれらの問題の究明への意欲を刺激するもの、あるいは手がかりになりつつあると思う。

グラフ1 実際のラインへの割当状況

