日程計画問題の"目の子式"解法

能　城　昌　子

(1969 年 11 月受付)

A Heuristic Solution of the Job-Shop-Scheduling Problem

Masako Noshiro

Some random trial methods have been proposed to the "job-shop-scheduling problem". In this note we consider a heuristic approach. The job, which is on the critical paths and has the maximum waiting time, or for which the machines wait for the longest time, is exchanged with another job if the exchange really improve the scheduling. Some numerical results, including the improved solutions of the previous authors, are shown.

The Institute of Statistical Mathematics

1. 日程計画問題

工場に m 台の異なる機械があり，1つの仕事を完成するには m 台のすべてを一定順序（それを機械の番号 $1, \cdots, m$ とする）で使用しなければならない．n 個の異なる仕事が与えられ，k 番目の仕事の j 番目の機械による処理時間 a_{kj} が定まっており，（確率的変動を考えない）既知であるとする．このとき，仕事全体の進行は n 個の仕事をどのような順序で処理するかに依存し，$n!$ 通りの可能な順列の関数である．このとき能率のよい順列を選び出す，というのが Job-Shop-Scheduling 問題のもっとも簡単なものである．

仕事の順列が定まっているとしよう．全体の作業の開始時点を0とし，k 番目の仕事の j 番目の機械による作業が終る時点を t_{kj} とすると，t_{kj} は a_{kj} より次のように定まる

$$
\begin{aligned}
t_{11} &= a_{11} \\
t_{1j} &= t_{1j-1} + a_{1j}, \qquad j = 2, \cdots, m \\
t_{k1} &= t_{k-11} + a_{k1}, \qquad k = 2, \cdots, n \\
t_{kj} &= \max(t_{k-1j}, t_{kj-1}) + a_{kj} \\
& \qquad\qquad\qquad \left\{ \begin{array}{l} j = 2, \cdots, m \\ k = 2, \cdots, n \end{array} \right.
\end{aligned}
$$

各時点で k 番目の仕事は j 番目の機械の前で

$$
L_{kj} = \left\{ \begin{array}{ll} t_{k-1j} - t_{kj-1} & t_{k-1j} > t_{kj-1} \\ 0 & t_{k-1j} \leqq t_{kj-1} \end{array} \right.
$$

という機械待ちの遊び時間をもち，逆に j 番目の機械は k 番目の仕事の到着まで

$$
l_{kj} = \left\{ \begin{array}{ll} t_{kj-1} - t_{k-1j} & t_{k-1j} < t_{kj-1} \\ 0 & t_{k-1j} \geqq t_{kj-1} \end{array} \right.
$$

という時間だけ遊ぶ．

計画の問題としては，機械待ち時間の総和を最小にする（サービス全体を良くする），仕事待ち時間の総和を最小にする（機械を賃借りするような場合に，その費用を減らす），という問題もあるが，ここでは仕事を完成するまでの総所要時間 t_{nm} を最小にすることを考える．

2. 問題の解法

機械が2台の場合の解法は簡単である．仕事が2つの場合も，やや面倒ではあるがアルゴリズムは確立されている．

一般の場合にはうまいアルゴリズムが見出だされておらず種々の試みがなされている．

Heller (1960) は機械の数 $m=10$, 仕事の数 $n=100$ の場合，まったく確率的に3000通りの仕事の順列を生成し，それぞれの総所要時間を求め，最小の順列を選んだ．

Page (1965) は Chained Monte Carlo 法というものを提案した．それは，仕事のある順列から出発し，これをK組に確率的に分割する．ただし K はあまり大きくない整数であり，順列にたいして K-1 個の分割点を確率的に選ぶのである．したがって K 個の各組の大きさは一様でない．このK組の内部の順列はそのままで小組を単位とした K！通りのすべての順列をつくり，その中で最適のものを選び出す．この，"確率的分割，K！通りの順列，最適の選出"，を適当回反復したならば，K を 1/2 に減らして再び同一のことを行なう．Page の実験では $m=10$, $n=20$, $K=16$, 8, 4, 2 と選んでいる．

この方法の意味は次のように解釈される．長さ n の順列が2つあるとき，その間の距離を次のように定義する．順列のあい隣る成分の対をつくる．2つの順列の間でこの n-1 個ずつの対のうちで一致しないものの数 (0, 1, ･･･, n-1) を順列間の距離とする．もしも順列を K 個の組に分け，組を単位とする順列だけをつくれば，これらの間の距離はたかだか K である．上の方式は順列を変化するのに，ある程度大きく動かして良い順列を定めたならば，次第にその近くでだけ探す，というのが趣旨である．

朝尾 (1967) は，仕事を同じ大きさの組に分け，各組内であらゆる順列をつくって最適順列を定め，各組の1位全部，2位全部，････を集めて新しい組をつくる，ということを反復した．これは，小さな組の中で先に処理した方がよいものは全体でも先に処理した方がよいだろう，という考慮であろう．

Page，朝尾は Heller のデータについて実験しているので，これらの結果を比較することができる．

表　1

Job　20 個の結果

	反復番号	step 数	最小値
Page	1	514	151
	2	368	151
	3	455	148
	4	532	147
	5	459	146
	6	643	150
Heller		3000	150
朝　尾	1	5	145
	2	11	147
	3	8	150

Job　100個の結果

朝　尾	1	5	542
	2	7	551
	3	9	543
	4	9	551
	5	8	550
	6	7	551
Heller	3000	平均値	656.81
		標準偏差	20.80

3. 新しいアルゴリズムの提案

こゝでの方法は，確率的に模索するよりは，できるだけこれまでの結果を"視察し"て改良の可能性の多い所を"手直し"していく，いわゆる"発見的な"，"目の子式"の方法によっている．

第1節での記号を用いる．もしも仕事を逆の順に並べ，機械を逆の順に使用し，時間も t_{nm} から始まって逆向きに進行させたとする．できるだけ“前に”つめて仕事を行なうとし，k' 番目の仕事の j' 番目の機械による作業，“開始時刻”を $T_{k'j'}$ とする

$$T_{nm}=t_{nm}$$

$$T_{n-k'm}=T_{n-k'+1m}-a_{n-k'+1m}$$

$$T_{nm-j'}=T_{nm-j'+1}-a_{nm-j'+1}$$

$$T_{n-k'm-j'}=\min\,(T_{n-k'm-j'+1}-a_{n-k'm-j'+1},\ T_{n-k'+1m-j'}-a_{n-k'+1m-j'})$$

$$k'=1,\cdots,n-1 \qquad j'=1,\cdots,m-1$$

すると，当然 $T_{n-k'm-j'}\geqq t_{n-k'm-j'}$ となるから，その差を

$$f_{n-k'm-j'}=T_{n-k'm-j'}-t_{n-k'm-j'}$$

表 2

	No.	Job	1		2		3		4		5	
	1	1	11	11	9	20	18	38	6	44	14	58
	2	2	17	28	10	38	15	53	6	59	19	78
a_{kj}, t_{kj} の表	3	3	10	38	16	54	3	57	16	75	20	98
	4	4	5	43	9	63	10	73	12	87	7	105
	5	5	11	54	17	80	7	87	19	106	16	122
	6	6	20	74	16	96	12	108	5	113	8	130
	1	1				−11		−20		−38		−44
L_{kj}, l_{kj} の表	2	2				−8				−9		−1
正が機械待ち；L_{kj}	3	3						−1	2		3	
負が仕事待ち；$-l_{kj}$	4	4			11			−6	2		11	
	5	5			9			−7				−1
	6	6			6			−9		−2	9	
	1	1	11	0	20	0	38	0	46	2	60	2
	2	2	28	0	38	0	53	0	59	0	79	1
T_{kj}, f_{kj} の表	3	3	38	0	54	0	59	2	75	0	99	1
	4	4	52	9	63	0	75	2	87	0	106	1
	5	5	63	9	80	0	87	0	106	0	122	0
Ko=1	6	6	89	15	105	9	117	9	122	9	130	0
	1	4	5	5	9	14	10	24	12	36	7	4
	2	1	11	16	9	25	18	43	6	49	14	6
	3	2	17	33	10	43	15	58	6	64	19	8
	4	3	10	43	16	59	3	62	16	80	20	10
	5	5	11	54	17	76	7	83	19	102	16	11
	6	6	20	74	16	92	12	104	5	109	8	12
	1	4				−5		−14		−24		−36
	2	1				−2		−1		−7		−6
	3	2				−8				−9		−1
	4	3						−1	2		3	
	5	5			5			−14		−3	1	
	6	6			2			−9		−2	10	
	1	4	5	0	15	1	25	1	43	7	50	7
	2	1	16	0	25	0	43	0	50	1	64	1
	3	2	33	0	43	0	58	0	64	0	83	0
	4	3	44	1	60	1	67	5	83	3	103	0
	5	5	60	6	77	1	84	1	103	1	119	0
	6	6	86	12	102	10	114	10	119	10	127	0

Ko'=4　Ko=3　Ko=2

と定義する．

$f_{kj}=0$ となる (k, j) の集合が，いわば t_{nm} の値を決定している“きわどい”点であり，その他の点は，仕事の開始時間を多少変えても t_{nm} に影響のないような点である．

第2表は計算機の出力そのものである．“きわどい”点は左上から右下へ，いわゆる臨界経路をなしている．（臨界経路は必ずしも一筋ではなく途中で分岐することもある），この上の点で，遊び時間の長い点を見出だし，その仕事に注目して仕事の順序の入れ替えを試行錯誤するのである．

4. 実行したアルゴリズム

計算機で実行する手順は次の通りである．ある与えられた仕事の順列について

I　a_{kj} の表より t_{kj}, l_{kj}, L_{kj}, T_{kj}, f_{kj} の表を計算する．

II　1)　$f_{kj}=0$ となるすべての点の上で，l_{kj} を最大にするような仕事 k_0 を選び出す．

2)　仕事 k_0 を選び出したならば

a)　k_0+1, k_0+2, $\cdots$, n 番目の仕事と k_0 番目の仕事を互換し，その各々について総所要時間を計算する．総所要時間最小の互換により改善が行なわれるならば，互換の結果を新しい仕事の順列としてIへ，改善がなければbへ．

b)　同様に $1, 2, \cdots, k_0-1$ 番目の仕事と k_0 番目の仕事との互換を行なう，総所要時間最小の互換により改善できればIへ，さもなければCへ．

c)　k_0+1, k_0+2, $\cdots$, n 番目の仕事を，k_0 番目の仕事の直前に移す．これらの並べ換えについての最小総所要時間内で改善が行なわれればIへ，さもなければIIIへ．（または停止する）．

III　1)　$f_{kj}=0$ なる点の上で L_{kj} が最大の k を見出だす．このような k を k_0' とする．

2)　k_0' 番目の仕事について

a)　これと，$k_0'+1, \cdots, n$ 番目の仕事とを交換する，

b)　これと，$1, \cdots, k_0'-1$ 番目の仕事とを交換する．

c)　k_0' 番目の仕事の直後に，$1, \cdots, k_0'$-1，番目の仕事を移す．

IIIにおいても最小総所要時間により改善されればIへ，そうでなければ次へ移る．

なお，次のようなプログラムの変種を用意した。

プログラムA　常にII, IIIの順に仕事の並べ換えを試みる．

プログラムB　II, III, の順の適用と，III, IIの順の適用とを交代する．

プログラムC　$f_{kj}=0$ の点より l_{kj} 最大の r 個の仕事，L_{kj} 最大の r 個の仕事を，重複があればさらに増して選ぶ．これら $2r$ 個の仕事を l_{kj}, L_{kj} の大きさの順に並べて順次 k_0（あるいは k_0'）として選び出し，IIa, IIb（IIIa, IIIbと同じ），IIc, IIIcの並べ換えを試みた．

5. 計 算 結 果

Job 1 から Job 20 を使用して，全くラ　ダムに仕事の順序をきめて7回 Programm A, B, C について計算した．

139 を得た順序

1 16 20 13 8　　2 15 14 19 3　　9 17 4 7 12　　6 10 11 18 5

140 を得た順序

R 1 (A)	1 14 9 5 6	4 16 15 17 2	13 8 12 3 18	20 19 11 10 7
R 3 (B)	13 1 14 9 2	8 3 16 20 6	18 19 15 17 10	4 12 7 11 5

R 1 (A)，R3 (B) に2つのパターンの一致がみられる．Programm A, B, C を比較すると，最

表　3

反復番号	Programm A		Programm B		Programm C	
	最小総所要時間	step 数	最小総所要時間	step 数	最小総所要時間	step 数
R 1	* 140	9	145	11	141	14
R 2	146	6	152	5	* 142	9
R 3	144	9	* 140	13	144	9
R 4	145	11	151	4	* 142	12
R 5	* 145	6	* 145	12	146	11
R 6	153	3	* 141	10	146	8
R 7	147	11	* 143	13	144	10
朝尾 best の順から開始	141	3	141	4	* 139	5

演算所要時間　平均50分 (HIPAC−103使用)　　統計数理研究所

* 印：ＡＢＣの中で最小総所要時間が最小のもの

小総所要時間が最小になったのは 8 回の試行うち A 1.5 回，B 3.5 回，C 3 回で，B, C はあまり差はみられない．8 回の最小総所要時間を平均したところでは A は 145.125, B は 144.75, C は 143 と C の方がよいといえる．

Job 1～Job 100 については

朝尾　best の順序すなわち 542 を得た順序から始めて，18 step 目に最小総所要時間 518 を得た．全くランダムな順序から出発した場合には

反復番号	最小総所要時間	step 数	演算所要時間 (IBM-360 使用)
1	525	30	2 分 41 秒
2	523	33	3 分 04 秒
3	534	22	1 分 51 秒

を得た．

朝尾　最短時間 (542) を与えた仕事の順序

22 46 66 95 8	43 74 87 13 29	51 99 20 34 57	77 2 40 65 82
1 37 60 90 25	48 68 97 6 30	71 79 14 32 54	81 19 45 61 91
16 49 72 98 15	41 62 83 21 26	69 86 3 33 55	92 7 38 56 80
4 42 52 88 17	50 70 100 23 27	73 76 9 35 58	84 11 39 63 93
24 36 59 85 10	28 53 96 12 31	64 78 18 44 67	89 5 47 75 94

518 を得た順序

1 74 13 49 46	66 95 51 8 82	29 87 99 20 57	77 2 40 65 43
22 37 60 90 25	48 68 97 6 30	71 79 14 32 54	81 19 45 35 91
16 72 98 15 41	62 83 21 26 69	86 3 33 55 92	7 38 56 94 4
42 52 88 17 44	70 100 23 27 73	76 9 61 58 84	11 39 63 93 24
36 59 85 10 28	53 50 96 12 31	75 18 78 64 67	89 80 5 34 47

ランダムな順に出発して 523 を得た順序

1 66 22 70 74	82 63 99 86 46	50 64 10 34 39	91 53 36 32 8
73 6 15 96 77	68 92 48 21 59	88 25 90 26 80	43 71 69 24 44
13 28 17 19 95	61 94 38 79 20	83 40 14 42 81	23 52 4 97 89
45 98 16 27 3	35 62 54 100 18	51 30 29 55 37	33 9 85 12 58
49 11 56 75 57	76 60 72 87 67	41 65 84 47 78	93 31 2 7 5

この方法は図 1 にもみられるように改善の巾がある程度限られる為に，全くランダムな順から始めるより Page の方法または朝尾の方法なりを組み合せた方がより効果的ではないかと思われる．

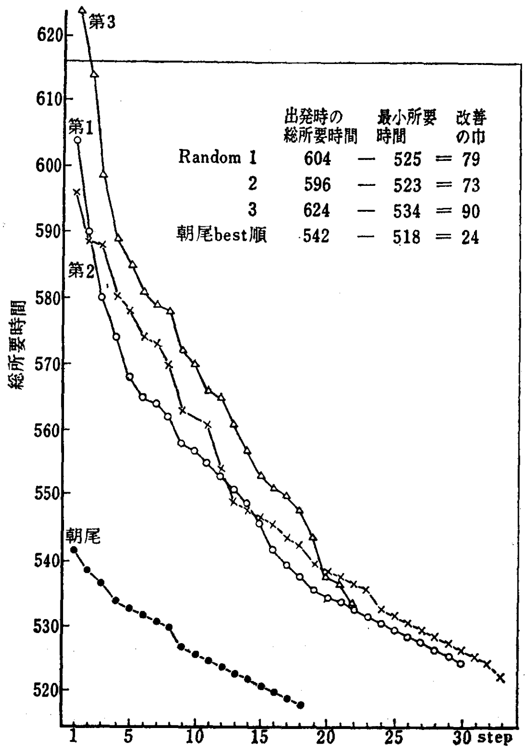

図 1

6. 謝 辞

このアルゴリズムは渡辺浩氏の御示唆によるものである．計算には統計数理研究所計算機室および三菱原子力工業株式会社電子計算部調査開発課の反町洋一氏，永島美奈子さんの皆様のお世話になった．論文の執筆は，渋谷政昭氏に指導していただいた．心からお礼を申し上げる次第である．

統計数理研究所

文 献

1) Heller, J.: Some Numerical Experiments for an M x J Flow Shop and Its Decision-Theoretical Aspects. J. O. R. S. A. 8. (1960), 178∼184.
2) Page, E. S.: On monte Carlo methods in Congestion Problems I. Searching for an Optimum in Discrete Situations. J. O. R. S. A. 13, 2 (1965) 291∼299.
3) 朝尾 正: 共用のある設備における生産順序の決定について．日本科学技術連盟 (1967)