≪綜合報告≫

流れ作業における Dynamic Programming の応用

鍋島一郎*

1. はしがき

組立流れ作業調整 (Assembly line balancing) の問題は永い間試行錯誤によって多くの時間が費されていたが，ここ数年来この問題が流行し，多くの解析的な論文が発表された。以下，Kilbridge と Wester〔1〕等に従ってその展望と簡単な解説を述べ，次に，Held, Karp, Shareshian〔2〕による動的計画法を用いた解析について報告する。

2. 解析的方法の展望

最初の論文は Salveson〔3〕による。彼は L.P. モデルで仕事場のあらゆる組合せを考えたが多大の計算を必要とし実用的でないので，整数の問題に直して組合せ論的解析に帰したが，すべての可能な仕事場への配分の計算がぼう大なものとなり実用的でない。それより試行錯誤法の方がより少い労力でよい結果に達する。しかし彼はこの問題を工学的に位置づけることにおいて充分な説明を与えた。

1956年に Jackson〔4〕が最適解に達する原理を発表した。その原理は第一の仕事場への仕事の実行可能な配分を数え上げ，その各々の配分に対して，第二の仕事場への仕事の実行可能な配分を数え上げる。これを実行可能な配分の表が完成するまで続ける方法である。これらの配分から，サイクルタイムより大きくない配分を選ぶ規則が与えられており，最終的には，このような配分の集合から仕事場の数を最小ならしめる配分が選ばれる。Jackson は彼の方法を D.P. を用い数学的に正しいことを示したが，実用には向かない。Bowman〔5〕は L.P. による解を与えたが計算量が多く実用的でない。

以上の方法は簡単な理論的な場合には非常にうまく行くが，現実のラインに応用されると手に負えない計算量を生ずる。それらのあるものは計算機のプログラムを作っているが，適当な時期に収束を確かめるための制約を導入しなければならない欠点がある。

ごく最近 Tonge〔6〕，及び Kilbridge と Wester〔7〕によって“発見的” (heuristic) と称する体系が発表された。それらは上に述べた方法より融通がきく。

先ず Tonge の発見的手順は次の三段階より成る。第一段階は優先図表（第1図参照）で隣接せる基本仕事を合成仕事に群分けすることによって初期の問題を次々に簡単化してゆく。

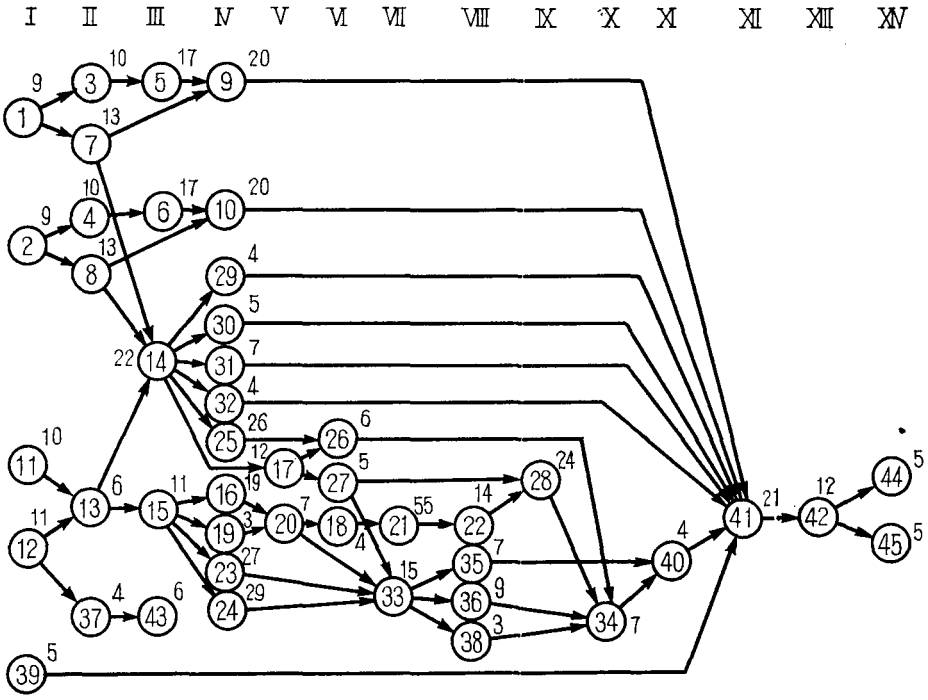
順序の制約に基いて，“集合” (set)，“鎖” (chain)，及び“Z”という三種類の合成仕事が定

* 幾徳工業高専　1964年1月10日受理　「経営科学」 第7巻4号

義される。

図2は“集合”と“鎖”を示す。基本仕事 U_{10}, U_{11}, U_{12}，の間に相対的順序関係はない。それらは同一の先行仕事と後行仕事とをもつ。それらをまとめて V_2 という“集合”で置き換えられる。合成仕事 V_2 と単一の仕事 U_{15} とはそれらの相対的順序が完全に決っているので一つの“鎖” V_3 と考えられる。

第3図は“Z”を示す。それは4つの基本仕事の群で，前の2つの仕事は同一の先行仕事をもち，後の2つの仕事は同一の後行仕事をもっている。



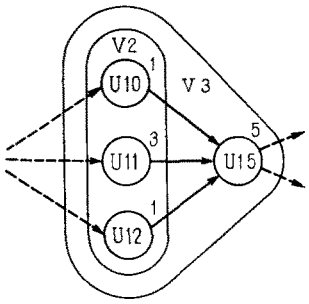
第1図

註：第1図の優先図表で，円内の数は基本仕事の番号を示し，円外の数は対応する加工時間（time durations）を示す。矢は優先関係を表わす。例えば基本仕事1は3と7に先んじなければならない。

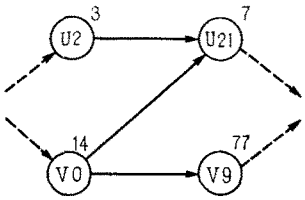
従って第一段階ではこれら合成仕事に群分けし，単一の仕事の代りにそれらを考察するので問題の複雑さが減少する。第二段階では，絶対的に必要な場合にのみ合成仕事を再分解するが，これらの合成仕事を仕事場に配分する。もし仕事の群分けをし直す必要のある時は一定の手順に従って行なう。第三段階では，仕事の分布が出来るだけ一様になるまで，仕事場の間に仕事を移動させることによって，調整を保つ。

かかる Tange の体系は多くの場合に用いられて，多大の効果と良い結果を与える。

一方，Kilbridge と Wester の発見的体系は，調整の道具として優先図表を更に重視する。図表において，列内で順序を変え得ることと，横への移動可能性という2つの性質が最適な調整を達



第 2 図



第 3 図

成する為に利用される。

位置及び固定用具の制約に基き，基本仕事の全分布を群及び部分分布に分離する。一般に固定用具は群を決定し，位置の制約は部分分布を決定する。そこでこれらの集合の各々は，互に独立にではないが，別々に調整される。

第 1 表　図 1 における基本仕事の配分（サイクルタイム0.69で 8 つの仕事場）

基本仕事の番号 i	基本仕事の加工時間 t_i	加工時間の和（分）	累積加工時間（分）	基本仕事の番号 i	基本仕事の加工時間 t_i	加工時間の和（分）	累積加工時間（分）
1	0.09			29	0.04		
2	0.09		仕事場 1	31	0.07		仕事場 5
11	0.10			32	0.04		
12	0.11			25	0.26		
7	0.13			17	0.12		
8	0.13			20	0.07		
37	0.04	0.69	0.69	27	0.05		
				18	0.04	0.69	3.45
39	0.05			21	0.55		仕事場 6
3	0.10		仕事場 2	22	0.14	0.69	4.14
4	0.10			10	0.20		
13	0.06			33	0.15		仕事場 7
14	0.22			35	0.07		
15	0.11			38	0.03		
30	0.05	0.69	1.38	28	0.24	0.69	4.83
5	0.17			26	0.06		
6	0.17		仕事場 3	36	0.09		
43	0.06			34	0.07		仕事場 8
24	0.29	0.69	2.07	40	0.04		
16	0.19			41	0.21		
19	0.03		仕事場 4	42	0.12		
23	0.27			44	0.05		
9	0.20	0.69	2.76	45	0.05	0.69	5.52

第 1 表は第 1 図に対する最適な配分を示している。

すべての基本仕事の加工時間の和は5.52であるから，完全な調整は，5.52(自明)，2.76, 1.84,

1.38, 0.92, 及び 0.69 分のサイクルタイムに対して達せられ得る。流れ作業上の仕事場の数はそれぞれ1，2，3，4，6，及び8である。これらのサイクルタイムに対して完全な調整が可能であるかどうかであるが，表1はサイクルタイム0.69分で8つの仕事場の場合に対しては可能であることを示す。

上にあげた6つの場合以外のサイクルタイムに対しては完全な調整は達成されない。その結果起る調整の遅れを見出すために，仕事場の数 n は，これら n 個の仕事場に基本仕事をまとめるに必要な最小量だけ $\sum_i t_i/c$ (c はサイクルタイム)を越える一つの整数として選ばれる。もし $\sum_i t_i/c$ より大きい最小整数 n に対して調整が達成されたならば，それの最適性は保証される。さもなければ，調整を達成するに必要な最小数の附加の仕事場だけ n を増加しなければならない。

これらの発見的体系は電子工学や自動工場における種々の組立流れ作業を調整するのに有用であることが実証されている。一般には最適解は唯一でなく，最適調整を損うことなく管理者が基本仕事の列を変える自由をもつことを許している。

1962年に Held と Karp〔8〕が順序づけ問題への動的計画法による接近において流れ作業調整問題を D.P. によって定式化し，それの計算手順を示し計算機による例を与えているが，1963年に Held, Karp 及び Shareshian〔2〕は前論文で簡単に論じられた内容を精密化し，動的計画法による接近を詳細に与えている。尚それ以前に1963年に Klein〔9〕が仕事の順序が一定である簡単な場合に，すべての可能なサイクルタイムに対し全遊休時間（サイクルタイムと各仕事場における実際の加工時間の差の総和）を最小ならしめるように仕事場へ仕事を配分する問題を解き，それを各実行可能列に応用しているがこれも計算量がぼう大となる。

3. 動的計画法による接近

以下 Held, Karp, Shareshian〔2〕による方法を詳述する。

優先関係のある作業の順序づけ問題として組立流れ作業調整の問題を考察し，M. Held と R. M. Karp による論文〔8〕で与えられた議論を拡大することによって，流れ作業調整の問題とその解を吟味する。解の方法は次の二段階で与えられる。すなわち，

(1) 小規模の問題の正確な解を与える D.P. の技法，

(2) 大規模の問題の近似解に対する繰返し手順

である。それらの計算例も与える。

最後の項で，優先条件をもった順序づけ問題への D.P. の方法の応用に必要な，半順序集合についてのある組合せ論的問題の形成とその解とを考察する。

最後に，この論文における方法と Jackson〔4〕によって提案された D.P. による形式化とが比較される。

問題

組立流れ作業は一単位の生産に必要な基本的仕事の部分集合が課せられる所の仕事場 (work station) の列と考えられる。組立流れ作業調整問題は，仕事場の数が或る条件に関して最小になるようにそれらの仕事を仕事場に配分する問題である。このような条件は基本的仕事の順序と，ラインの組織及び生産率に関連する。

ここでは原則的にはかかる問題の簡単化された型を取り扱う。すなわち，組立流れ作業は次の条件を満足しなければならないと仮定する。

(1) 直列ライン (Seral Line) 各単位は一定の順序で仕事場に課せられ，どんな2つの場も同時に同じ単位を作業することはない。かくして，全組立ラインは，支線も並行せる部分組立ラインも持たず，直列であると考える。

(2) 優先関係 (Precedence Relations) 仕事群の履行の順序に関する制約は，“仕事 i が仕事 j に優先する”（これを記号 $i<j$ で表わす）という優先関係によって表示する。

(3) サイクルタイム (Cycle Time)，連続せる単位の完成間の期間は一定 T である。これをサイクルタイムと呼ぶ。従って，各仕事場はそれに配分された仕事を単位当り T 時間以内に完成しなければならない。

(4) 加工時間と仕事場 (Excution Times and Work Stations) 各の基本的仕事は各の単位に対して丁度一度だけ遂行される。一つの仕事の加工時間は一定で既知であり，それがかけられる仕事場や前後の仕事に独立である。特に，優先関係さえ満たされる限り，どんな仕事も任意の仕事場に配分され得ると仮定する。一つの仕事場に配分される仕事の集合は各の単位に対して同一であり，その集合の加工時間は個々の仕事の加工時間の和であって，サイクルタイムを越えてはならない。従ってこのモデルはすべての仕事場を交換可能とみなし，生産されるすべての単位を同一のものと見なす。そして準備時間 (set-up times) を考えない。

以上のモデルで，与えられたサイクルタイム$^{+}$の下に仕事場の数を最小にする問題を取り扱う。

この問題は最初 Salveson〔3〕によって提案され，整数線型計画法に基く解法が Salveson〔3〕及び Bowman〔5〕によって既に与えられた。しかしながら，このような方法はひかえ目な問題に対してさえもぼう大な計算量を要求するので実用的ではない。発見的接近が Kilbridge と Wester〔7〕，Tonge〔6〕，Helgeson と Birnie〔10〕及び Burgeson と Daum〔11〕によって暗示され た。Jackson〔4〕は動的計画法による技法を与えている。

（＋：この問題の解は，仕事場の数が与えられたとき，サイクルタイムを最小にする問題を解くのに，繰り返して利用される。）

動的計画法による解（文献〔8〕における原理の精密化）

組立流れ作業調整問題は配分されるべき仕事 $J_1, J_2, \cdots\cdots, J_n$ の集まりと，それらの加工時間 $t_1, t_2, \cdots\cdots, t_n$：サイクルタイム T：および仕事間の優先関係を表わす半順序によって特徴づけられる。

半順序に関連する次の概念を導入する。

仕事の部分集合 $S=\{J_{i1}, J_{i2},\cdots\cdots, J_{in(S)}\}$ は，$J_j \in s$ で $J_i<J_j$ であるとき $J_i \in s$ となるとき実行可能 (feasible) と呼ぶ。

従って実行可能部分集合は何か他の仕事を前もって加工することなしにある順序で加工されうる仕事群である。

一つの列（順序集合）$\sigma=(J_{i1}, J_{i2},\cdots\cdots, J_{in(\sigma)})$ は，$1\leqq q\leqq n(\sigma)$ に対して $\{J_{i1}, J_{i2},\cdots\cdots, J_{iq}\}$ が実行可能集合であるとき実行可能と云われる。従って実行可能列は何か他の仕事を前もって加工することなしに，それに示された順序で加工される列である。

$\sigma=(J_{i1}, J_{i2},\cdots\cdots, J_{in(\sigma)})$ が実行可能列ならば $F(\sigma)=\{J_{i1}, J_{i2},\cdots\cdots, J_{in(\sigma)}\}$ は実行可能集合であり，逆に S が実行可能集合ならば $F^{-1}(S)=\{\sigma|F(\sigma)=S\}$ が実行可能列であるという2つの写像によって実行可能列と実行可能集合との間に本質的な対応が存在する。

実行可能列 σ の仕事を示された順序で加工するのに必要な仕事場の数を最小ならしめるような仕事の配分を σ に対する誘導配分 (induced assignment) と云う。この配分は次のようにして得られる：各仕事場に配分される仕事の加工時間の和がサイクルタイム T を越えないように，列 σ の始めから出来るだけ多くの仕事を第一の仕事場に配分し，残りの列の始めから出来るだけ多くの仕事を第二の仕事場に配分する。以下同様につづけてゆく。

σ に対する誘導配分が r 個の仕事場を必要とし，$t^{(r)}$ が r 番目の仕事場に配分される仕事の加工時間の和であるとすると，量 $c_\sigma=(r-1)T+t^{(r)}$ は列 σ を加工するときの“費用” (cost) の一つの測度となっている。

もし一つの実行可能列 σ^* が σ の終りに一つの仕事 J_l を加えることによって作られるとすれば，

$$c_{\sigma^*}=c_\sigma+\Delta(c_\sigma, t_l)$$

である。

ここに Δ は，(i) $[x/T]=[(x+y)/T]$

または $[(x+y)/T]=(x+y)/T$ ならば，$\Delta(x, y)=y$,

(ii), $[x/T]<[(x+y)/T]$ で $[(x+y)/T]<(x+y)/T$ ならば，$\Delta(x, y)=T[(x+y)/T]+y-x$;

である。($[x]$ は x の整数部分を表わす。)

従って，σ に対する誘導配分における最後の仕事場が J_l を受け入れるに十分な“ひまな時間” (idle time) を持つ時は $\Delta=t_l$ で，そうでない時は，未使用のひまな時間は Δ の計算において t_l に加えられなければならない。

任意の実行可能集合 S に対して一つの費用

$$C(S)=\operatorname*{Min}_{\sigma\in F^{-1}(S)} c_\sigma$$

が対応する。

特に，もし $C(\{J_1, J_2,\cdots\cdots, J_n\})=(\bar{r}-1)T+t^{(\bar{r})}$,

$0<t^{(\bar{r})}\leqq T$ ならば，全ラインに対して必要な仕事場の最小数は $\bar{r}$ である。

上の定義より，最適性原理から次の繰返し関係の成り立つことが示される。

(a) $[n(S)=1]: c(\{J_t\})=t_t$

(b) $[n(S)>1]: C(S)=\underset{\substack{J_t\in S \\ S-J_t \text{ feasible}}}{\text{Min}}[C(S-J_t)+\Delta[c(S-J_t),\ t_t]]$;

ここに $S-J_t$ は S から J_t を除くことによって得られる集合である。

これらの繰返し関係によって，実行可能列よりもはるかに少ない実行可能集合のみを含む計算によって $C(\{J_1, J_2, \cdots\cdots, J_n\})$ を決定することができる。$C_\sigma=C(\{J_1, J_2, \cdots\cdots, J_n\})$ のとき，$\sigma=(J_{i1}, J_{i2}, \cdots\cdots, J_{in})$ に対する誘導配分は最適であると呼ぶ。この配分は仕事場の数を最小にするから，流れ作業調整問題の一つの解を与える。

σ に対する誘導配分が最適であるための必要十分条件は次式(1)の成り立つことである。

：すなわち

$$C(\{J_{i1}, J_{i2}, \cdots\cdots, J_{ip}\})=C(\{J_{i1}, J_{i2}, \cdots\cdots, J_{iP-1}\})+\Delta(C(\{J_{i1}, J_{i2}, \cdots\cdots, J_{iP-1}\})t_{iP}). \quad \cdots\cdots\cdots(1)$$

$(2\leqq p\leqq n)$.

従って，すべての実行可能集合 S に対して $C(S)$ の表が得られたならば，最適な誘導配分をもつ列 $\sigma=(J_{i1}, J_{i2}, \cdots\cdots, J_{in})$ が(1)から $J_{in}, J_{in-1}, J_{in-2}, \cdots\cdots, J_{i1}$ の順に求まる。

逐次近似 (Successive Approximations)

前項における D.P. 原理は取り尽くし法に対して顕著な改良を与え，限られた大きさの組立流れ作業調整問題を現存する計算機にかけて正確な解を求めることを可能ならしめる。IBM 7090 に対する経験的プログラムは 13,000 個の実行可能集合をもった 36 個の仕事の問題を略20秒で解く。問題の大きさが増すに従って，その原理の直接の応用は多量の記憶装置と計算時間を必要とする。従って，大きな問題を解くにはその原理が何か他の手順によって補われることを必要としている。

かくして選ばれた手順は，他の順序づけ問題〔8〕を近似的に解くのに効果的に用いられた技法に幾らか従っている逐次近似という手順であった。それは，

(i) 先ず最初に仕事の集合は一つの実行可能列 $\sigma^{(0)}$ に配列される。

(ii) 計算の i 番目の段階において，実行可能列 $\sigma^{(i-1)}$ は

$C_{\sigma(i)}\leqq C_{\sigma(i-1)}$ なる列 $\sigma^{(i)}$ によって置き換えられる。

これは次項に述べる一般的組立流れ作業調整問題の構造をもつ所の一つの誘導問題 (a derived problem) を解くことによって達成され，修正された D.P. 原理の応用によって解くことができる。

この逐次近似手順はしばしば一つの最適解を与える一つの実行可能列を生み出す。どんな場合も経験上非常に良い近似解が得られている。

以下，この逐次近似を詳しく述べる。一般的組立流れ作業調整問題を解くための D.P. 原理を次項で与え，その次に，かかる一般的問題の系列の解に基く逐次近似手順の構成を論じ，最後に

その計算結果を与える。

一般的流れ作業調整問題 (Generalized Line-Balancing problem)

一般的流れ作業調整問題とは，順序づけられるべき要素が仕事の列であるような問題である。

これらの仕事列 $\sigma_1, \sigma_2, \cdots\cdots, \sigma_u$ は，ある特定の組立流れ作業調整問題で配分されるべきすべての仕事を含む一つの実行可能列

$\sigma=(J_{i1}, J_{i2}, \cdots\cdots, J_{in})$ を分解することによって得られる。

この分解は次の形をとる：

$$\sigma=(J_{i1}, \cdots\cdots, J_{in})=(J_{i1}, \cdots\cdots, J_{in1})(J_{in_1+1}, \cdots\cdots, J_{in_2})$$
$$\cdots\cdots(J_{in_{u-1}+1}, \cdots\cdots, J_{in})=\sigma_1\sigma_2\cdots\cdots\sigma_u \qquad \cdots\cdots\cdots\cdots(2)$$

仕事の半順序は σ_i の半順序を引きおこす：すなわち，$J_{ik}<J_{il}$ なる如き仕事 $J_{ik}\in\sigma_f$, $J_{il}\in\sigma_g$ が存在するならば $\sigma_f<\sigma_g$ である。加えて，$\sigma_1<\sigma_j<\sigma_u(j=2, 3, \cdots\cdots u-1)$ という人工的制約を課するのが便利であることがわかる。

かかる誘導半順序と両立する所の σ_j のどんな順序も新らしい実行可能列を与える。

実行可能列のかかる組の中で $C_{\sigma'}$ が最小 ($C_{\sigma'}\leqq C_\sigma$) である如き

$\sigma'=(\sigma_1, \sigma_{j2}, \cdots\cdots, \sigma_{jn-1}, \sigma_n)$ を求めよう。

その順序は前に与えたと類似の D.P. の計算によって見出される。この計算は $l-1$ 個の列の一連 $\theta=\sigma_{j1}, \sigma_{j2}, \cdots\cdots, \sigma_{jl-1}$ に一つの列 $\xi=\sigma_{jl}$ が加えられたとき引き起される増加費用 $\delta(C_\theta, \xi)=C_{\theta\xi}-C_\theta$ の概念に依存する。

$\xi=(\cdots\cdots, J_{iq+p-1}, J_{iq+p})$ とすると，列 $\theta\xi=\sigma_{j1}\sigma_{j2}\cdots\cdots\sigma_{jl}$

によって引きおこされる配分において，ξ からの仕事を含む最初の仕事場は仕事 $J_{iq}, J_{iq+1}, \cdots\cdots, J_{iq+k_1}$ を含むであろう。：ここに

$$k_1=\max\{k\,|\,k\leqq p, \sum_{j=0}^{k} t_{iq+j}\leqq T\}$$

である。そのとき，$\delta(C_\theta, \xi)$ は次のようにして求まる。

(a) もし $k_1=p$ で $J_{iq}\in\theta$ ならば，$\delta(C_\theta, \xi)=C_\xi$;

(b) もし $k_1=p$ で $J_{iq}\in\xi$ ならば，

$$\delta(C_\theta, \xi)=T([C_\theta/T]+r)-C_\theta+C_\xi$$

で，$r=\begin{cases}1 & ([C_\theta/T]<C_\theta/T) \text{ のとき}\\ 0 & (\text{上記でないとき})\end{cases}$

(c) もし $k_1<p$ ならば

$$\delta(C_\theta, \xi)=T([C_\theta/T]+1)-C_\theta+C_\beta,$$
$$\beta=(J_{iq+k_1+1}, J_{iq+k_1+2}, \cdots\cdots, J_{iq+p})$$

である。そこで列 σ_i の最適な再順序づけに対して D.P. の原理を特性化することが可能で，この原理は組立流れ作業調整問題の正確な解に対して前に得られた原理との形式的類似を与える。以下これについて述べる。

部分集合 $S\subseteq\{\sigma_1, \sigma_2, \cdots\cdots, \sigma_u\}$ は，もし $\sigma_g\in S$ で $\sigma_f<\sigma_g$ のとき $\sigma_f\in S$ なるとき実行可能であると云う。

S が実行可能であるとき，$S=\{\sigma_{j_1}, \sigma_{j_2}, \cdots\cdots, \sigma_{j_r}\}$ なるすべての実行可能列 $\sigma_{i1}, \sigma_{j_2}, \cdots\cdots, \sigma_{j_r}$ に対して $C_{\sigma_{j_1}\sigma_{j_2}\cdots\cdots\sigma_{j_r}}$ の最小値を $C(S)$ とする。

そのとき，すべての実行可能集合 S に対して次の繰返し関係の成り立つことが示される。

(a) $C=(\{\sigma_i\})=C\sigma_i$,

(b) $$C(S)=\operatorname*{Min}_{\substack{\sigma_i\in S\\ S-\sigma_i\text{feasibe}}}\{C(S-\sigma_i)+\delta[C(S-\sigma_i), \sigma_i]\}$$

ここに $S-\sigma_i$ は S より σ_i を除くことによって得られる集合である。

σ_i の最適順序を計算するには，前に述べたと類似の計算が用いられる。

逐次近似手順の構成

逐次近似法は，前項で導入した一般的流れ作業調整問題の型をしている誘導問題の列を解くことによって進められる。方程式(2)の形の任意の分解によって誘導問題を選ぶと，再配分や費用C_σの改善を許さない程列 σ_i の間に強い優先関係が存在することが起って都合が悪い，特に σ_i と σ_j とが共にいくつかの仕事を含むとき，それらの間の優先関係の存在がありうる。従って，一つの実行可能列の分解は，大多数の仕事が1つまたは2つの列に配分され，他の列の各々には少数の仕事のみが含まれているように選ばねばならい。そこで，次の2種類の分解が用いられる：すなわち，

第一の型	第二の型
$\sigma_1=(J_{i1}, J_{i2}, \cdots\cdots, J_{is})$,	$\sigma_1=(J_{i1}, J_{i2}, \cdots\cdots, J_{is})$,
$\sigma_2=(J_{is+1})$,	$\sigma_2=(J_{is+1}, J_{is+2})$,
$\sigma_3=(J_{is+2})$,	$\sigma_3=(J_{is+3})$,
……	$\sigma_4=(J_{is+4}, J_{is+5})$,
$\sigma_{u-1}=(J_{is+u-2})$,	$\sigma_5=(J_{is+6})$,
$\sigma_u=(J_{is+u-1}, \cdots\cdots, J_{in})$.	……
	$\sigma_u=(J_{is+q}, J_{is+q+1}, \cdots\cdots, J_{in})$.

である。

逐次近似手順は次の4つの局面より成る。第1及び第3の局面は第一の型の分解を用い，第2と第4の局面は第二の型の分解を用いる。可能なすべての第一及び第二型の誘導問題を取り扱うことは不必要であるし時間の浪費であるから，選ばれる誘導問題はさらに三通りの方法で制限される。先ず第一に，各々の誘導問題において u の値は，実行可能集合 S の数の記憶の制限と両立する限り，できるだけ大きく選ばれる。この制約の下に，各局面における σ_1 の構成はその誘導問題を決定する。さらに問題の数を制限するためには，各局面において σ_1 の次のような逐次構成によって定まる問題のみを取り扱う：

$\sigma_1=(\phi)$〔空の列〕, $\sigma_1=(J_{i_1}, J_{i_2}, \cdots\cdots, J_{ia})$,

$\sigma_1=(J_{i_1}, J_{i_2}, \cdots\cdots, J_{i2a})$, ……

(a は演算のパラメーターであって，a の値は各局面において考察される誘導問題の数を決定する。a の大きすぎる値は貧弱な計算結果に導き，少さすぎる値は不必要に長い計算に導く，経験上，$a=9$ なる値が多くの問題に対して妥当な値である。)

このようにして，誘導問題はすべての順序の上を蛙飛びするように選ばれる。

第一及び第三の局面において，全体の順序における直接の費用 C の改善を生じ得ないような誘導問題を避けるために選択判定原理が用いられる。それは，列 $\sigma_1\sigma_2\cdots\cdots\sigma_u$ が与えられると，列 $\sigma_2\sigma_3\cdots\cdots\sigma_{u-1}$ の順序変更は $\sigma_1\rho\sigma_u$ の形のすべての列を生ずるが，

$C_{\sigma_1\rho}\geqq C_{\sigma_1}+\sum_{J_i\in\rho}t_i$ より，

$$C_{\sigma_1\rho\sigma_u}\geqq C_{\sigma_1}+\sum_{J_i\in\rho}t_i+\delta\left(C_{\sigma_1}+\sum_{J_i\in\rho}t_i,\ \sigma_u\right) \qquad \cdots\cdots\cdots\cdots(3)$$

を得るから，(3)における下限が $C_{\sigma_1\sigma_2\cdots\cdots\sigma_u}$ によってとられるならば，列 $\sigma_1\sigma_2\cdots\cdots\sigma_u$ を考察しても，C の改善は望めない。

ゆえに，かかる列 $\sigma_1\sigma_2\cdots\cdots\sigma_u$ を避ける。

第三及び第四の局面では，この選択条件は弱められ，($\sigma_1\sigma_2\cdots\cdots\sigma_u$ による配分において) 列 σ_u に属する仕事のみを含む最初の仕事場で順序づけが終ったかのように考える。そこでその仕事場の後で起るすべての仕事を σ_u から除いて得られる一つの列 $\sigma_{u'}$ を σ_u の代りに考察する。このことによって全体の費用 C を改善することは不可能ではあるが，最終的な費用改善の道をえる所の順序の変更に導くかもしれない誘導問題の取り扱いを許すという効果をもつ。

上に概観した逐次近似法の構成は，主として経験に基いてなされたいくつかの任意決定を含んでいる。次の項はこれらの決定が十分に満足すべき結果に導いたことを示している。

電子計算機による結果

前項で述べた逐次近似法を使用した IBM7090 に対する一つの経験的プログラムが若干の流れ作業調整問題を解くのに用いられた。

逐次近似の四つの局面が完了した時，または，最小数の仕事場を有する配分の得られたことが分った時プログラムは終了する。この第二の場合は，一つの実行可能解 $\sigma=(J_{i_1}, J_{i_2}, \cdots\cdots, J_{in})$ に対して「C_σ/T」=「$\sum_{i=1}^{n}t_i/T$」(「x」は x 以上の最小の整数) となるときに起る。

次に得られた結果の一覧表を与える。

各々の問題はサイクルタイム $T=1$ なるように標準化され，パラメーター $a=9$ である。また初期実行可能列 $\sigma^{(0)}$ は無作意に選んだ。

(i) Kilbridge と Wester による45個の仕事の問題

計算番号	初期費用	最終費用	$\sum t_i$	計算時間 (分)
1	9,072	8,000	8,000	1
2	9,319	8,000		1

(ii) 任意の t_i をもった111個の仕事の問題，優先関係は現実の組立流れ作業からとられた。

計算番号	初期費用	最終費用	$\sum t_i$	計算時間（分）
1	32,359	26,558	26,253	2
2	31,808	26,982		2

(iii) 無作意に選んだ加工時間と優先関係をもつ180個の仕事の問題

計算番号	初期費用	最終費用	$\sum t_i$	計算時間（分）
1	68.439	55,198	54,486	5
2	69,716	55,277		7

$a=6$ とすると5分で費用 54,964 の解を得た。

(iv) 任意に選ばれた t_i ときつい優先条件をもった400個の仕事の問題

計算番号	初期費用	最終費用	$\sum t_i$	計算時間（分）
1	95,905	84,952	84,295	3
2	97,905	84,905		3

(v) 555個の仕事の問題，それらの仕事は互の間に優先関係の存在しない111個ずつの仕事の5つの集合に分けられる。その各々における仕事は (ii) におけると同じ優先関係と加工時間を有する。

計算番号	初期費用	最終費用	$\sum t_i$	計算時間（分）
1	160,519	132,549	131,267	31
2	162,824	133,269		26

(vi) 無作為に選んだ加工時間と優先関係をもつ612個の仕事の問題

計算番号	初期費用	最終費用	$\sum t_i$	計算時間（分）
1	174,541	149,701	148,291	24
2	175,629	149,585		24

$a=6$ にとると30分で費用 148,970 をもつ解を得た。

実行可能集合に関する計算と写像の問題

ここで，上に与えられた流れ作業調整原理を計算機に履行させることに関して起った2つの問題を考察する。

それは，一つの半順序有限集合の実行可能部分集合について，

(a) 計算——実行可能集合の数を決定すること。

(b) 写像——実行可能集合を連続せる整数の集合の上へ一対一に写す写像 $A(S)$ を決定すること。

各実行可能集合 S に対し，$C(S)$ の記憶場所が対応するから，(a) の解は必要とする記憶場所の数を与え，演算に必要な時間の大ざっぱな測度を与える。

(b) の解は実行可能集合に連続せる記憶場所を配分することを決定する。

経験的プログラムにおいて，これらの問題は実行可能集合のリストを辞書編さん的に記憶することによって解かれる。そこでは，S と $C(S)$ とが各々のリストの最初の番地に関して同じ位置を占めるように，量 $C(S)$ は第2のリストに記憶される。第2のリストにおける $C(S)$ の位置は第1のリストに S を位置づけるための二分法 (binary search) を実行することによって決定されるであろう。その演算は実行される各二分法の範囲にきつい初期限界が課せられるように計画される。従って，かかる素朴な接近法は速く作動するが記憶を浪費するプログラムを生ずる。

2つのリストの記憶を必要としない別の方法は，探索手順によるよりは半順序の構造に基く一つの原理によって $A(S)$ を決定することである。

このような原理が存在すると仮定すれば，第一のステップは $A(S)$ の増加順序に実行可能集合を並べることである。集合 S を $C(S)$ で置き換えながらリストの順に D.P. の計算が進められる。$C(S)$ の計算には $C(S\text{–}l)$ が必要で，その位置は $A(S\text{–}l)$ によって決定さ条る。$C(S\text{–}l)$ は $C(S)$ の前に計算されねばならないから，A は $A(S\text{–}l)<A(S)$ なるように定められねばならない。理論的には，もし実行可能集合の数を計算し，そして A^{-1} を計算する原理が可能ならば，すべての実行可能集合の列挙は全く避けることができるであろう。この場合，記憶場所は逐次的に考察され，そして記憶場所 k に結びついた集合 S は $S=A^{-1}(k)$ として計算される。

以下において，実行可能集合を計算し，そして番地づける原理を与える。これらの原理は本質的に興味があり，きびしい記憶の制限がある場合にこの原理は有用である。

計算 (counting)

前に導入した実行可能集合という概念を相対化して考える。

$P=\{1, 2, \cdots\cdots, n\}$ を与えられた半順序 $<$ をもつ一つの集合とする。部分集合 $S\subseteq P$ は，$j\in S$ で $i<j$ のとき $i\in S$ となるとき P に関して実行可能であると云う。

B_0 は空集合で，$i>0$ に対して $B_i=\{j \mid j=i$ または $j<i\}$ なる実行可能集合：$B_0, B_1, \cdots\cdots B_n$ を P に関する基本実行可能集合という。

我々は P の要素の番号は半順序と適合すると仮定する：

すなわち，もし $j<i$ ならば $j<i$ である。

P に関して実行可能な集合全体は集合の結び (union) 及び交わり (intersection) なる演算の下に束 (lattice) を作り，基本実行可能集合はこの束の生成元 (generator) である。

P に関する基本実行可能集合 $B_0, B_1, \cdots\cdots, B_n$ に対して，

$$\bar{B}_i=\{j \mid j<i \text{ かつ } j\notin B_i\}$$

によって定義される集合 $\bar{B}_0, \bar{B}_1, \cdots\cdots, \bar{B}_n$ を，P に関する基本補元 (basic complements) という。すなわち $\bar{B}_i$ は i と優先関係のない i より低い番号の要素から成っている。

次の定理が示すように基本補元 $\bar{B}_i$ は実行可能集合の数の計算において中心的な役割を演ずる。

定理 1. $[P]=\sum_{i=0}^{n}[\bar{B}_i]$ である。

ここに $[X]$ は半順序集合 X の実行可能部分集合の数を表わす。半順序集合の部分集合は誘導された順序の下に半順序であると考えられる。

証　明

P に関して実行可能な最高指数 k の集合 S と，$\bar{B}_k$ に関して実行可能な集合 R の間には，$S=B_k \cup R$ により，一対一の対応が存在する。従って $[\bar{B}_k]$ は P に関して実行可能な最高指数 k の集合の数である。よって定理が成立つ。（証終）

実行可能集合の数を計算するのに有用な別の概念は半順序集合 P の成分 (component) なる概念である。

P の2つの要素 a と b とは，もし P の要素 $x_1, x_2, \cdots\cdots, x_\eta$ が存在して a は x_1 と比較でき，x_1 は x_2 と比較でき，……，x_η は b と比較できるとき，連結している (connected) と呼ばれる。連絡性の関係は同値関係であるから，半順序集合 P は互に素な同値類 P_i（これを P の成分と呼ぶ）に唯一通りに分解できる。

P の2つの要素 i と j とが $i<j$ で $i<k<j$ なる第3の要素 k が存在しないとき，i を j の直前元，j を i の直後元という。i が j の直前元であるとき i から j への辺が存在する如き有向グラフによって半順序集合を表わすと便利である。そのとき P の成分はそのグラフの連結部分である。実行可能集合を計算するときの成分の役割は次の定理による。

定理　2.　半順序集合 $P=\{1, 2, \cdots\cdots, n\}$ の成分を $P_1, P_2, \cdots\cdots, P_r$ とすると，

$$[P]=\prod_{i=1}^{r}[P_i] \quad \text{である。}$$

証　明

$P, P_1, P_2, \cdots\cdots, P_r$ に関して実行可能な集合をそれぞれ $S, S_1, S_2, \cdots\cdots, S_r$ とすると，$S \Rightarrow S_1 \cup S_2 \cup \cdots\cdots \cup S_r$ なる関係により，S と r 個の組 $(S_1, S_2, \cdots\cdots, S_r)$ との間に一対一の対応が成り立つことから定理が成立する。（証終）

実行可能集合を計算するための一般的手順を詳しく述べる前に，特種な型の半順序集合にのみ応用可能な計算技術を与えよう，この技術は一般的手順の一部として有用であることを後で示す。

T は次の性質をもつ半順序集合であるとする。

(a)　直前元をもたない唯一の要素が存在する。この要素を T の根 (root) と呼ぶ。

(b)　他のすべての要素は唯一つの直前元をもつ，従って T に関する有向グラフは根から出るあらゆる有向矢をもつ一つの樹 (tree) である。

定理　3.　$[T]$ は次のようにして決定される。

(a)　直後元をもたない各要素に数2をつける。

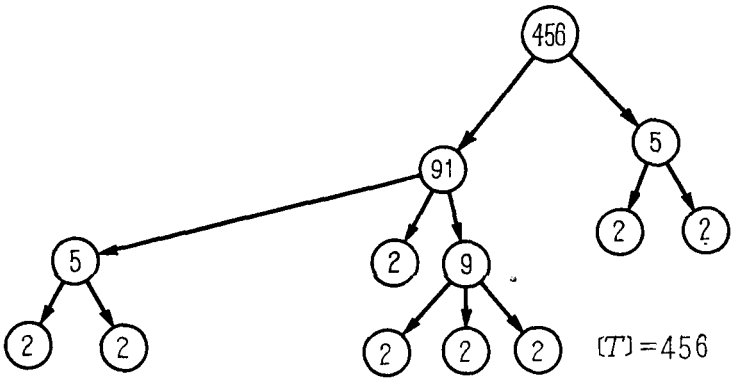
(b)　逐次的に，各要素にそれのいくつかの直後元についた数の積に1を加えた数をつける。（図4参照）

(c)　$[T]$ は根についた数に等しい。

証　明

上のようにして各要素につけた数がその要素とそれの後のすべての要素から成る半順序集合に関する実行可能集合の数を与えることに注意すれば帰納的に証明される。（証終）

（**例 1**）



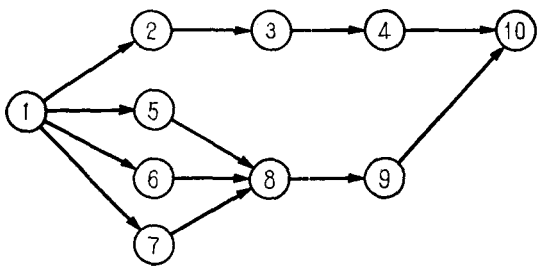
第 4 図

P を任意の半順序集合とするとき，P^1 がその半順序を逆にすることによって P から得られる半順序集合であるとしても（すなわち，P に関する有向グラフにおいて矢を逆向きにする），$[P]=[P^1]$ であることが実行可能集合の定義より成り立つ。我々はこのことを“逆向きの樹”(reversed trees) の場合に応用することができる。

一般の半順序集合 P に対しての実行可能集合の計算は定理 1，2，及び 3 を繰り返し応用することに基く一つの原理によって達せられる。この計算手順の一般的段階において実行可能部分集合の数の未決定な半順序集合の表が考察される。最初，この表は P のみを含む。各段階において最大数の要素をもつ集合 S が次のようにして表から除かれる。（例 2 参照）

もし S が一つの樹または逆向きの樹に対応すれば $[S]$ は定理 3 より決定される。そうでなければ，S が成分 S_i をもてばそれらの成分が表に加えられ $[S]=\Pi[S_i]$ となる。もし S が成分に分裂しないならば S に関する基本補元 $\bar{B}_i$ が表に加えられ $[S]=\sum[\bar{B}_i]$ となる。このようにして終には一つの要素のみの集合のみ表に残り，これらの集合は定理 3 によって取り扱うことができる

（**例 2**）P は次のグラフによって表わされているとする。



第 5 図

から $[P]$ が計算される。

この手順は計算機に対して効果的にプログラムすることは，表の構造や型認識の必要故に困難であるかも知れないが，前の例の示すように手計算に適している。

先ず第一に，表は単一集合 $\{1,2,3,4,5,6,7,8,9,10\}$ を含む。定理2及び3は適用され得ないから定理1が用いられ，次のようになる。

$$[\{1,2,3,4,5,6,7,8,9,10\}]=[\Phi]+[\Phi]+[\Phi]+[\Phi]+[\Phi]+[\{2,3,4\}]+[\{2,3,4,5\}]$$
$$+[\{2,3,4,5,6\}]+[\{2,3,4\}]+[\{2,3,4\}]+[\Phi]\quad(\Phi\text{ は空集合})$$

この段階では表は $\{2,3,4,5,6\}$，$\{2,3,4,5\}$，$\{2,3,4\}$，Φ を含んでいるから最大数の要素を含む集合 $\{2,3,4,5,6\}$ が選ばれ表から除かれる。この集合に対して定理3は応用できないが定理2より，$[\{2,3,4,5,6\}]=[\{2,3,4\}]\cdot[\{5\}]\cdot[\{6\}]$ となる。そこで表は，$\{2,3,4,5\}$，$\{2,3,4\}$，$\{5\}$，$\{6\}$。Φ となる。このようにして以下次の各段階が続く。

$[\{2,3,4,5\}]=[\{2,3,4\}]\cdot[\{5\}]$　（定理2）

$[\{2,3,4\}]=4$　（定理3）

$[\{5\}]=2$　（定理3）

$[\{6\}]=2$　（定理3）

$[\Phi]=1$

各段階を逆にたどって，

$[\{2,3,4,5\}]=4\times2=8$，$[\{2,3,4,5,6\}]=4\times2\times2=16$ よって最後に，

$[\{1,2,3,4,5,6,7,8,9,10\}]=1+1+1+1+1+4+8+16+4+4+1=\underline{42}$

を得る。

かかる手順が必要とする仕事量は要素に番号つける方法に依存する。仕事量を少なくするためのよい規則は，$j<i$ のとき $j<i$ なるように直列的に順序のついた要素に連続する番号をつけることである。

写像 (mapping)

この項で我々は半順序集合 P に関して実行可能な集合 S の各々を整数 $0,1,2,\cdots\cdots,[P]-1$ の上に写像する一つの関数 $A(S\mid P)$ を構成する問題を考察する。この関数は繰返し的に定められ，そしてその構成は前項の計算手順において用いられた定理 1，2 及び3に基いている。その定義は次の3つの場合の考察を必要とする。

第一の場合（定理3に対応する場合）：P に関する有向グラフが根から出ている有向矢をもつ一つの樹であると仮定する。もし P が唯一の要素 i から成る特種な場合ならば，そのグラフは一つの樹であり，$A(\Phi\mid P)=0$，$A(\{i\}\mid P)=1$ と定める。そうでない場合には，根に対応する P の要素が除かれると，残りの半順序集合は部分樹 (subtrees) に対応する連続成分 P_1，P_2，$\cdots\cdots$，P_r をもつ，そこで関数 $A(S\mid P)$ は次のように構成される：

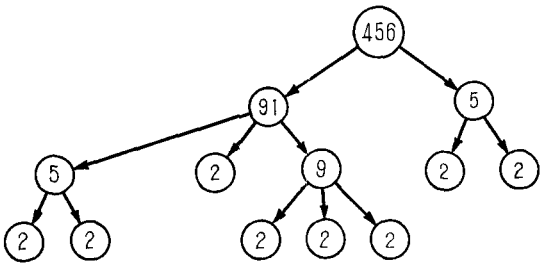
(a)　S が空集合のとき，$A(S\mid P)=0$;

(b) S が空集合でないとき，

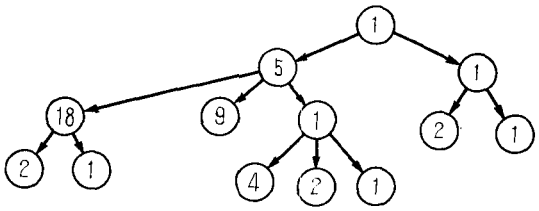
$$A(S\mid P)=A(S\cap P_r\mid P_r)\prod_{i=1}^{r-1}[P_i]+A(S\cap P_{r-1}\mid P_{r-1})\prod_{i=1}^{r-2}[P_i]+\cdots\cdots+A(S\cap P_2\mid P_2)[P_1]$$
$$+A(S\cap P_1\mid P_1)+1$$

単一の要素の樹のみが残るまで同じ型の分解が P の各部分樹に対し繰返し応用される。また，逆向きの樹も上と同様に取扱われる。以上の過程は番号をつける簡単な手順によって完成されたことを例1の半順序集合につて説明しよう。

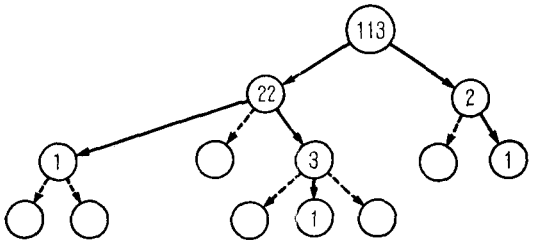
（**例3**）



第6図 (a)



第6図 (b)



第6図 (c)

第6図 (a) は [p] を決定するとき用いた例1の第4図である。第6図 (b) の node 内の数は，その node の右にあり，同じ直前元をもつ nodes の第6図 (a) における数の積である。もしそのような nodes が存在しなければこの積は1に等しいとおく，部分樹は右から左へ番号づけられるという約束の下にこの過程は (b) における重さ $[P_1], [P_1][P_2], \cdots\cdots$ を与える。第6図 (c) においては実線の矢で連結している nodes は一つの実行可能集合 S を表わしている。この S の nodes は次のようにして番号づけられる：S のすべての最後の nodes に1をつける，その後のすべての nodes が番号づけられているような S の node に対する番号は，その後の nodes の番号と第6

図 (b) における対応する番号との積を作り，これらの積を加えてそれに 1 を加えることによって得られる。この例では，かくして

$A(S|P)=113$ を得る。

第二の場合　（定理 2 に対応する場合）：P は成分 $P_1, P_2, \cdots\cdots, P_r$ をもつとすると，P に関して実行可能な一つの部分集合 S に対して，

$$A(S|P)=A(S\cap P_r|P_r)\prod_{i=1}^{r-1}[P_i]+A(S\cap P_{r-1}|P_{r-1})\prod_{i=1}^{r-2}[P_i]$$
$$+\cdots\cdots+A(S\cap P_2|P_2)[P_1]+A(S\frown P_1|P_1)$$

である。

第三の場合　（定理 1 に対応する場合）：P が唯一つの成分のみをもち，一つの樹に対応しないならば，

$$A(S|P)=\sum_{j=0}^{t-1}[\bar{B}_j]+A(S\cap\bar{B}_t|\bar{B}_t)$$

である。ここに i は P の実行可能な部分集合の最高指数で，$\bar{B}_k$ は P に関する基本補元である。

以上の繰返し的定義を繰返し応用すると $A(S|P)$ に対する数値を得ることができる。

（例 4）

例 2 の半順序集合 P を用いると，$A(\{1,2,3,4,5,6\})|\{1,2,3,4,5,6,7,8,9,10\})$ は次のようにして計算される：

$A(\{1,2,3,4,5,6,\}|\{1,2,3,4,5,6,7,8,9,10\})=9+A(\{2,3,4,5\}|\{2,3,4,5\})$（場合 3）

$A(\{2,3,4,5\}|\{2,3,4,5\})=A(\{5\}|\{5\})[\{2,3,4\}]+A(\{2,3,4\}|\{2,3,4\})$（場合 2）

$$=4A(\{5\}|\{5\}+A(\{2,3,4\}|\{2,3,4\})$$

$A(\{2,3,4\}|\{2,3,4\})=3$（場合 1）　$A(\{5\}|\{5\})=1$（場合 1）

従って，$A(\{2,3,4,5\}|\{2,3,4,5\})=4\times1+3=7$，$A(\{1,2,3,4,5,6\}|\{1,2,3,4,5,6,7,8,9,10\}$

$$=9+7=16$$

以上与えられた手順は整数 0, 1, 2……, $[P]-1$ の集合の上への一対一写像である一つの関数 $A(S|P)$ を与える。そしてそれは，$S_1\subset S_2$ ならば $A(S_1|P)<A(S_2|P)$ という性質をもっている。

上に与えられたと類似の手順により $A(S|P)$ の増加する順に P に関する実行可能集合 S を数えるための関数 A^{-1} を与えることができる。これらの手順は要素，部分樹，成分の番号づけに依存することに注意しなければならない。特に番地づけの関数 $A(S|P)$ とその逆関数 A^{-1} の定義はどんな約束 (conventions) が選ばれるかに従って変るであろう。しかし，そのどんな選択に対しても要求された性質をもつこれらの関数を与えることができる。

Jackson [4] による原理

各実行可者集合 $S\subseteq\{J_1, J_2, \cdots\cdots, J_n\}$に対し，$C(S)$ より大きいか又はそれに等しい最小の整数 $D(S)$ を対応させる。

従って $D(S)$ は S 内の仕事の配分に必要な仕事場の最小数を与える。もし $D(S)=k$ で $D(T)$

$=k$ なる実行可能集合 $T \supset S$ が存在しないとき，実行可能集合 S は k-maximal と呼ばれる。特に $D(\{J_1, J_2, \cdots\cdots, J_n\}) = k^*$ ならば $\{J_1, J_2, \cdots\cdots, J_n\}$ は唯一の k^*-maximal な集合で，k^* は組立流れ作業に必要な仕事場の最小数を与える。計算は系統的に 1-maximal, 2-maximal, …… k^*-maximal な集合を数え上げることによって進められる。このことが達成されたならば既に述べたと類似の手順によって最小の仕事場をもつ一つの配分が得られる。

k-maximal 集合から $(k+1)$-maximal 集合を得るには次の方法による。もし S が k-masimal で，(1) $S \cup R$ が実行可能で，(2) $\sum_{i \in J_i \in R} t_i \leqq T$ で，そして(3)(1)及び(2)を満足する集合 $R^1 \supset R$ が存在しないとき，R は 1-maximal $|S$（これを "1-maximal given S" と読む）であるという。

各の k-maximal な集合 S に対して 1 maximal $|S$ である集合 R の表が第2表に示した原理を用いて得られる。

第2表 1-maximal $|S$ なる集合 R を得るための原理

No.	段　　階	解　　析
1.	L は最初 $\{\phi\}$ とおく。	L＝半順序的 1-maximal sets given S の表 ϕ＝空集合（$\{\phi\}$ は空の表でないことに注意）
2.	任意の $V \in L$ を選び，$k = \max\{i \mid J_i \in V\}$ とおく。	もし $V = \phi$ ならば $k = 0$
3.	L より V を削除する。	
4.	$l > k$ で性質 P を満足するすべての $J_l \notin S$ に対して，$V \cup \{J_l\}$ を L 内に入れる。もしかかる J_l が存在しないときは5に移り，そうでないときは2に移る。	性質 P：(i) $S \cup V \cup \{J_l\}$ が実行可能 (ii) $\sum_{i \in V \cup \{J_l\}} t_i \leqq T$ 我々は $J_m < J_n$ ならば $m < n$ なるように番号づけられていると仮定する。
5.	V を M 内におく。	M＝1-maximal $|S$ である集合の表。
6.	L が尽くされたならば止める。そうでないならば2に移る。	

このとき，$\Omega = \{S \cup R \mid S$ は k-maximal, R は 1-maximal given $S\}$ なる表 Ω は容易に得られる。すべての $(k+1)-$maximal 集合は S が k-maximal で R が 1-maximal $|S$ として $S \cup R$ の形に表わせるから Ω はすべての $(k+1)-$maximal 集合を含む。しかしこの逆は云えないから，Ω から余分を除くために Ω を，仕上げて (trimm)，$(k+1)-$maximal な集合のみを残さねばならない。

Held, Karp, Shareshian の原理による手順の量と Jackson の原理による手順の量の比較。

前に述べた原理による計算量は，本質的には量 $Q = C(S - J_l) + \triangle[C(S - J_l), t_l]$ を評価する時間数によって決定される。そしてこれは $(S, S - J_l)$ の形の実行可能集合の対の数である。優先関係のない特種な場合には，かかる対の数は

$\sum_{k=1}^{n} k\binom{n}{k} = n2^{n-1}$ である。ここに n は仕事の数である。Q を評価するための時間の大部分が計算にではなく，$C(S-J)$ が位置する記憶場所を決定するのに使われることに注意しなければならない。（計算及び写像問題参照）これに対し Jackson の原理に対し必要な計算量を推定するに

は，第2表の原理の実行と表 Ω を仕上げる過程とを考察することが必要である。第2表の原理を実行することに費される時間量は，本質的には性質 P（第2表参照）がテストされる時間数に比例する。優先関係が存在せず，各仕事が加工時間 T/q（q は正整数）をもつという特種な場合に対しては，性質 P のテストの数は，

$$\sum_{k=0}^{[\frac{n}{q}]-1} \binom{n}{kq}\left[\binom{n-kq}{1}+\binom{n-kq}{2}+\cdots\cdots+\binom{n-kq}{q+1}\right]$$

に等しい。n 及び q の多くの値に対して，これは我々の原理に必要な Q の $n2^{n-1}$ なる評価よりはるかに大きい。また，表 Ω を仕上げることは全く時間の浪費であることを証明できる。上式において $n=12$, $q=3$ とおくと 2-maximal 集合の数は $\binom{12}{6}=924$ である：しかしながら仕上げの前に表 Ω は各集合の20個の場合を含む。ひかえ目にみても，その仕上げに必要とされる比較の数は 10^7 を越えるであろうと推定される。その比較は一様には我々の方法に好都合ではなく，手計算に従う小数例においては特に，Jackson の方法がある場合にはより効果的である。しかしながら我々の原理は多くの場合において優れているし，更に，単に実行可能集合を計算することによって，その原理を応用するとき必要な記憶の数を前もって予測することができる。かかる簡単な予測は Jackson の原理の場合において可能であるとは思えない。

文　　献

1) M. D. Kilbridge and L. Wester, " A Review of analytical Systems of Line Balancing." Oper. Res. Vol. 10. (1962) pp. 626～638

2) M. Held. R. M. Karp and R. Shareshian, " Assembly—Line Balancing—Dynamic Programming with precedence Constraints " Oper. Res. Vol. 11. *No.* 3 (1963) pp. 442～459

3) M. E. Salveson, " The Assembly Line Balancing Problem," J. Ind. Eng. 6 (1955) pp. 18～25

4) J. R. Jackson, "A Computing Procedure for a Line Balancing Problem," Manag. Sci. Vol. 2 No. 3 (1956) pp. 261～271

5) E. H. Bowman, "Assembly–Line Balancing by Linear Programming," Oper. Res. Vol. 8 (1960) pp. 385～389

6) F. M. Tonge, " Summary of a Heuristic Line Balanceing Procedure," Manag. Sci. Vol. 7 (1960) pp. 21～39

——, A Heuristic Program of Assembly Line Balancing, Prentice Hall. 1961

7) M. Kilbridge and L. Wester, "A Heuristic Method of Assembly Line Balancing," J. Ind. Eng. Vol. 1 2. No. 4 (1961) pp. 292～298

——, " Heuristic Line Balancing: A Case," J. Ind. Eng. Vol. 13 (1962) pp. 139～149

——, " The Balance Delay Problem," Manag. Sci. Vol. 8 (1961) pp. 69～84

8) M. Held and R. M. Karp, "A Dynamic Programming Approach to Sequencing Problems," J. Soc. Ind. appl. Math. Vol. 10 No. 1 (1962) pp. 196～210

9) M. Klein, " On Assembly Line Balancing," Oper. Res. Vol. 11 No. 2 (1963) pp. 274～281

10) W. B. Helgeson and D. P. Birnie, "Assembly Line Balancing Using the Ranked Positional Weight Technique," J. Ind. Eng. Vol. 12 (1961) pp. 394～398

11) J. W. Burgeson and T. E. Daum, " Production Line Balancing," 62 pp., File Number 10. 3. 002, I. B. M. Corp., Akron Ohio, 1958

1964年度秋季研究発表会

本誌第7巻3号で秋季研究発表会について，関西と四国の予告をいたしましたが，その後関西地区で開催することに決定いたしました。日時と場所は次の通り予定しております。多数御参加下さるようお願い致します。

記

日時　11月5，6，7日又は12，13，14日

場所　甲南大学

会員名簿作成について

本学会では，この度会員名簿を作成のため準備をすすめておりますが，1962年度以降住所，勤務先，その他を変更されて，学会事務局にまだ届出のなされていない方は至急御連絡下さるようお願い致します。永らく名簿の発行がなく，会員相互の連絡にも御不便をおかけの事と思いますが，もうしばらくお待ち下さい。