Dominant schedules of a steady job-flow pair*

By J. TANKÓ

A specific approach to some non-finite deterministic scheduling problems is the scheduling of a steady job-flow pair model. Its non-preemptive scheduling problem was discussed earlier [4]. The more general preemptive scheduling is discussed below. A very simple scheduling discipline leads to the dominant set of the so-called consistent economical schedules (CESs). The proof of dominance is the main goal of this article. An algorithm to evaluate the dominant schedules and choose an optimal one is given as well.

1. Introduction

In an earlier article [4] we defined the general scheduling model of steady job-flow pairs as a new approach to some non-finite deterministic scheduling problems. There we referred to the study [2] and to the dissertation [3] of the author dealing with this problem and to other works dealing with scheduling problems related to our problem. Some practical cases the model may be applicable in are mentioned there.

Some statements below bear some resemblance to those of non-preemptive scheduling [4] but, for example the cardinal of the dominant set, is not bounded as in the non-preemptive case. The task of determining the optimal schedule under the restriction of non-preemption is simpler than without this restriction. In a non-preemptive case the dominant set of the so-called consistent natural schedules have six elements maximum. These elements can be evaluated at once, e.g., by the method of reduction [4]. The general problem of determining or producing an optimal schedule (preemptive if necessary) for any steady job-flow pair is not completely solved until now.

We reduce below the set of feasible schedules to a dominant set of consistent economical schedules containing optimal schedules and give an algorithm to choose an optimal schedule by evaluation of the whole set if it is finite.

* This article reports on some results of a study of the author supported by the Computer and Automation Institute of the Hungarian Academy of Sciences.

2. Definitions

The scheduling problem of steady job-flow pairs is to schedule three processors $\mathcal{P}=(P_A, P_{B1}, P_{B2})$ to service, without conflicts, pairs $Q=(Q^{(1)}, Q^{(2)})$ of steady job-flows $Q^{(i)}=\{C_{ij}, j=1, 2, ...\}$ consisting of task-pairs $C_{ij}=(A_{ij}, B_{ij})$ with service demands η_i and ϑ_i on processor P_A and P_{Bi}, respectively. The order of servicing the tasks is strictly serial within job-flows but it is not restricted among job-flows. Conflicts might only be on the processor P_A and the efficiency of a scheduling R is measured by the utilization of the processor P_A. Define P_A*-utilization* of a section from time t_1 to time t_2 of a scheduling R by $\lambda(t_1, t_2)/(t_2-t_1)$ with P_A*-usage* $\lambda(t_1, t_2)$ as the sum of activity durations of P_A in the while from t_1 to t_2. Let $\lambda(t)=\lambda(0, t)$. The *efficiency* of a scheduling R is defined by the limit

$$\gamma = \gamma(R) = \lim_{t\to\infty} \frac{\lambda(t)}{t}. \tag{1}$$

The efficiency of any scheduling cannot be greater than 1 or the sum $\gamma^{(1)}+\gamma^{(2)}$ of the P_A-utilizations of the job-flows $Q^{(1)}$ and $Q^{(2)}$ which are given by $\gamma^{(i)}=\eta_i/\tau_i$, $i=1, 2$. We use the notations

$$\tau_i = \eta_i+\vartheta_i, \quad i = 1, 2, \quad \eta = \eta_1+\eta_2, \quad \vartheta = \vartheta_1+\vartheta_2.$$

The scheduling procedure is a decision process determining for all moment $t\geqq 0$ and state of processors and job-flows the way of continuation of the servicing process. The plan or result of a scheduling procedure is a *schedule* R as an ordered set of situations σ. The *situation* σ characterises the state of processors, the state of demand cycles under service, if any, of both job-flows and the duration of these states in a given phase of the scheduling.

Two components of σ are the functions $\beta^{(i)}(t)$, $i=1, 2$, $t\geqq 0$, the value of $\beta^{(i)}(t)$ being the demand not served yet from the demand cycle started but not finished (active), if it exists, of the job-flow $Q^{(i)}$, and 0 otherwise.

A schedule is *consistent* if the scheduling decision is the same when the situation σ has the same value. A schedule is *tight* if processor is never idle when demand it could serve exists. A schedule is *non-preemptive* if the service of every task finishes without breaks after its beginning. The specific class of non-preemptive schedules is discussed in [4]. Here now we allow the service of a task to be preempted and resumed later on the same processor.

The instance of a scheduling problem is fully determined by the values $Q=(\eta_1;\ \vartheta_1;\ \eta_2;\ \vartheta_2)$ of the service demands of tasks type A_1, B_1, A_2, B_2, respectively. $\eta_1, \vartheta_1, \eta_2, \vartheta_2$ are called *parameters* and the quaternaries Q are called *configurations*. The non-negative sixteenth $\mathcal{Q}$ of the four-dimensional Cartesian space constitutes the *configuration space*. The goal of the study of the model defined is to find a method for choosing a schedule R^* for every configuration $Q\in\mathcal{Q}$ for which $\gamma(R^*)$ exists and has the maximum value among all the feasible schedules. This schedule is called an *optimal schedule*. Simple method for finding optimal schedule for all $Q\in\mathcal{Q}$ i.e. an optimal scheduling strategy is not found yet.

Two schedules R and R' are *essentially-the-same* and denoted by $R\approx R'$ if they are congruent after some finite initial sections of them. $\gamma(R)=\gamma(R')$ if $R\approx R'$. The schedule R' *dominates* the schedule R if for the efficiency values $\gamma(R')$ and $\gamma(R)$

defined by (1) the relation $\gamma(R') \geqq \gamma(R)$ is true. The set $\mathscr{R}'$ of schedules is a *dominant set* if for every feasible schedule R there exists an $R' \in \mathscr{R}'$ dominating it.

Looking for an optimal schedule the investigation of a dominant set $\mathscr{R}'$ is enough for. We obtain a dominant set of schedules by means of the concept of the dominant decision.

The scheduling *decision* s' *dominates* s in a situation σ if the minimal next following cycle-finishes of both job-flows are not later by s' than by s. A *decision* s is *economical* if decision s' dominating it does not exist (see Fig. 2 below). A schedule R is an *economical schedule* (ES) if the scheduling decisions in its every situation are economical. Let $\mathscr{R}(Q)$ denote the class of all economical schedules for the configuration $Q \in \mathscr{Q}$. Let $\mathscr{R} = \bigcup_{Q \in \mathscr{Q}} \mathscr{R}(Q)$. We will show that $\mathscr{R}$ is a dominant set of schedules.

3. Economical schedules

The importance of the economical schedules (ESs) lies in their dominance which we show below.

Theorem 1. *The class* $\mathscr{R}$ *of economical schedules constitutes a dominant set.*

Proof. Let R be any feasible schedule having scheduling decisions not economical. Let s be a not economical decision in the situation σ of R. There exists an economical decision s' in σ dominating s because s would be economical decision otherwise. By exchanging s for s' both the next following cycle-ends could come forward and this eventually makes possible to anticipate all cycle-ends. This transformation does not diminish the function $\lambda(t)$ and, consequently, γ in (1). The new schedule R' obtained by this transformation dominates R as a result. Starting from $t=0$ and initial situation $\sigma=\sigma_0$, we can construct a dominating ES R' for any feasible schedule R. This was to be proven. □

The class $\mathscr{R}$ is a true part of the set of all feasible schedules but it can be very big to choose an optimal schedule by direct evaluations. To show this and to look for further reduction of the dominant set we investigate the characteristics of the ESs.

It is easy to be seen that the economical decision is unique in all situations σ except an enumerable set of situations for every ES. The exceptional situations are called *critical situations*. The economical decisions made in this situations are defined as *critical decisions*. The initial situation σ_0 of every schedule and the initial decision s_i, $i=1, 2$, for servicing the task A_{i1} first, are always critical but we mean by first critical situation of an ES the next one if it exists. Fig. 1 shows the types of critical situations and the possible alternative critical decisions. These and their conditions are the following:

Type	Decisions	Conditions
σ_0	s_1, s_2	$\beta^{(1)}(t) = \beta^{(2)}(t) = 0$
$\sigma_{i,1}$	s_0, s_i	$\beta^{(i)}(t) = 0, \quad \vartheta_{3-i} < \beta^{(3-i)}(t) < \tau_{3-i}, \quad i = 1, 2$

Fig. 2 illustrates the dominance of scheduling decisions. The graphs (a) and (b) illustrate that the idleness of a processor cannot be a dominating decision if

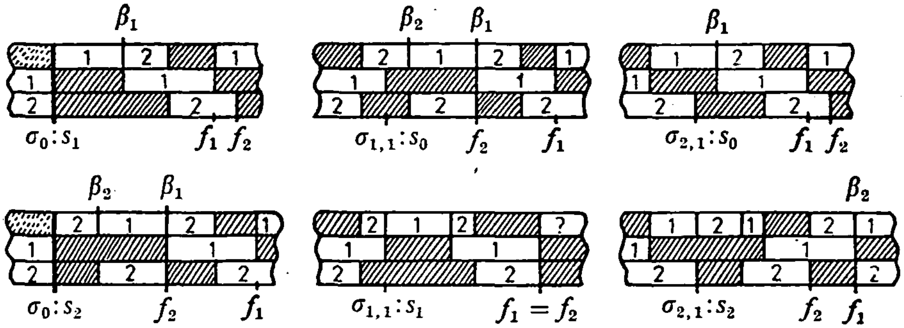


Fig. 1
Critical situations and decisions

demand waiting for service does exist. The graphs (c)—(d) show that the decisions s_i' causing preemption for not a complete service of the preempting task are not dominant as well. The graph (e) shows the non-dominance of the preemption of a preempting task.

It follows that the ESs are tight, usually preemptive schedules but have no superfluous preemptions. Only cycle-ends f_i can be critical situations and they really are if the processor P_A is busy or demanded simultaneously by the other

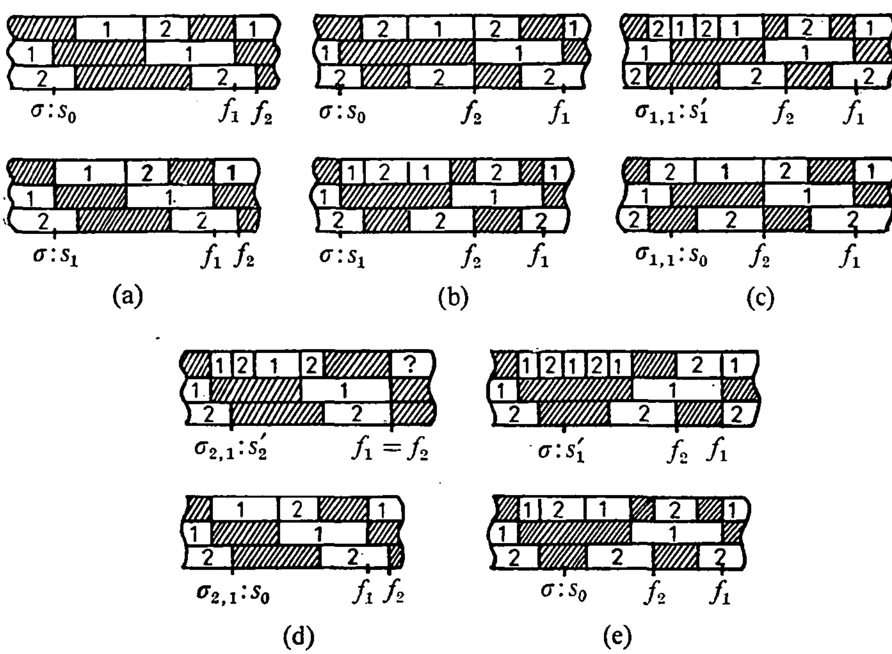


Fig. 2
Dominating decisions

job-flow. Preemption can only occur in critical situations and every critical decision causes a delay of the service of the job-flow not preferred by the decision. Delay is not caused by decisions other than critical. Between critical situations the sections of any ES are uniquely determined by the initial situation and decision. These sections are, therefore, called *determined sections*. The infinite section starting with the last critical situation if it exists, is the *last* determined section.

All ESs start with the service of the task A_{i1} without preemption in the interval $(0, \eta_i)$ in accordance with the initial decision s_i, $i=1, 2$. Accordingly, the class $\mathscr{R}$ bursts into two subclasses $\mathscr{R}^{(i)}$, $i=1, 2$, consisting of ESs with the initial decisions s_i, $i=1, 2$, respectively. The initial decision s_i uniquely determines the first determined section together with the closing critical situation — the first — if it exists. It follows that all elements of $\mathscr{R}^{(i)}(Q)$ have the same first determined sections and critical situations σ_i' if the latters exist at all. Let T_i' be the length of the first determined section. There is no preemption and delay on the first determined section except the initial delay of $Q^{(3-i)}$ in the interval $(0, \eta_i)$. Use the notation $\sigma^{(i)}$ for the situation of schedules $R \in \mathscr{R}^{(i)}$ in the point $t_i'=\eta_i$.

The concepts of critical situation and decision were introduced for the natural schedules defined in [4] as well. The types of critical situations were σ_0 and $\sigma_{i,0}$, $i=1, 2$, and the conditions for σ_0 were the same as here. The conditions of $\sigma_{i,0}$ there and the Fig. 1 show that a situation type $\sigma_{i,1}$ in ESs is always preceded by a situation type $\sigma_{3-i,0}$ being critical situation of a natural schedule but not of an economical one. This simultaneousness of $\sigma_{3-i,0}$ and $\sigma_{i,1}$ has a particular importance at the first determined sections playing a central role in the discussion of ECs (see Theorem 2). Out of types $\sigma_0, \sigma_{i,0}$ and $\sigma_{i,1}$ the natural and economical decisions are the same for every situation and cause no preemptions or delays. The first determined sections for the ESs are, therefore, almost the same as for the natural schedules. The differences are only in the last subsections of the ESs starting with $\sigma_{3-i,0}$ and ending with $\sigma_{i,1}$. The processor P_A is busy throughout the subsections. If the first critical situation does not exist, the set $\mathscr{R}^{(i)}$ consists of a single schedule R_{i0} being natural schedule, simultaneously.

The connection between the first critical situations of the natural and economical schedules allow us to simply prove an important theorem concerning typical situations by reference. *Typical situations* of an ES are defined as its critical situations and the β_i-situations which are not $\sigma^{(i)}$ situations directly following critical situations [4]. *β_i-situation* is a situation in which an A_i-task finishes and an A_{3-i}-task starts at the same moment. Let σ_i^* denote the first typical situation of the ESs of $\mathscr{R}^{(i)}(Q)$ if it exists. The possible first typical situations are illustrated in Fig. 3. We also use the wording *characteristic situations* for the critical and every β_i-situations.

Theorem 2. *In one and the same cases all elements of $\mathscr{R}^{(a)}(Q)$ have a first typical situation σ_a^* iff the simultaneous inequalities*

$$0 \leqq \Delta_a \leqq \eta, \quad \omega_a \geqq (1, 0) \tag{2}$$

have a solution, where $\omega_a=(B_a, A_a)$ are integers and $\Delta_a \equiv B_a\tau_a - A_a\tau_{3-a}$, $a=1, 2$.

When (2) has no solution, $\mathscr{R}^{(a)}(Q)$ consists of the single (non-preemptive and consistent) schedule R_{a0}. This occurs in the cases

$$\eta=0, \ \vartheta_1 \text{ and } \vartheta_2 \text{ are rationally independent} \tag{3}$$

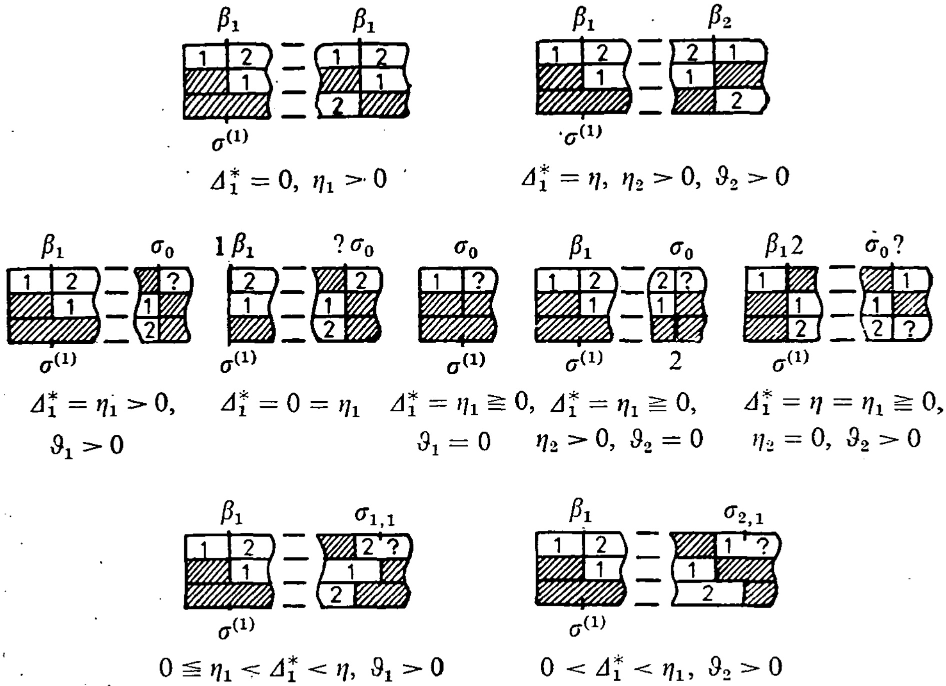


Fig. 3

First typical situations and their conditions ($\Delta_1^* \equiv B_1^* \tau_1 - A_1^* \tau_2$)

and

$$\vartheta_a > 0, \quad \tau_{3-a} = 0. \tag{4}$$

When (2) *has a solution, the type (and place) of* σ_a^* *is determined by the error* Δ_a^* *of the least solution* $\omega_a^* = (B_a^*, A_a^*)$ *of* (2) *according to the table*

σ_a^*	Conditions
β_a	$\Delta_a^* = 0 < \eta_a$
β_{3-a}	$\Delta_a^* = \eta > \eta_a, \quad \vartheta_{3-a} > 0$
σ_0	$\Delta_a^* = \eta_a$ *or* $\Delta_a^* = \eta > \eta_a$ *but* $\vartheta_{3-a} = 0$
$\sigma_{a,1}$	$\eta_a < \Delta_a^* < \eta$
$\sigma_{3-a,1}$	$0 < \Delta_a^* < \eta_a$

Proof. The assertions of the theorem follow from Theorem 4 of the article [4] and the comments made above. □

The problem of finding the least solution of (2) is a coincidence problem [2].

If $\sigma^{(a)}$ is not a critical situation, it is always a β_a-situation. It follows that β_a returns periodically and σ_a' does not exist if $\sigma_a^* = \beta_a$. If $\sigma_a^* = \beta_{3-a}$ then the first

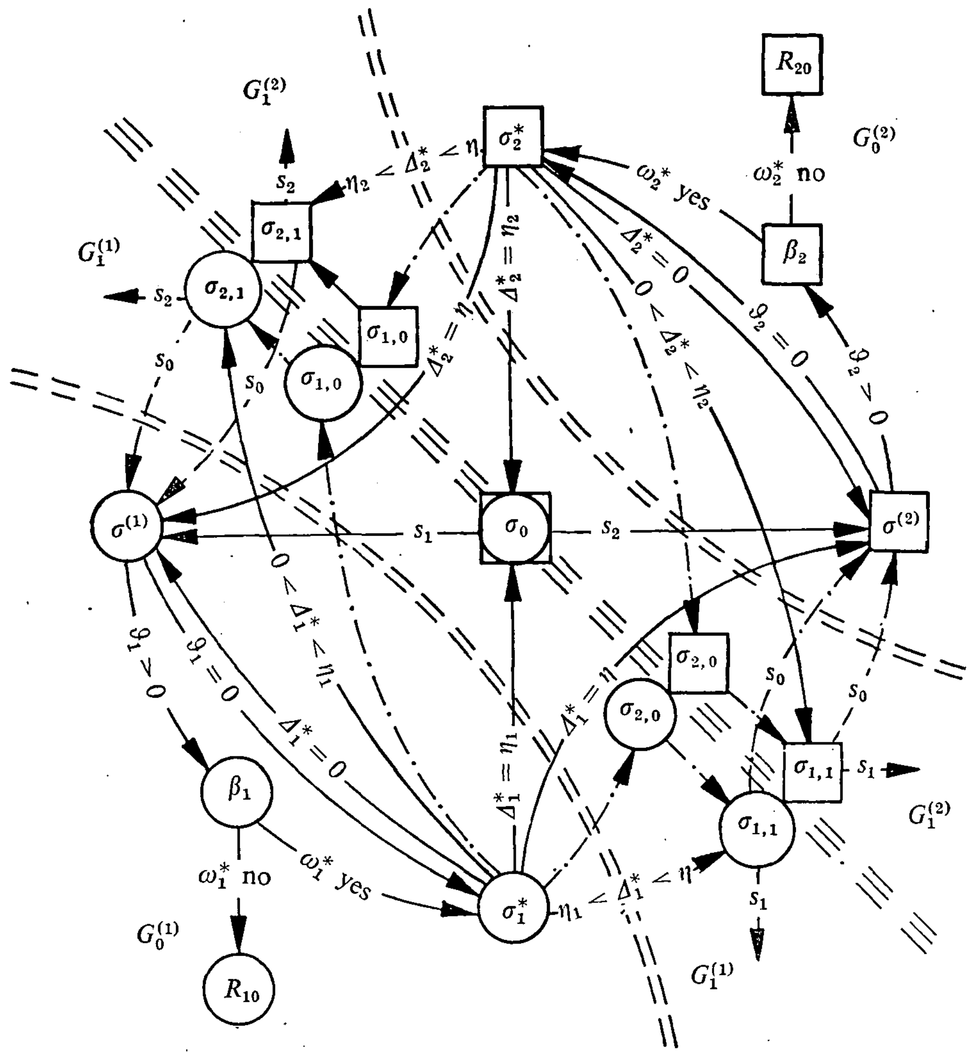


Fig. 4
The cyclic graph G_0 of the first determined sections

determined section of $\mathscr{R}^{(a)}(Q)$ from its β_{3-a}-situation on is congruent with the first determined section of $\mathscr{R}^{(3-a)}(Q)$ from its $\sigma^{(3-a)}=\beta_{3-a}$-situation on.

The assertions of Theorem 2 are well illustrated by the cyclical graph G_0 of Fig. 4 showing the possible characteristic situations of the first determined sections of ESs. The vertices of the graph represent situations and the (directed) arcs successions or identities. The arcs are labeled by critical decisions after critical situations and by conditions for Δ_a^* and the parameters after other vertices. The vertices framed by circles or squares can be the situations of $\mathscr{R}^{(1)}$ and $\mathscr{R}^{(2)}$, respectively, until the

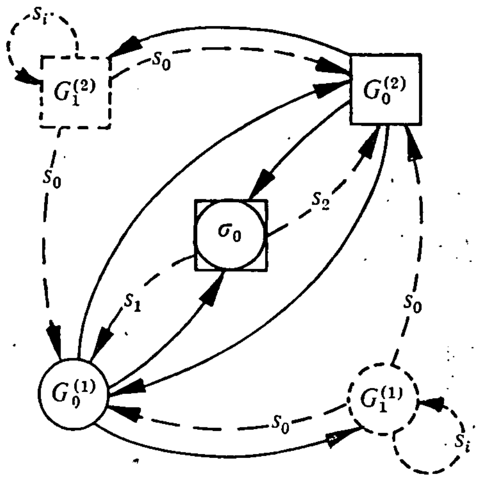


Fig. 5
The partitioning of the graph G_0

first typical situations. The graph G_0 represents all the possible cases for the whole configuration space $\mathcal{Q}$. For every $Q \in \mathcal{Q}$ only one arc going from a not critical situation is right. The graph can be partitioned into four subgraphs by Fig. 5. On the graphs the results of the decisions in the first critical situations are drawn by broken arcs.

Before we investigate further characteristic situations of the ESs, we show an example by Fig. 6. The part (a) shows the Gantt-chart of an $R \in \mathcal{R}^{(1)}(Q)$, the part (b) is the graph $G_0(Q)$ and the part (c) illustrates the graph $G(Q)$ of the ESs of $\mathcal{R}(Q)$.

EXAMPLE. $Q=(4.5;\ 3.5;\ 1;\ 2),\ \tau_1=8;\ \tau_2=3,\ \eta=5.5,\ \vartheta=5.5.$

$\omega_1^*=(1, 1),\ \Delta_1^*=5 \in (4.5;\ 5.5)$ and so $\sigma_1^*=\sigma_{1,1}$.

$\omega_2^*=(1, 0),\ \Delta_2^*=3 \in (1;\ 5.5)$ and so $\sigma_2^*=\sigma_{2,1}$.

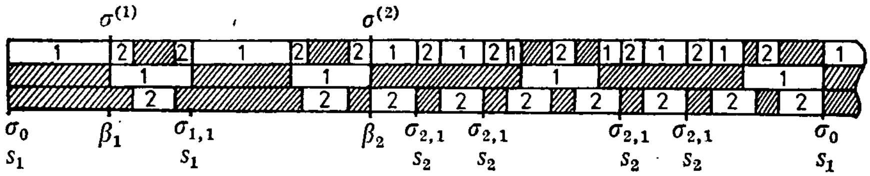
It is seen that always the characteristic situation $\sigma^{(3-a)} \in G_0$ occurs after the critical decision s_0 in a critical situation type $\sigma_{a,1}$. This means that new characteristic situation value can only be generated by decision s_i in a situation type $\sigma_{i,1}$. The type of the generated critical situation can be either of $\sigma_{j,1}$, $j=1, 2$, σ_0 and β_j, $j=1, 2$. The situations except type $\sigma_{j,1}$ are not new and lead back into the subgraph G_0. But the generated critical situation value must be new if its type is $\sigma_{j,1}$, $j=1, 2$. This is the consequence of the fact that determined sections are determined by their closing critical situations as well. Returning of an earlier $\sigma_{j,1}$ value after $\sigma_{i,1}$ would contradict this fact.

All the possibilities of the ES elements $R \in \mathcal{R}$ can well be illustrated by G_0 and the further critical situations according to the graph G on Fig. 7. The vertices $\sigma_{i,1}$ all illustrate different values of critical situations of type $\sigma_{1,1}$ and $\sigma_{2,1}$ independently of each other. The graph G is composed from five subgraphs by Fig. 7/b. $G_1^{(a)}$, $a=1, 2$, are the *branches* of G. The number of different vertices of G is infinite as we show below.

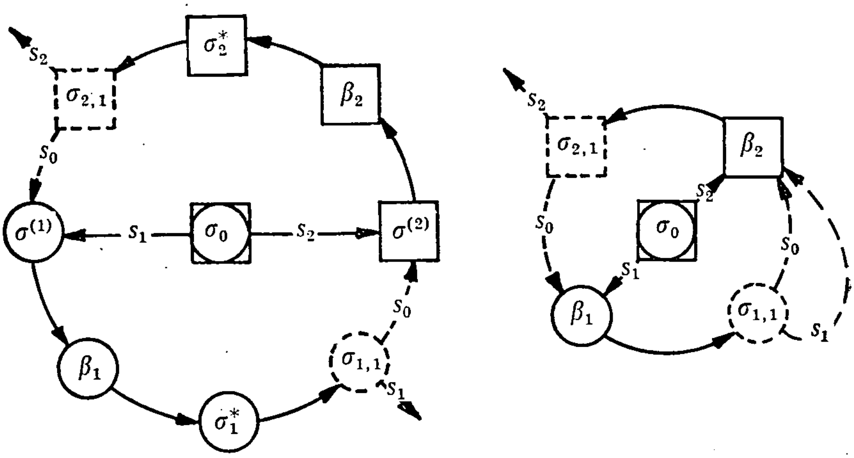
For any given configuration $Q \in \mathcal{Q}$ the elements $R \in \mathcal{R}(Q)$ can similarly be illustrated by a graph $G(Q)$ which is the subgraph of G (see Fig. 6/c). The dotted arcs on Fig. 7/a, b may be present only of a branch of $G(Q)$ is finite or missing. From the arcs going out from $G_0^{(a)}$ at most one can be present in any $G(Q)$. The number of vertices of $G(Q)$ can be infinite. Examples for infinity are the configurations with

$$\eta_a \vartheta_{3-a}=0,\ \vartheta_a \text{ and } \tau_{3-a} \text{ rationally independent} \tag{5}$$

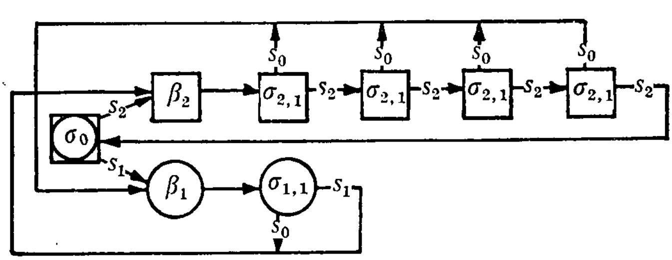
(see Fig. 8/b, c). The general conditions of the infinite vertices of $G(Q)$ is an open question. Perhaps, the above conditions are necessary.



(a) Gantt-chart



(b) Graphs $G_0(Q)$



(c) Graph $G(Q)$

Fig. 6

Graphical illustrations of the ESs for the configuration $Q=(4.5;\ 3.5;\ 1;\ 2)$

For any $Q\in\mathcal{Q}$ every $R\in\mathcal{R}(Q)$ can well be illustrated by a subgraph $G(R)$ of $G(Q)$. The configurations $Q\in\mathcal{Q}$ and the schedules $R\in\mathcal{R}(Q)$ can be classified e.g. by some significant characteristics of their graphs as well. Such characteristics can be the existence and number (one or two) of the branches $G_1^{(a)}(R)$, the finiteness, the number of loops in $G(R)$, etc. We will use some classifications below.

Let $R\in\mathcal{R}(Q)$ be an ES and $G(R)$ the graph representing it. $G(R)$ may have finite or infinite vertices. Let us call the *tour of R* the passage along the arcs and

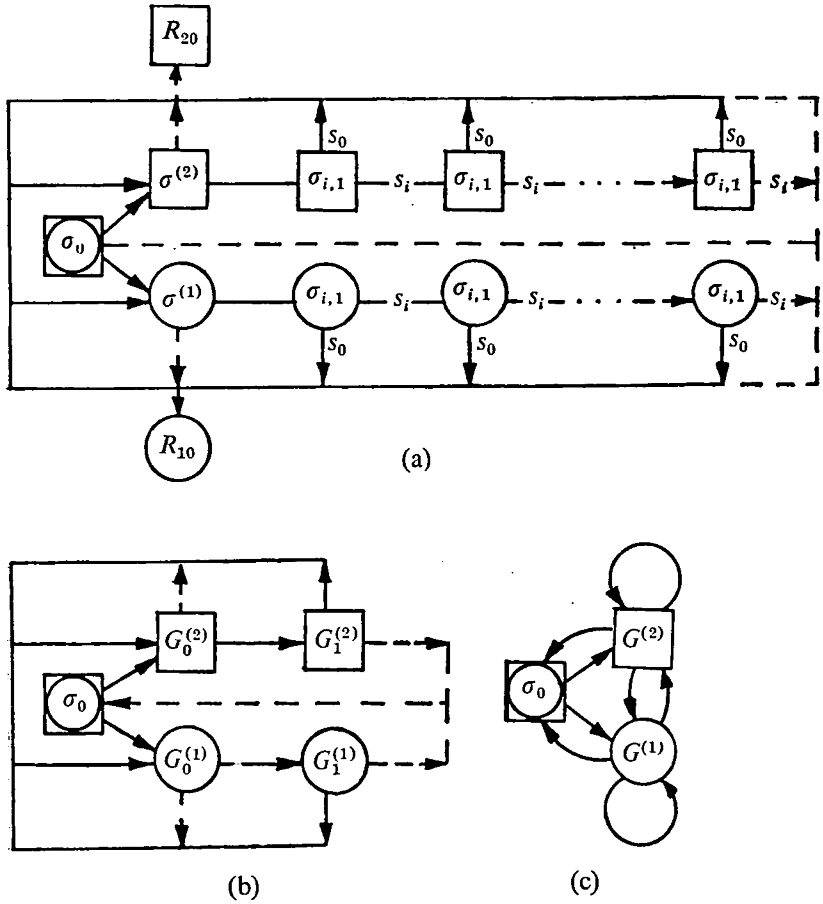


Fig. 7
The graph G of the elements of $\mathscr{R}$ and its partitions

vertices of $G(R)$ in accordance with all the characteristic situations of R. The passage of R may be finite ending in a vertex R_{i0} or infinite with finite or infinite number of loops. A *simple loop* in any graph is a loop having no other loops as its part. For any loop in $G(Q)$ there is at least one path from the vertex σ_0 to the loop without any other loop. The first vertex of the loop reached by the path from σ_0 to the loop is called a *root* of the loop.

For some reasons it may be necessary to allow demands of tasks to be zeros. The job-flow $Q^{(i)}$ is *defective* if one of η_i and ϑ_i is zero and is *degenerate* if both are zeros. For degenerate configurations (for which $\tau_1=0$ or $\tau_2=0$) we can impose specific restrictions to better model practical cases in which demands of one job-flow are negligible with respect to others. In such cases our methods could lead to optimal schedule not reasonable with regard to other optimal schedules. A re-

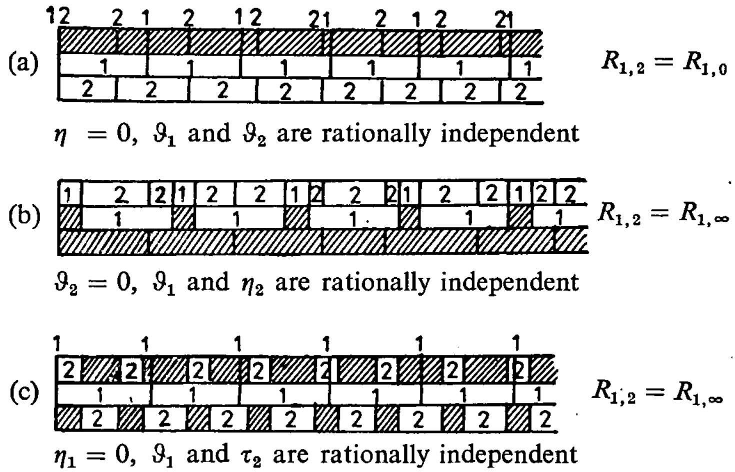


Fig. 8
Examples for CESs not periodic and having infinitely many different critical situation values

striction may be the prohibition of servicing repeatedly the cycles of the same degenerate job-flow alone [2, 3]. Such restrictions further complicate the discussion of the schedules. In degenerate cases the ESs are non-preemptive and are discussed in the course of non-preemptive scheduling of steady job-flow pairs [3].

4. Consistent economical schedules

After the preparations made in the previous paragraph, we are near to be able to prove our most important assertion: the class of consistent economical schedules is a dominant set.

An ES is a *consistent economical schedule* (CES) if its critical decisions are consistent: they are the same in every occurrence of the same critical situation values. Note that two situations of the same type, $\sigma_{i,1}$ say, may well have different values by having different values of $\beta^{(1)}(t)$ or $\beta^{(2)}(t)$, for instance. Let $\bar{\mathscr{R}}(Q) \subset \mathscr{R}(Q)$ be the class of CESs for Q and $\bar{\mathscr{R}} = \bigcup_{Q \in \bar{\mathscr{Q}}} \bar{\mathscr{R}}(Q)$.

The graphs $G(R')$ of CESs $R' \in \bar{\mathscr{R}}$ have specific characteristics. It can only have one out-arc from any vertex except the vertex R_{i0}, $i=1, 2$, if it is in $G(R')$. R_{i0} has no out-arc. Any vertex has only one in-arc except eventually the vertex σ_0 and one more. σ_0 has no in-arc if R_{i0} is in $G(R')$ or $G(R')$ is infinite. In case of a finite number of vertices and without R_{i0}, $G(R')$ has exactly one simple loop with root σ_0 if σ_0 has an in-arc or with another root which has two in-arcs then. The CES R' is said constructed from this loop. For any simple loop of $G(Q)$ there is at least one $G(R')$ composed from the loop and a path leading from σ_0 to the root of the loop. The tour of R' is the path from σ_0 to the root and infinitely many repetitions of the loop after. The efficiency of the CES so constructed is the P_A-

utilization of the constituent loop. This CES is *periodic* with *periods* represented by the loop. If $G(Q)$ is infinite, let $R_{a,\infty}$ denote the CES with a tour from σ_0 through $\sigma^{(a)}$ and vertices $\sigma_{i,1}$ to the infinity without any loop.

Theorem 3. *The class $\bar{\mathcal{R}}$ of the consistent economical schedules is a dominant set.*

Proof. Let $R\in\mathcal{R}$ be any ES with efficiency $\gamma(R)$. We will show a CES $R'\in\bar{\mathcal{R}}$ dominating R. The dominance follows if R is CES or is essentially-the-same as a CES R'.

If the graph $G(Q)$ does not have loops, all ESs are consistent and R may not be other as well. If the P_A-utilizations of the simple loops of $G(Q)$ have a maximum, the R' constructed from a simple loop with maximal P_A-utilization will dominate every other ESs except eventually those which are essentially-the-same as R_{i0} or $R_{i,\infty}$, $i=1, 2$.

The only crucial $G(Q)$ is that in which the P_A-utilizations of simple loops have no maximum. But if the $G(R)\subset G(Q)$ has a simple loop with P_A-utilization not less than $\gamma(R)$, the CES R' constructed from this loop will dominate R. Thus the dominatedness of R with finite $G(R)$ by CESs is proved. If $G(R)$ is infinite but with a finite number of simple loops, the tour of R cannot have a loop after a finite initial section and is essentially-the-same as an $R_{i,\infty}$.

The only crucial $G(R)$ is, therefore, that which has infinitely many simple loops without one having maximum P_A-utilization. Whether such a $G(R)$ does or does not exist is an open but irrelevant question now. The length of loops cannot be bounded in this case. The schedule R is composed from two kinds of simple loops represented by Fig. 9.

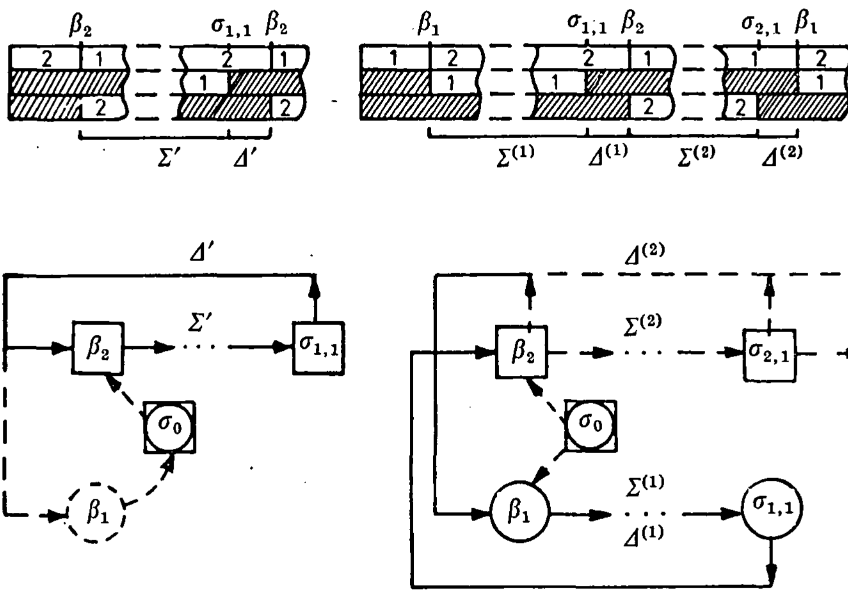


Fig. 9
The two possibilities of simple loops

By definition (1) of $\gamma(R)$ we can choose a sequence $\Sigma_1, \Sigma_2, \ldots, \Sigma_n, \ldots$ of initial sections of R which are ending with simple loops and for which

$$\gamma(R) = \lim_{n\to\infty} \frac{\lambda(\Sigma_n)}{t(\Sigma_n)}$$

where $\lambda(\Sigma_n)$ and $t(\Sigma_n)$ are the P_A-usage and length, respectively, of the section Σ_n. But

$$\gamma(\Sigma_n) = \frac{\lambda(\Sigma_n)}{t(\Sigma_n)}$$

is the weighted mean of the P_A-utilizations of the finite many simple loops composing Σ_n. Let $\Delta\Sigma_1, \Delta\Sigma_2, \ldots$ a sequence of simple loops carved out of $\Sigma_1, \Sigma_2, \ldots$, respectively, with maximal P_A-utilizations. By assumptions

$$\gamma(\Sigma_n) \leq \gamma(\Delta\Sigma_n) < \gamma(R)$$

and so the convergence $\gamma(\Delta\Sigma_n)\to\gamma(R)$ is true. The sequence $\Delta\Sigma_1, \Delta\Sigma_2, \ldots$ must have a subsequence with monotonically increasing length and P_A-utilization because the contrary would lead to contradiction with the assumptions $\gamma(\Delta\Sigma_n)\to\gamma(R)$ and no finite loop with $\gamma(\Delta\Sigma_n)\geqq\gamma(R)$ exists. Let $\Delta\Sigma_1, \Delta\Sigma_2, \ldots$ be this subsequence already. Clearly $\gamma(\Delta\Sigma_n)\to\gamma(R)$. Every $\Delta\Sigma_n$ could be composed either from an initial section Σ_n' of an $R_{i,\infty}$, $i=1, 2$, and a section Δ_n' of bounded length or from an initial section $\Sigma_n^{(1)}$ of $R_{1,\infty}$, an initial section $\Sigma_n^{(2)}$ of $R_{2,\infty}$, a section $\Delta_n^{(1)}$ and a section $\Delta_n^{(2)}$ of bounded lengths, as in Fig. 9. Because of boundedness of sections Δ_n', $\Delta_n^{(1)}$ and $\Delta_n^{(2)}$ they do not influence the limit of $\gamma(\Delta\Sigma_n)$ and

$$\lim_{n\to\infty} \gamma(\Delta\Sigma_n) = \lim_{n\to\infty} \gamma(\Sigma_n^{(1)} \cup \Sigma_n^{(2)})$$

allowing one of $\Sigma_n^{(1)}$ and $\Sigma_n^{(2)}$ to be missing. In the sequence $\Delta\Sigma_1, \Delta\Sigma_2, \ldots$ at least one of $\Sigma_n^{(1)}$ and $\Sigma_n^{(2)}$ tends to $R_{1,\infty}$ or $R_{2,\infty}$, respectively. $\gamma(\Delta\Sigma_n)$ cannot be greater in limit than the maximum of limits of $\gamma(\Sigma_n^{(1)})$ and $\gamma(\Sigma_n^{(2)})$. Therefore, the maximum of $\gamma(R_{1,\infty})$ and $\gamma(R_{2,\infty})$ will not be less than $\gamma(R)$ and the corresponding CES $R_{i,\infty}$ dominates R. This concludes our proof. □

The set $\bar{\mathscr{R}}(Q)$ of CESs can have fairly many — if not infinite — elements in general. Methods for reducing further the dominant set or a simple algorithm to choose an optimal schedule from $\mathscr{R}(Q)$ are not known. A direct method to determine the optimal schedule is to survey the whole set $\bar{\mathscr{R}}$ and compare the efficiencies of the elements. In some cases this is a feasible arrangement. To judge better the amount of work on this way we can use the *number* $N_L(Q)$ *of simple loops* in $G(Q)$ and the *number* $\bar{N}(Q)$ *of elements* of $\bar{\mathscr{R}}(Q)$. To determine these we need the graph $G(Q)$ or at least some data of it.

Let us define the following data (see Fig. 6 and Fig. 7 as illustration):

$$\begin{aligned} &n_0 \text{ is the number of } R_{i0} \text{ vertices in } G(Q) \\ &n_{aj} \text{ is the number of vertices } \sigma_{j,1} \text{ of the branch } G_1^{(a)}(Q) \end{aligned} \tag{6}$$

for $a = 1, 2, \quad j = 1, 2$

$$\delta_{aj} = \begin{cases} 1 & \text{if the last arc of } G^{(a)} \text{ leads to vertex } \sigma^{(j)} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

for $a = 1, 2 \quad j = 0, 1, 2$ and $\sigma^{(0)} = \sigma_0$.

Use the notations

$$n_a = n_{a1} + n_{a2}, \quad a = 1, 2. \tag{8}$$

n_a is the number of vertices in the branch $G_1^{(a)}(Q)$. All the data can be read from two schedule-sections $\Sigma^{(a)}$, $a=1, 2$, constructed in the following way. For $\Sigma^{(a)}$ schedule Q economically with critical decisions $s(0)=s_a$ and $s(\sigma_{i,1})=s_i$, $i=1, 2$, until the first typical situation other than $\sigma_{i,1}$ occurs. This procedure is finite iff $G(Q)$ is finite. From these two schedule-sections we can read the P_A-usages $\lambda(\Delta\Sigma)$ and lengths $t(\Delta\Sigma)$ of determined sections $\Delta\Sigma$ which are necessary to evaluate the CESs of Q. These two schedule-sections enable us to draw simply the graph $G(Q)$ and determine the data (6)—(8). To illustrate this method, Fig. 12 below can be considered. The way to use the data to determine $N_L(Q)$ and $\bar{N}(Q)$ is stated by the following lemma.

Lemma 1. *The number N_L of the simple loops of $G(Q)$ and the number $\bar{N}$ of the elements of $\bar{\mathcal{R}}(Q)$ can be expressed as*

$$N_L = (n_{11}+\delta_{10}+\delta_{12})(n_{22}+\delta_{20}+\delta_{21})+(n_{12}+\delta_{10}+\delta_{11})+(n_{21}+\delta_{20}+\delta_{22})-\delta_{10}\delta_{20} \tag{9}$$

$$\bar{N} = (n_{11}+\delta_{12})(n_2+\delta_{20}+\delta_{21}+\delta_{22})+(n_{22}+\delta_{21})(n_1+\delta_{10}+\delta_{11}+\delta_{12})+ \\ +(n_{12}+\delta_{10}+\delta_{11})+(n_{21}+\delta_{20}+\delta_{22})+n_0 \tag{10}$$

where n_j, n_{aj} and δ_{aj} are defined by (6)—(8).

Proof. Consider Fig. 7 as illustration. We count the number of simple loops of the graph $G(Q)$ and the number of different paths from σ_0 to the loop without other loops.

The number $N_L^{(aa)}$ of loops not leading out from the subgraph $G^{(a)}$ is the number of vertices $\sigma_{3-a,1}$ plus one if the last arc of $G^{(a)}$ leads to the vertex $\sigma^{(a)}$. This gives $N_L^{(aa)}=n_{a,3-a}+\delta_{aa}$. The root $\sigma^{(a)}$ of these loops can be reached directly from σ_0 or through $\sigma^{(3-a)}$ if arcs connect $G^{(3-a)}$ to $\sigma^{(a)}$. The number of the latter arcs is the number of vertices $\sigma_{a,1}$ in $G^{(3-a)}$ plus one if the last arc of $G^{(3-a)}$ leads to $\sigma^{(a)}$. This gives the number of paths from σ_0 to $\sigma^{(a)}$ as $1+n_{3-a,3-a}+\delta_{3-a,a}$ and the number $\bar{N}^{(aa)}$ of the CESs as $\bar{N}^{(aa)}=(n_{a,3-a}+\delta_{aa})(1+n_{3-a,3-a}+\delta_{3-a,a})$. Further loops arise from arcs leading from $G^{(1)}$ to $\sigma^{(2)}$ and back from $G^{(2)}$ to $\sigma^{(1)}$. The number of arcs leading from $G^{(a)}$ to $\sigma^{(3-a)}$ is the number of vertices $\sigma_{a,1}$ in the branch $G_1^{(a)}$ plus one if the last arc of $G^{(a)}$ leads to $\sigma^{(3-a)}$ as well. This gives the number $N_L^{(0)}$ of simple loops as $N_L^{(0)}=(n_{11}+\delta_{12})(n_{22}+\delta_{21})$. Any of these loops can be reached directly through $\sigma^{(1)}$ or $\sigma^{(2)}$ giving the number of CESs as $\bar{N}^{(0)}=2(n_{11}+\delta_{12})(n_{22}+\delta_{21})$. There are loops between σ_0 and $G^{(a)}$ if the last arc in $G^{(a)}$ leads to σ_0. Because the vertex σ_0 is the component of the loop, one or other of the paths $\sigma_0\to\sigma^{(1)}$ and $\sigma_0\to\sigma^{(2)}$ is an arc of the loop and determine the possible loops. The arc $\sigma_0\to\sigma^{(a)}$ is the part of only one loop if $\delta_{a0}=1$. The arc $\sigma_0\to\sigma^{(3-a)}$ is the part of loops

$\sigma_0 \to \sigma^{(3-a)} \to \sigma_{3-a,3-a} \to \sigma^{(a)} \to \sigma_0$ the number of which is $n_{3-a,3-a}+\delta_{3-a,a}$. These give the number of loops $N_L^{(a)}=(1+n_{3-a,3-a}+\delta_{3-a,a})\delta_{a0}$. Each loop is the constituent of exactly one CES and this fact gives the number $\bar{N}^{(a)}=(1+n_{3-a,3-a}+\delta_{3-a,a})\delta_{a0}$.

Adding up the numbers for $a=1$ and $a=2$, we have

$$N_L = N_L^{(11)}+N_L^{(22)}+N_L^{(0)}+N_L^{(1)}+N_L^{(2)} =$$

$$= n_{12}+\delta_{11}+n_{21}+\delta_{22}+(n_{11}+\delta_{12})(n_{22}+\delta_{21})+(1+n_{11}+\delta_{12})\delta_{20}+(1+n_{22}+\delta_{21})\delta_{10}$$

and

$$\bar{N}'=\bar{N}^{(11)}+\bar{N}^{(22)}+\bar{N}^{(0)}+\bar{N}^{(1)}+\bar{N}^{(2)}=(n_{12}+\delta_{11})(1+n_{22}+\delta_{21})+$$

$$+(n_{21}+\delta_{22})(1+n_{11}+\delta_{12})+2(n_{11}+\delta_{12})(n_{22}+\delta_{21})+(1+n_{11}+\delta_{12})\delta_{20}+(1+n_{22}+\delta_{21})\delta_{10}.$$

If $G_0^{(a)}(Q)$ contains the vertex R_{a0}, the subgraph $G_0^{(a)}$ in Fig. 7/b has no out-arc and cannot take part in any cycle but represents a CES the path of which ends in vertex R_{a0}. This means that the value $\bar{N}'$ obtained above must be corrected by adding n_0 to the number of CESs generated by loops. The identity of the so obtained expressions of N_L and $\bar{N}'+n_0$ with (9) and (10) is obvious. □

For the *example* of Fig. 6 we get

$$n_{11}=1,\quad n_{12}=0,\quad \delta_{10}=0,\quad \delta_{11}=0,\quad \delta_{12}=1$$

$$n_{21}=0,\quad n_{22}=4,\quad \delta_{20}=1,\quad \delta_{21}=0,\quad \delta_{22}=0.$$

From these data the numbers are

$$N_L=11 \quad \text{and} \quad \bar{N}=19.$$

If $G(Q)$ has no branches, i.e. $n_{ai}=0, a=1,2, i=1,2,$ then the particular formulae are

$$N_L(Q)=(\delta_{10}+\delta_{12})(\delta_{20}+\delta_{21})+(\delta_{10}+\delta_{11})+(\delta_{20}+\delta_{22})-\delta_{10}\delta_{20} \leqq 2 \qquad (9')$$

$$\bar{N}(Q)\equiv 2. \qquad (10')$$

The relations can be proved simply by taking the possible values of n_0 and every δ_{aj}.

The CESs having the same simple loop as their constituent (period) are essentially-the-same. The number of essentially different CESs is N_L and $\bar{\mathscr{R}}(Q)$ represents at most N_L different efficiency values.

Except the trivial cases of existence of a vertex R_{a0} in $G_0(Q)$ — which can only be in the defective cases (3) and (4) — the relations

$$\delta_{a0}+\delta_{a1}+\delta_{a2}=1,\quad a=1,2,\quad n_0=0 \qquad (11)$$

are always true and the expressions (9) and (10) can be written in the simpler forms

$$N_L=(n_{11}+1-\delta_{11})(n_{22}+1-\delta_{22})+(n_{12}+1-\delta_{12})+(n_{21}+1-\delta_{21})-\delta_{10}\delta_{20} \qquad (9'')$$

$$\bar{N}=(n_{11}+\delta_{12})(n_2+1)+(n_{22}+\delta_{21})(n_1+1)+(n_{12}+1-\delta_{12})+(n_{21}+1-\delta_{21}). \qquad (10'')$$

The expressions (9) and (9″) show how the number N_L of the possible CESs representing different values of efficiency depends on the numbers n_{aj}, $a, j=1,2,$

of the vertices in the branches of $G(Q)$. N_L is finite if all n_{aj} are finite and if $n_{ii}=\infty$ but $n_{i,3-i}, n_{3-i,i}$ are finite and $n_{3-i,3-i}+\delta_{3-i,0}+\delta_{3-i,i}=0$ (provided that this last case is possible for some configuration Q).

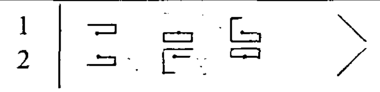
For the sake of reference, we have to identify the elements of $\bar{\mathscr{R}}(Q)$. In view of evaluation, the identification of the simple loops is enough. We introduce a symbolism for this purpose.

We identify the vertices of the branches $G_1^{(a)}$, $a=1, 2$, by numbering them serially with $1, 2, \ldots, n_a$ in the order of occurrences in $G_1^{(a)}$. Let the vertex $\sigma^{(a)}$ have the serial number 0 and the vertex of $G(Q)$ the last arc of $G^{(a)}(Q)$ leads to the serial number n_a+1. This last vertex can be either σ_0 or $\sigma^{(1)}$ or $\sigma^{(2)}$. The serial numbers of vertices of $G^{(1)}$ and $G^{(2)}$ of our example in Fig. 6 will be 0, 1, 2 and 0, 1, 2, 3, 4, 5, respectively. The last number of $G^{(1)}$ represents the vertex $\sigma^{(2)}$ and the last number of $G^{(2)}$ represents the vertex σ_0. Every simple loop is composed from one or two sections belonging to subgraphs $G^{(1)}$ and $G^{(2)}$, respectively. Every *loop-section* of $G^{(a)}$ starts with the vertex $\sigma^{(a)}$, goes through some further vertices of $G_1^{(a)}$ if they exist, and finishes in σ_0, $\sigma^{(1)}$ or $\sigma^{(2)}$. A loop-section of a given $G^{(a)}(Q)$ can be identified by the maximum of serial numbers of its vertices. The character of a loop-section can well be given by a code (abc) constructed from the number "a" of the subgraph it belongs to, from the maximal serial number "b" of its vertices and from the code "c" of its last vertex by the coding:

type	σ_0	$\sigma^{(1)}$	$\sigma^{(2)}$
c-code	0	1	2

The code (ac) identifies the shape of the loop-section which can be symbolized in the following way:

a \ c	0	1	2	$\sigma_0 \to \sigma^{(a)}$



The simple loops are composed from one or two sections directly or by means of a section $\sigma_0 \to \sigma^{(1)}$ or $\sigma_0 \to \sigma^{(2)}$ symbolized by $\diagdown$ and $\diagup$.

To identify a simple loop we can use the b-codes of its component loop-sections. The loop identified with $(b_1 b_2)$ has vertices from $G^{(1)}$ and $G^{(2)}$ with maximum serial number b_1 and b_2, respectively. If a loop has no vertex from $G^{(a)}$, the component b_a is zero.

The elements R of $\bar{\mathscr{R}}$ can be characterized by the code $(b_1 b_2)$ of its simple loop. The CESs R_{a0} for degenerate configurations (3) and (4) will be characterized by the code (00). The code $(b_1 b_2)$ of a CES is called its *type*. The code $(b_1 b_2)$ represents an essentially-the-same class of $\bar{\mathscr{R}}(Q)$, the number of which was counted in the proof of Lemma 1.

Not every code $(b_1 b_2)$ can represent an existing loop in $G(Q)$. In Table 1 we marked by sign + or — that a loop of code $(b_1 b_2)$ composed from the existing loop-section pair $(1b_1c_1)$, $(2b_2c_2)$ did or did not exist, respectively. The code (00) is possible if at least one vertex R_{a0} of $G(Q)$ exists (and $n_a=0$, of course). In this case the only possible value of b_a is 0. The other (b_1b_2) entries of Table 1 for given (abc) codes can be easily made. We put sign − in every entries of rows with $c_1=1$ and of columns with $c_2=2$ except their first entries. In row $b_1=0$ we put − in entries with heading $c_2=1$ and in column $b_2=0$ we put − in entries with heading $c_1=2$. If an entry with $c_1=c_2=0$ existed, we put − in it. In the remaining entries we put signs +.

Table 1. The existing codes (b_1b_2) of simple loops.

b_1 \ b_2	c_1 \ c_2	0 (R_{20})	b_2: 2	b_2: 1	n_2+1: 0	n_2+1: 2	n_2+1: 1
0	(R_{10})	(+)	+	−	+	+	−
⋮							
b_1	1	+	−	−	−	−	−
	2	−	−	+	+	−	+
⋮							
n_1+1	0	+	−	+	−	−	+
	1	+	−	−	−	−	−
	2	−	−	+	+	−	+

Example of Fig. 6

b_1 \ b_2	c_1 \ c_2	0 (−)	1 (1)	2 (1)	3 (1)	4 (1)	5 (0)
0	−	−	−	−	−	−	⊕
1	2	−	⊕	+	+	+	+
2	2	−	⊕	+	+	+	+

Table 1 says which loops have to be evaluated for determining the optimal one. The possibilities for some specific types of CESs are represented by Fig. 10/a.

The set $\bar{\mathscr{R}}(Q)$ always contains exactly two non-preemptive schedules $R_{a,0}$, $a=1, 2$, which are the two tight consistent natural schedules defined in [4]. These are the *non-preemptive priority schedules,* at the same time [2]. Two other remarkable elements of $\bar{\mathscr{R}}(Q)$ are the *priority schedules* $R_{a,3-a}$, $a=1, 2$. $R_{a,3-a}$ is defined as the CES in which the job-flow $Q^{(a)}$ has absolute priority against $Q^{(3-a)}$ which means that every task A_{aj}, $j=1, 2, \ldots,$ is serviced by P_A at the moment it is ready for service, independently of the state of P_A. The priority schedules are schedules of great practical importance. With the help of Table 1 it is easy to determine the types $(b_1'b_2')$ of the priority schedules $R_{a,0}$ and $R_{a,3-a}$, $a=1, 2$, by their definitions.

$R_{a,0}$ is determined by the restriction that no preemption is allowed and $s(0)=s_a$. This means that $R_{a,0}=R_{a0}$ and has type (00) if the vertex R_{a0} exists. Otherwise, $b_a'=1$, $b_{3-a}'=0$ except if $c_{a,1}=3-a$ when $b_{3-a}'=1$, and $c_{3-a,1}=c_{a,1}=3-a$ when $b_a'=0$, moreover.

$R_{a,3-a}$ is determined by the fact that any task type A_{3-a} must and any task type A_a must not be preempted in conflicting situations $\sigma_{i,1}$, $i=1, 2$. This means that $s(\sigma_{a,1})=s_a$ and $s(\sigma_{3-a,1})=s_0$. The possibilities are illustrated by Fig. 10/b. If the vertex R_{a0} exists, then $R_{a,3-a}=R_{a,0}=R_{a0}$ with type (00). Otherwise, b_a' cf

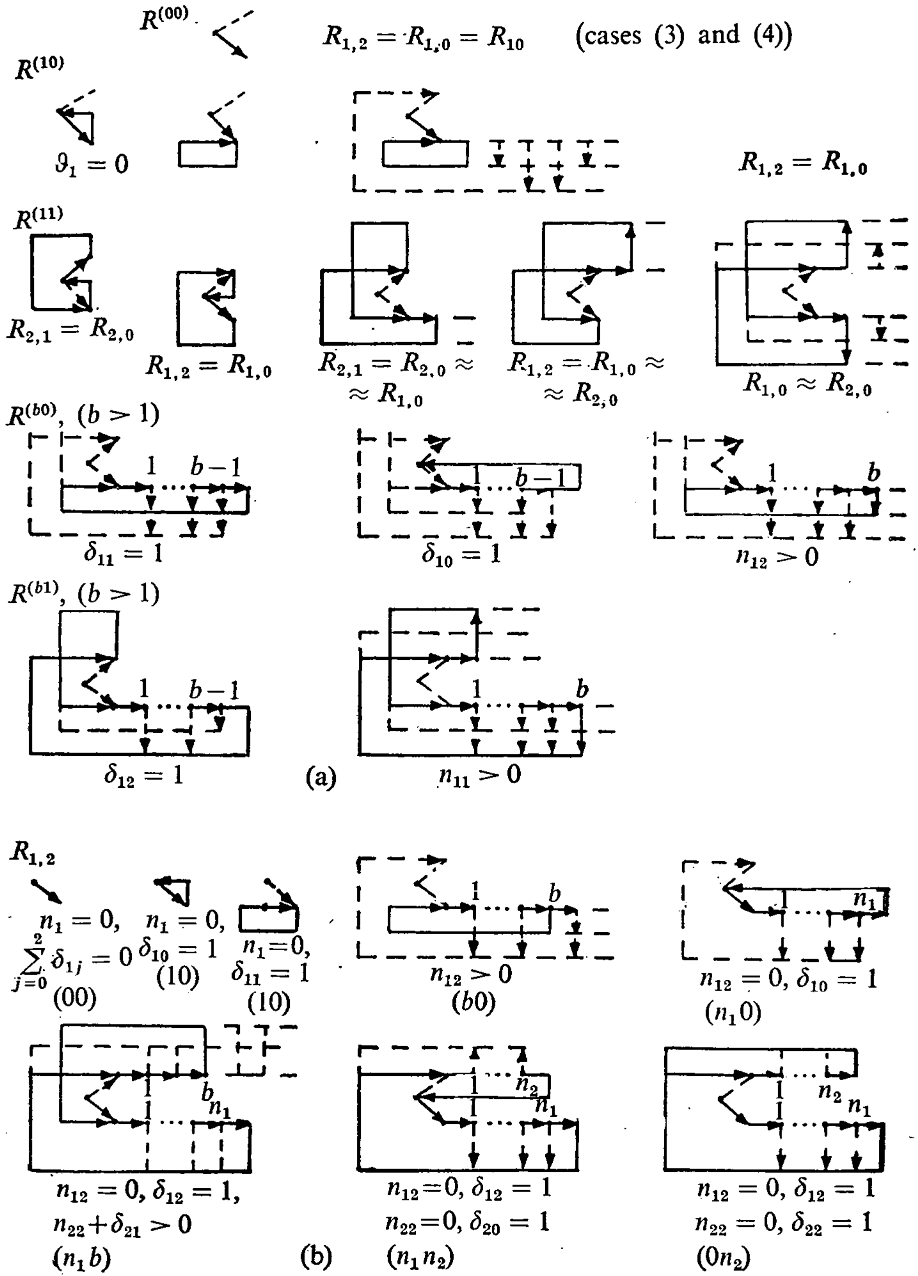


Fig. 10

Special types $R^{(b_1 b_2)}$ and types of $R_{1,2}$

$R_{a,3-a}$ is the serial number of the first vertex type $\sigma_{3-a,1}$ in the branch $G_1^{(a)}$, if it exists ($n_{a,3-a}>0$) and $b'_a=n_a+1$ or $b'_a=0$ otherwise (when $n_{a,3-a}=0$). $b'_a=n_a+1$ if $\delta_{a,3-a}=0$ or $\delta_{a,3-a}=1$ and $n_{3-a,3-a}+1-\delta_{3-a,3-a}>0$. $b'_a=0$ if $\delta_{a,3-a}=1$ and $n_{3-a,3-a}+1-\delta_{3-a,3-a}=0$. The value of b'_{3-a} of $R_{a,3-a}$ is 0 when $n_{a,3-a}+$ $+1-\delta_{a,3-a}>0$, the serial number of the first vertex $\delta_{3-a,1}$ in the branch $G_1^{(3-a)}$, if it exists ($n_{3-a,3-a}>0$) and $n_{3-a}+1$, otherwise, when $n_{a,3-a}+1-\delta_{a,3-a}=0$.

In the completed Table 1 we can pick out the types of $R_{a,0}$ and $R_{a,3-a}$ as follows. $R_{a,0}$ is represented by the sign + encounters first in counter-clockwise for $a=1$ and clockwise for $a=2$ in the left upper 2×2 subtable and $R_{a,3-a}$ is represented by the first + encounters on the border of the whole table counter-clockwise for $a=1$ and clockwise for $a=2$ starting from the entry (00). If Table 1 consists only from one row then $R_{1,0}=R_{1,2}=R_{10}$ and if it consists only from one column then $R_{2,0}=R_{2,1}=R_{20}$.

Let $\mathscr{R}_0(Q)=\{R_{1,2}, R_{2,1}\}$ be the pair of priority schedules. This is a subset of $\bar{\mathscr{R}}(Q)$. If $\bar{\mathscr{R}}(Q)=\mathscr{R}_0(Q)$ then $\mathscr{R}_0(Q)$ is a dominant set. In this case $R_{a,3-a}=R_{a,0}$, $a=1,2$. An example for this is the configuration $Q=(1;\ 4;\ 2;\ 5)$ with $R_{1,2}(Q)$ optimal. If $\bar{\mathscr{R}}(Q)\neq\mathscr{R}_0(Q)$, the set $\mathscr{R}_0(Q)$ is not necessarily dominant. Trivial examples for this are the configurations Q with $\vartheta_i<\eta_{3-i}<2\vartheta_i$, $i=1,2$. For these configurations the CESs $R_{1,0}\approx R_{2,0}$ are optimal with efficiency $\gamma=1$. A non-trivial example is the configuration $Q=(4.5;\ 3.5;\ 1;\ 2)$ in Fig. 6 as we will see in the next paragraph.

Though the priority schedules are not dominant, they are interesting on their own, because they are often used in practice and can be produced by simple rules. They are investigated in the study [2]. The evaluation of $R_{1,2}$ and $R_{2,1}$ is not a trivial task at all. The priority schedules were investigated also for the stochastic version of job-flow pairs [1, 5].

5. Evaluation of the CESs

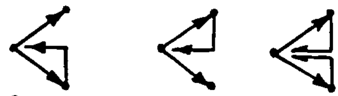
Though the cardinal of the dominant set $\bar{\mathscr{R}}(Q)$ of the consistent economical schedules is not necessarily finite, we give an algorithm for the direct evaluation of the CESs. This is applicable only when $\bar{\mathscr{R}}(Q)$ is finite. $\bar{\mathscr{R}}(Q)$ is finite exactly then when the graph $G(Q)$ is finite. For some cases the automatic application of the given algorithm can be superfluously complicate. Four such cases will be mentioned below as cases (i)—(iv). These cases contain the configurations we know as having $G(Q)$ with infinite vertices. By *general case* non-defective configurations are meant. The special cases (i)—(iv) are illustrated by Fig. 11.

Case (i). $\tau_1\tau_2=0$, degenerate configurations (see (4)). The CESs are the $R_{a,0}$, $a=1,2$, and $\gamma_{a,0}=0$. If the number of cycles of the same degenerate job-flow scheduled directly after each other is restricted, the maximal efficiency $\gamma^{(1)}+\gamma^{(2)}$ can be achieved.

Case (ii). $\eta=0$, $\vartheta_1\vartheta_2>0$. $R_{a,0}$, $a=1,2$, are the only CESs with $\gamma=0$. $R_{a,0}=R_{a0}$ and has no typical situations for the configurations (3).

Case (iii). $\tau_1\tau_2>0$, $\eta>0$ but Q is defective. If $\eta_a\vartheta_{3-a}=0$ then $R_{a,3-a}$ has the maximum efficiency of $\gamma=\gamma^{(3-a)}$ (see Fig. 8). The shape of the graph $G(Q)$ depends

Case (i). $\tau_1\tau_2 = 0$

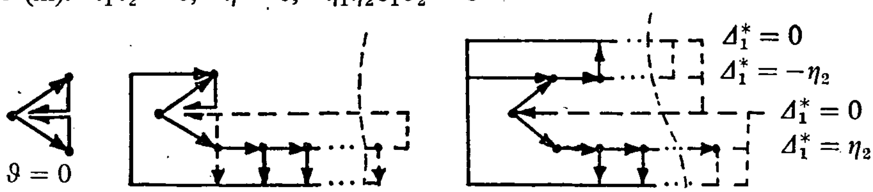


$\vartheta_2 > 0$ $\quad\vartheta_1 > 0$ $\quad\vartheta = 0$.

Case (ii). $\eta = 0, \quad \vartheta_1\vartheta_2 > 0$

$\frac{\vartheta_1}{\vartheta_2}$ irrational $\quad\frac{\vartheta_1}{\vartheta_2}$ rational

Case (iii). $\tau_1\tau_2 > 0, \quad \eta > 0, \quad \eta_1\eta_2\vartheta_1\vartheta_2 = 0$



$\vartheta_1 > 0, \vartheta_2 = 0, \frac{\vartheta_1}{\vartheta_2}$ rational $\quad \eta_1 = 0, \vartheta_2 > 0,$ solution of $(*)$ exists

Case (iv). $\eta_1 > \vartheta_2 > 0, \quad \eta_2 > \vartheta_1 > 0$

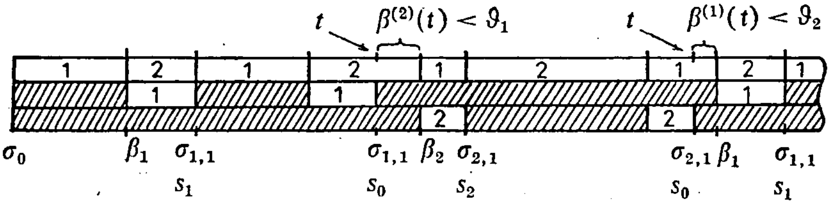


Fig. 11
Trivial cases for optimal schedule

on the existence and relations of the least non-trivial non-negative integer solutions (X_1^*, X_2^*) of the equations

$$\Delta_a \equiv X_a\vartheta_a - X_{3-a}\tau_{3-a} = \begin{cases} 0 \\ \pm\eta_{3-a} \end{cases} \quad a = 1, 2 \qquad (*)$$

but this fact is irrelevant from the point of view of optimality. There is no solution oj $(*)$ in cases (5).

Case (iv). $\eta_i \geqq \vartheta_{3-a} > 0$, $i = 1, 2$. The maximal efficiency of the CESs is $\gamma = 1$ and any $R \in \bar{\mathscr{R}}(Q)$ with decisions $s(\sigma_{i,1}) = s_0$ if only $\beta^{(3-i)}(t) - \vartheta_{3-i} < \vartheta_i$, is optimal. E.g. also the $R_{a,0}$, $a = 1, 2$, are optimal with $\gamma_{a,0} = 1$.

Before we give an algorithm for the general case, we show the evaluation of the CESs of the example configuration $Q = (4.5;\ 3.5;\ 1;\ 2)$.

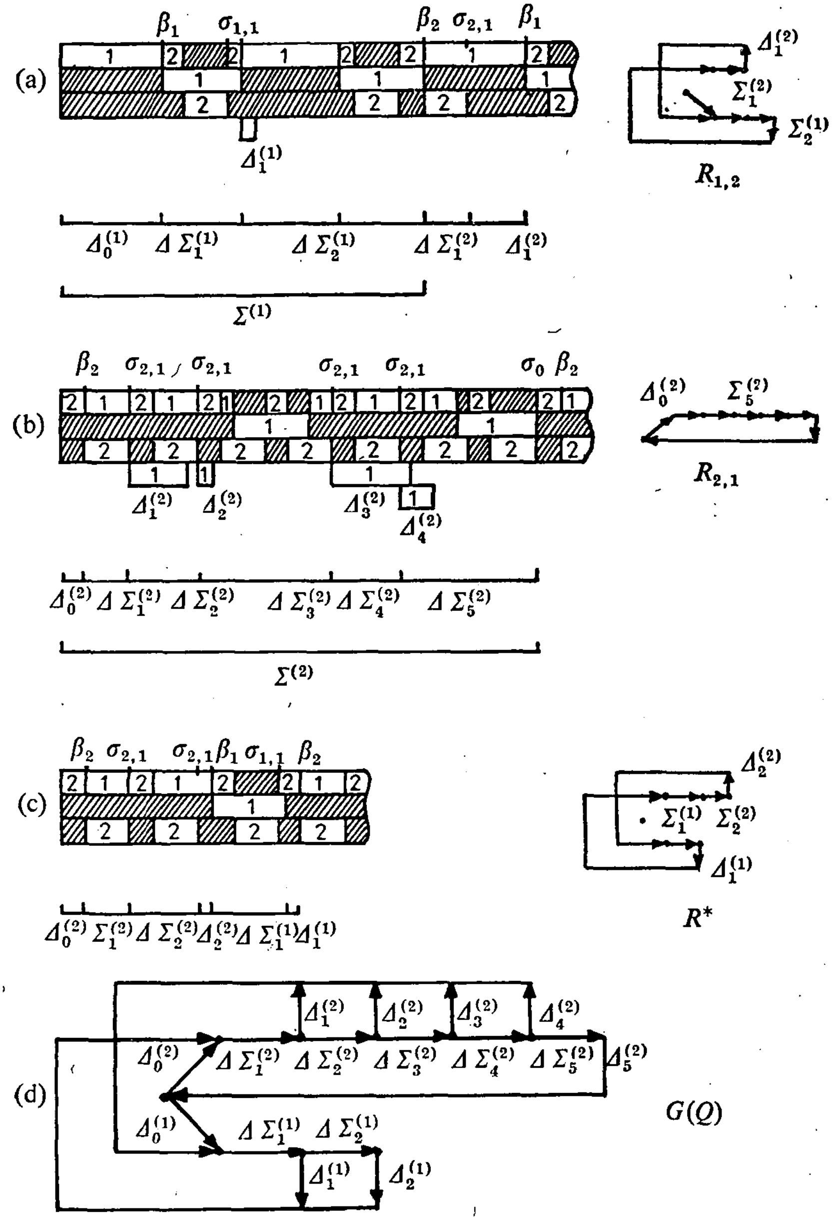


Fig. 12

The sections $\Sigma^{(a)}$, $a=1,2$, the priority and the optimal schedules of the example $Q=(4.5;\ 3.5;\ 1;\ 2)$

In Fig. 12/a, b we show the Gantt-charts of the two schedule sections $\Sigma^{(1)}$ and $\Sigma^{(2)}$ expanded here to provide $R_{1,2}$ and $R_{2,1}$ at the same time. It can be realized that every loop-section is composed from consecutive subsections $\Delta\Sigma_j^{(a)}, j=1, \dots, b$, of $\Sigma^{(a)}$ and a section $\Delta_b^{(a)}$ of full P_A-utilization as $\Sigma_b^{(a)} \cup \Delta_b^{(a)}$. This fact is illustrated by Fig. 12/d. The lengths and P_A-usages of the subsections can be read from $\Sigma^{(1)}$ and $\Sigma^{(2)}$ and are given in Table 2. The data (lengths and P_A-usages) of loop-sections are

$$t(\Sigma_b^{(a)})+\Delta_b^{(a)} \quad \text{and} \quad \lambda(\Sigma_b^{(a)})+\Delta_b^{(a)}$$

with

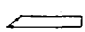
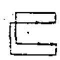
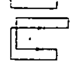
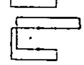
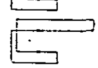
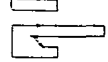
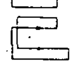
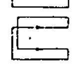
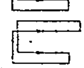
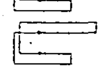
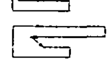
$$\Sigma_b^{(a)} = \bigcup_{j=1}^{b} \Delta\Sigma_j^{(a)}, \quad b = 1, 2, \dots, n_a+1, \quad a = 1, 2.$$

These data are given in Table 2 as well.

Table 2. The data of loop-sections of the example of Fig. 12

a	b	Type of sect.	c	$\lambda(\Delta\Sigma)$	$t(\Delta\Sigma)$	Δ	$\lambda(\Sigma+\Delta)$	$t(\Sigma+\Delta)$
	0	$\sigma^{(1)}$	—	—	—	4.5	—	—
1	1	$\sigma_{1,1}$	2	1.5	3.5	0.5	2	4
	2	$\sigma^{(2)}$	2	6	8	0	7.6	11.5
	0	$\sigma^{(2)}$	—	—	—	1	—	—
	1	$\sigma_{2,1}$	1	2	2	2.5	4.5	4.5
2	2	$\sigma_{2,1}$	1	3	3	0.5	5.5	5.5
	3	$\sigma_{2,1}$	1	3.5	6	3.5	12	14.5
	4	$\sigma_{2,1}$	1	3	3	1.5	13	15.5
	5	σ_0	0	3.5	6	0	15	20

Table 3. The simple loops and their characteristics for the example of Fig. 12

No.	$(b_1 b_2)$	$G(R)$	Composition	$\lambda(\Sigma)$	$t(\Sigma)$	$\gamma(\Sigma)$	Rmk.
1	(0, 5)		$\Sigma_5^{(2)} \cup \Delta_0^{(2)}$	16	21	0.762	$R_{2,1}$
2	(1, 1)		$\Sigma_1^{(1)} \cup \Delta_1^{(1)} \cup \Sigma_1^{(2)} \cup \Delta_1^{(2)}$	6.5	8.5	0.765	$R_{1,0} \approx R_{2,0}$
3	(1, 2)		$\Sigma_1^{(1)} \cup \Delta_1^{(1)} \cup \Sigma_2^{(2)} \cup \Delta_2^{(2)}$	7.5	9.5	0.789	R^*
4	(1, 3)		$\Sigma_1^{(1)} \cup \Delta_1^{(1)} \cup \Sigma_3^{(2)} \cup \Delta_3^{(2)}$	14	18.5	0.757	
5	(1, 4)		$\Sigma_1^{(1)} \cup \Delta_1^{(1)} \cup \Sigma_4^{(2)} \cup \Delta_4^{(2)}$	15	19.5	0.769	
6	(1, 5)		$\Sigma_1^{(1)} \cup \Delta_1^{(1)} \cup \Sigma_3^{(2)} \cup \Delta_0^{(1)}$	21.5	28.5	0.754	
7	(2, 1)		$\Sigma_2^{(1)} \cup \Sigma_1^{(2)} \cup \Delta_1^{(2)}$	12	16	0.750	$R_{1,2}$
8	(2, 2)		$\Sigma_2^{(1)} \cup \Sigma_2^{(2)} \cup \Delta_2^{(2)}$	13	17	0.765	
9	(2, 3)		$\Sigma_2^{(1)} \cup \Sigma_3^{(2)} \cup \Delta_3^{(2)}$	19.5	26	0.750	
10	(2, 4)		$\Sigma_2^{(1)} \cup \Sigma_4^{(2)} \cup \Delta_4^{(2)}$	20.5	27	0.759	
11	(2, 5)		$\Sigma_2^{(1)} \cup \Sigma_5^{(2)} \cup \Delta_0^{(1)}$	27	36	0.750	

The c-codes of the loop-sections are easy to determine from the fact that the result of the decision $s(\sigma_{i,1})=s_0$ is $\sigma^{(3-i)}$, $i=1,2$, and the vertex the last arc of $G^{(a)}$ leads to can be obtained as the last typical situation of $\Sigma^{(a)}$ with $\beta_i=\sigma^{(i)}$, $i=1,2$.

From the possible (abc) codes Table 1 of the possible types $(b_1 b_2)$ of the simple loops can be completed. The data of the simple loops can be obtained from those loop-sections which are shown in Table 3. The last datum is $\gamma(\Sigma)$, the efficiency of the corresponding simple loop. Comparing these data we can choose the maximum value as 0.789. The type of the optimal schedule R^* is (1, 2) and its Gantt-chart can be seen in Fig. 12/c.

The table

R	$(b_1 b_2)$	R	$100\gamma/\gamma^*$
R^*	(1, 2)	0.789	100
$R_{1,0}$	(1, 1)	0.765	96.9
$R_{2,0}$	(1, 1)	0.765	96.9
$R_{1,2}$	(2, 1)	0.750	95.0
$R_{2,1}$	(0, 5)	0.762	96.5

shows that the priority schedules are not optimal. The efficiency γ^* of the optimal schedule is 88% of the sum $\gamma^{(1)}+\gamma^{(2)}=4.5/8+1/3=0.896$ and the efficiency of every priority schedule is less than γ^*. $\gamma_{1,2}$ is the minimum of the efficiency values of the CESs. This is 95% of the value γ^*. To find a good estimation for the $\min_{R\in\bar{\mathscr{R}}(Q)} \gamma(R)/\gamma^*$ is an open question. A trivial estimation is clearly $\max_{i=1,2} \gamma^{(i)}/(\gamma^{(1)}+\gamma^{(2)})$. In the example $\gamma_{2,1}$ is not minimal but there are 8 other CESs with greater efficiency. Also $R_{1,0}\approx R_{2,0}$ have better efficiency.

Fig. 12/c shows that the economic decisions in the optimal schedule are chosen such that the delay d caused by the decision be minimal. This heuristic scheduling strategy can often give a not bad schedule but not optimal in general. One can argue that a unit delay of the job-flow with a higher P_A-utilization $\gamma^{(i)}=\eta_i/\tau_i$ is worse than a unit delay of the other job-flow. Therefore, we can expect better schedule by the strategy which decides such that the *loss of utilization* $D_i=\gamma^{(i)}d_i$ by the delay d_i of $Q^{(i)}$ be minimum. For our example the critical situations of R^*, the delays d_i, the losses D_i and the decisions s^* are from the Fig. 12/c as follows:

σ'	d_1	D_1	d_2	D_2	s^*
σ_0	1	0.56	4.5	1.50	s_2
$\sigma_{2,1}$	1	0.56	2.5	0.83	s_2
$\sigma_{2,1}$	1	0.56	0.5	0.17	s_0
$\sigma_{1,1}$	0.5	0.28	4.5	1.50	s_0

The table shows that the optimal decisions correspond to the *strategy of minimizing local losses* of utilization. This strategy is not optimal in every cases either. We show this by the example configuration $Q=(1;\ 3.5;\ 2;\ 1.5)$ in Fig. 13. The graph

$G(Q)$ with the data $\lambda(\Delta\Sigma)$, $t(\Delta\Sigma)$ and Δ is the part (*d*). The data (6)—(10) are

$$n_0 = 0,\quad n_{11} = 0,\quad n_{12} = 0,\quad \delta_{10} = 1,\quad \delta_{11} = 0,\quad \delta_{12} = 0,\quad n_1 = 0,$$
$$n_{21} = 0,\quad n_{22} = 1,\quad \delta_{20} = 0,\quad \delta_{21} = 1,\quad \delta_{22} = 0,\quad n_2 = 1,$$
$$N_L = 3,\quad \overline{N} = 3.$$

The possible three CESs are $R_{1,2}=R_{1,0}$, $R_{2,1}$ and $R_{2,0}$ by Fig. 13/a, b, c. The efficiency values are $\gamma_{1,2}=\gamma_{1,0}=0.667$, $\gamma_{2,1}=0.743$, $\gamma_{2,0}=0.727$. $R^*=R_{2,1}$ is the

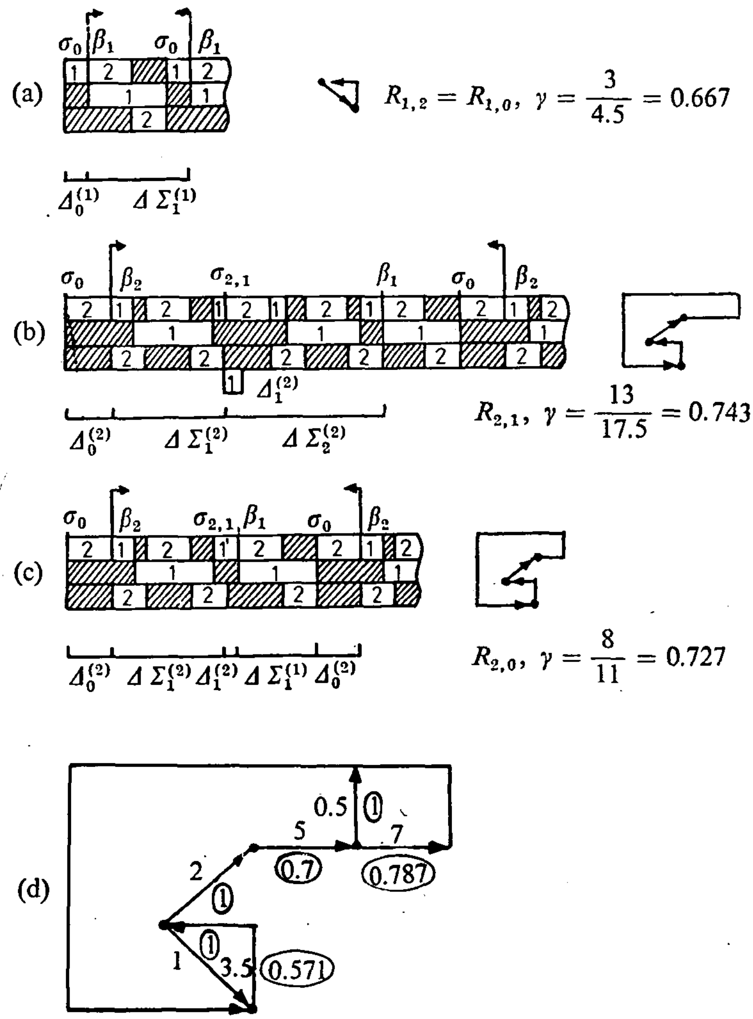


Fig. 13

Example configuration for no optimal minimum local losses strategy $Q=(1;\ 3.5;\ 2;\ 1.5)$

optimal schedule. The delays, losses and optimal decisions are the following ($\gamma^{(1)}=0.222$, $\gamma^{(2)}=0.571$):

σ'	d_1	D_1	d_2	D_2	s^*
σ_0	2	0.444	1	0.571	s_2
$\sigma_{2,1}$	2	0.444	0.5	0.285	s_2

The preemption $s(\sigma_{2,1})=s_2$ causes a greater delay (2) and local loss (0.444) than the decision $s(\sigma_{2,1})=s_0$ would but it is, nevertheless, optimal. The decision $s(\sigma_{2,1})=s_0$ results in $R_{2,0}$ which is not an optimal schedule (see Fig. 13/c). This example shows that the "locally optimal" decisions are not "totally optimal". An evident problem is the ratio γ/γ^* of the efficiency of the schedule with minimal local losses and the efficiency of the optimal schedule.

After the examples we give, now, an algorithm to determine an optimal schedule by direct evaluation and comparison of the CESs in finite cases. Formally we divide the algorithm into two parts and formulate the parts as the S-algorithm and the E-algorithm.

The S-algorithm produce the series of vectors

$$Z_{ab}=(\lambda_{ab}, t_{ab}, c_{ab}), \quad b=1,2,\dots,n_a+1, \quad a=1,2$$

with components

$$\lambda_{ab}=\lambda(\Sigma_b^{(a)})+\Delta_b^{(a)}, \quad t_{ab}=t(\Sigma_b^{(a)})+\Delta_b^{(a)}, \quad c_{ab}$$

as P_A-usage, length and c-code of the loop-section with code (ab). An auxiliary variable is in the algorithm $X=(\lambda, t, \Delta)$ as P_A-usage, length of subsections of $\Sigma^{(a)}$ and the length of a next section which will be inspected afterwards. Another auxiliary variable is $\bar{Y}=(\bar{\lambda}, \bar{t})$ the cumulated P_A-usages and lengths of the subsections. The algorithm supplies also the data n_{aj}, δ_{aj} defined by (6)—(7) and used in (9)—(10).

S-algorithm. *Input data:* $Q=(\eta_1;\ \vartheta_1;\ \eta_2;\ \vartheta_2)$;

Output data: $n_a, n_{aj}, j=1,2, \delta_{aj}, j=0,1,2, a=1,2, Z_{ab}=(\lambda_{ab}, t_{ab}, c_{ab})$, $b=1,\dots,n_a+1$, $a=1,2$;

Step 0: $\tau_1:=\eta_1+\vartheta_1$; $\tau_2:=\eta_2+\vartheta_2$; $a:=1$; $n:=1$; $i:=2$;
Step 1: $X:=(0,0,\vartheta_a)$; $\bar{Y}:=(0,0)$;
Step 2: $l:=[\Delta/\tau_i]$; $\Delta':=\Delta-l\tau_i$;
Step 3: If $\Delta'>\eta_i$ then $X:=(\lambda+(l+1)\eta_i, t+\Delta, \tau_i-\Delta')$, $i:=3-i$ and go to *Step 2;*
If $\Delta'=\eta_i$ then $\bar{Y}:=(\bar{\lambda}+\lambda+(l+1)\eta_i, \bar{t}+t+\Delta)$, $Z_{an}:=(\bar{\lambda}, \bar{t}, i)$, $\delta_{ai}:=1$ and go to *Step 4;*
If $\Delta'=0$ then $\bar{Y}:=(\bar{\lambda}+\lambda+l\eta_i, \bar{t}+t+\Delta)$, $Z_{an}:=(\bar{\lambda}, \bar{t}, 0)$, $\delta_{a0}:=1$ and go to *Step 4;*
$\bar{Y}:=(\bar{\lambda}+\lambda+l\eta_i+\Delta', \bar{t}+t+\Delta)$; $\Delta:=\eta_i-\Delta'$; $Z_{an}:=(\bar{\lambda}+\Delta, \bar{t}+\Delta, i)$;
$n_{a,3-a}:=n_{a,3-a}+1$; $i:=3-i$; $k:=[\Delta/\vartheta_i]$; $\Delta':=\Delta-k\vartheta_i$;
If $k>0$ then $\bar{Y}:=(\bar{\lambda}+\tau_i, \bar{t}+\tau_i)$, $Z_{an+j}:=(\bar{\lambda}+\Delta-j\vartheta_i, \bar{t}+\Delta-j\vartheta_i, 3-i)$, $j=1,\dots,k$ and $n_{ai}:=n_{ai}+k$;
$n:=n+k$;

If $\Delta'=0$ then $\bar{Y}:=(\bar{\lambda}+\tau_i, \bar{t}+\tau_i)$; $Z_{an}:=(\bar{\lambda}, \bar{t}, 3-i)$; $n_{ai}:=n_{ai}-1$;
$n:=n-1$; $\delta_{a,3-i}:=1$ and go to *Step 4;*
$n:=n+1$;
If $\vartheta_{3-i}+\Delta'-\vartheta_i \geqq 0$ then $X:=(\eta_i+\Delta', \tau_i, \vartheta_{3-i}+\Delta'-\vartheta_i)$ and go to *Step 2;*
$X:=(\eta_i+\Delta', \eta_i+\Delta'+\vartheta_{3-i}, \vartheta_i-\vartheta_{3-i}-\Delta')$; $i:=3-i$; go to *Step 2;*

Step 4: If $a=2$ then $n_2:=n$ and go to *End;*
$n_1:=n$; $n:=1$; $a:=2$; $i:=1$; go to *Step 1;*

End.

The output data of the S-algorithm corresponds to the data of Table 2 and the data (6)—(7). From these data the efficiency values of the possible simple loops can be determined by the E-algorithm. The flow-chart of the S-algorithm is shown in Fig. 14.

The E-algorithm uses the output data n_a, $a=1, 2$, and Z_{ab_a}, $b_a=1, \dots, n_a+1$, $a=1, 2$, of the S-algorithm and determines the efficiency values γ of the simple loops and provides the type $(b_1^* b_2^*)$ and efficiency γ^* of a simple loop with maximum efficiency. The order of evaluation of the simple loops will determine which of the possibly more than one simple loops with maximum efficiency will be chosen. This order can be seen in Table 1: the + entries of the first column with increasing b_1, the + entries of the first row with increasing b_2 and the other + entries by rows after.

E-algorithm. *Input data:* $\eta_1, \eta_2, n_1, n_2, Z_{ab}=(\lambda_{ab}, t_{ab}, c_{ab})$, $b=1, 2, \dots, n_a+1$, $a=1, 2$;

Output data: b_1^*, b_2^*, γ^*;

Definition of operation F: If $\gamma>\gamma^*$ then $b_1^*:=b_1, b_2^*:=b_2$ and $\gamma^*:=\gamma$;

Begin: $b_1^*:=b_2^*:=\gamma^*:=b_2:=0$;

For $b_1:=1$ step 1 until n_n+1 do if $c_{1b_1}=1$ then $\gamma:=\lambda_{1b_1}/t_{1b_1}$ and F;
If $c_{1b_1}=0$ then $\gamma:=(\lambda_{1b_1}+\eta_1)/(t_{1b_1}+\eta_1)$ and F; $b_1:=0$;
For $b_2:=1$ step 1 until n_2+1 do if $c_{2b_2}=2$ then $\gamma:=\lambda_{2b_2}/t_{2b_2}$ and F;
If $c_{2b_2}=0$ then $\gamma:=(\lambda_{2b_2}+\eta_2)/(t_{2b_2}+\eta_2)$ and F;
For $b_1:=1$ step 1 until n_1+1 do if $c_{1b_1}=2$ then
begin For $b_2:=1$ step 1 until n_2+1 do
if $c_{2b_2}=1$ then $\gamma:=(\lambda_{1b_1}+\lambda_{2b_2})/(t_{1b_1}+t_{2b_2})$ and F;
If $c_{2b_2}=0$ then $\gamma:=(\lambda_{1b_1}+\lambda_{2b_2}+\eta_1)/(t_{1b_1}+t_{2b_2}+\eta_1)$ and F;
end;
If $c_{1b_1}=0$ then for $b_2:=1$ step 1 until n_2+1 do
if $c_{2b_2}=1$ then $\gamma:=(\lambda_{1b_1}+\lambda_{2b_2}+\eta_2)/(t_{1b_1}+t_{2b_2}+\eta_2)$ and F;

End.

Fig. 15 shows the flow-chart of the E-algorithm. This clarifies the meaning of the "for-step-until-do" cycles used in the algoritm.

The verification of the S-algorithm is easy e.g. by following its operations graphically on the Gantt-charts of some configurations as of $Q=(4.5;\ 3.5;\ 1;\ 2)$ in Fig. 12. The E-algorithm does not need further verification.

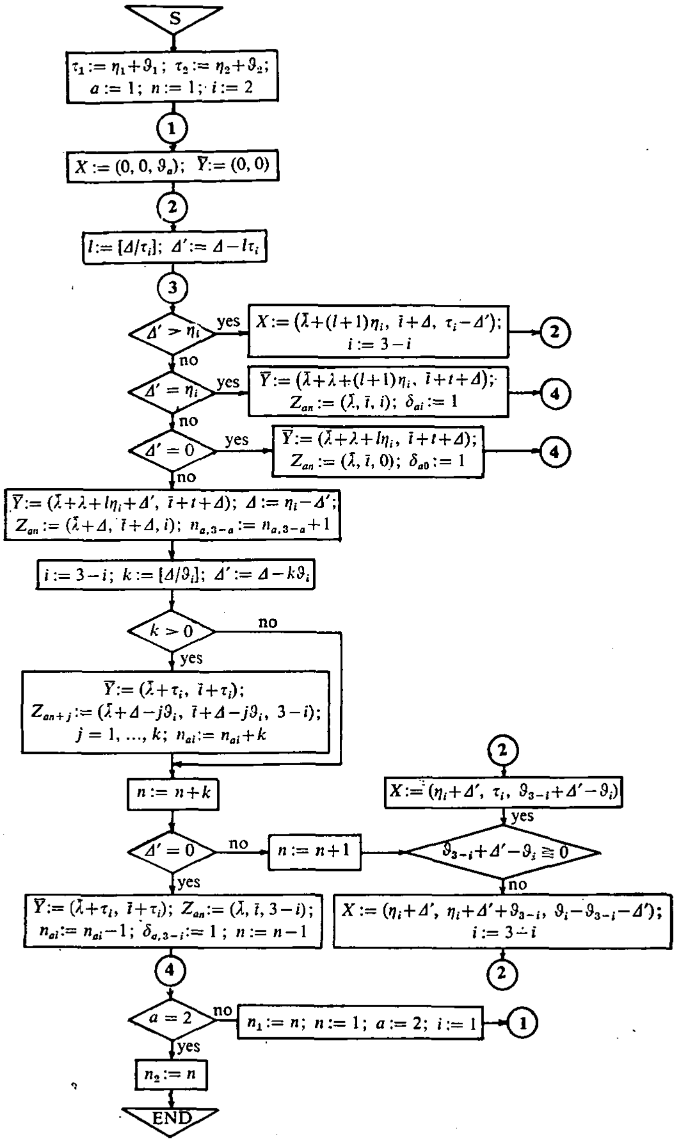


Fig. 14
The flow-chart of the *S*-algorithm

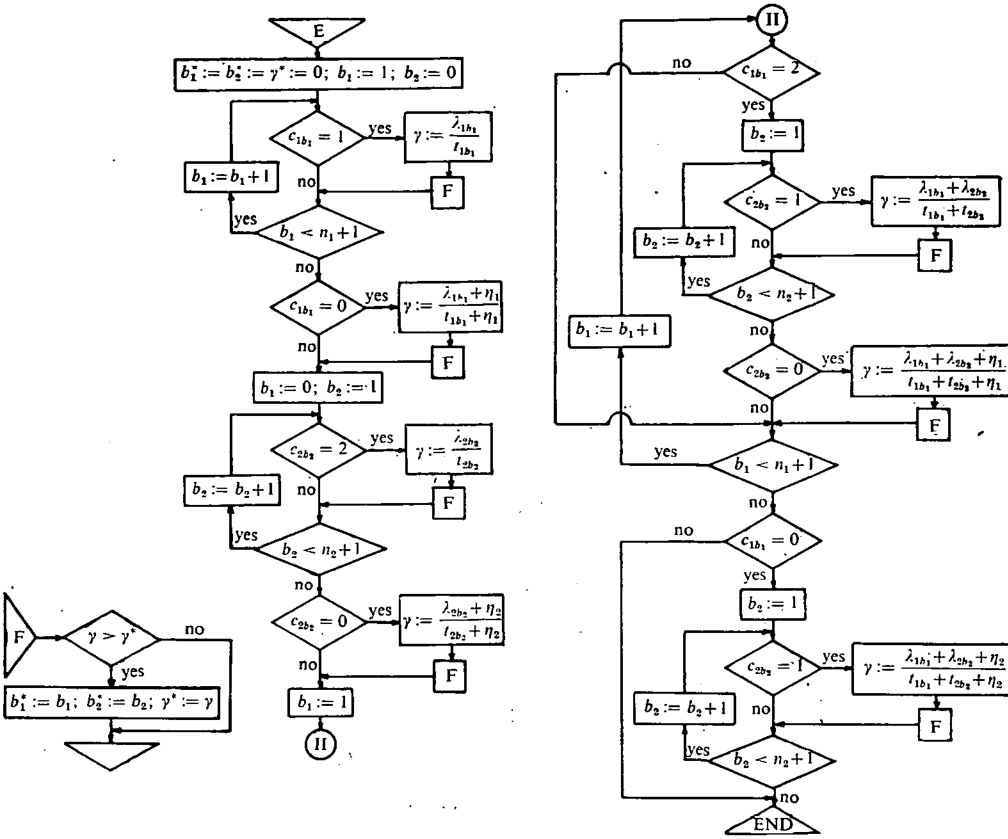


Fig. 15
The flow-chart of the *E*-algorithm

6. Summary

No simple rule to produce nor any simple method to choose an optimal schedule $R^*(Q)$ of any job-flow pair configuration Q is known. The dominance of the class of the consistent economical schedules (CESs) is proven here. We investigated the structure of the CESs and gave a classification for them. This is based upon the graph $G(Q)$ of the typical (critical) situations of two schedule sections $\Sigma^{(a)}$, $a=1, 2$. The information necessary to obtain $G(Q)$ and its data can be got by the *S*-algorithm if only $G(Q)$ is finite. In this case the *E*-algorithm supplies an optimal schedule and its efficiency. The discussion has shown the importance of some open problems which require further investigation. Such problems are: necessary and sufficient conditions for $G(Q)$ to be finite; estimations for the ratio of the efficiency values of CESs to the maximum value; detailed information about some heuristic strategies such as priority schedules and the schedule with minimum local losses.

KEYWORDS: steady job-flow pairs, preemptive scheduling, economic schedules, dominance.

COMPUTER SERVICE FOR
STATE ADMINISTRATION
CSALOGÁNY U. 30—32.
BUDAPEST, HUNGARY
H—1015

References

[1] ARATÓ, M., Diffusion approximation for multiprogrammed computer systems, *Comput. Math. Appl.*, v. 1, 1975, pp. 315—326.

[2] TANKÓ, J., A study on scheduling steady job-flow pairs, *Tanulmányok — MTA Számítástech. Automat. Kutató Int. Budapest,* v. 82, 83, 1978 (in Hungarian).

[3] TANKÓ, J., On non-preemptive scheduling of steady job-flow pairs, Dissertation, Hungarian Academy of Sciences, Budapest, 1979 (in Hungarian).

[4] TANKÓ, J., Non-preemptive scheduling of steady job-flow pairs, *Found. Control Engrg.*, to appear. Reduction method for non-preemtive scheduling steady job-flow pairs, *Found. Control Engrg.*, to appear.

[5] TOMKÓ, J., Processor utilization study, *Comput. Math. Appl.*, v. 1, 1975, pp. 337—344.

(Received Oct. 24, 1979)