DE GRUYTER OPEN

studia humana
QUARTERLY JOURNAL

Studia Humana
Volume 3:1 (2014), pp. 38—47
DOI: 10.2478/sh-2014-0004

Continuous Logic and Scheduling in Systems with Indeterminate Processing Times

Vitaly I. Levin

Mathematics Department, Penza State Technological University
1-a, Baidukov pr., Penza, 440039, Russia

e-mail: vilevin@mail.ru

Abstract:
A general approach to the synthesis of an optimal order of executing jobs in engineering systems with indeterminate (interval) times of job processing is presented. As a mathematical model of the system, a two-stage pipeline is taken whose first and second stages are, respectively, the input of data and its processing, and the corresponding mathematical apparatus is continuous logic and logic determinants.

Keywords: system, optimal order of jobs, continuous logic

1. Introduction

The study of engineering systems begins with determining the dependence of the performance factor of system on its parameters. This dependence can be used for estimating the performance factor of the system, to analyze it qualitatively, to optimally synthesize the system, etc. As a rule, the existing estimation methods for performance of engineering systems are oriented only to calculating performance factor of system and are not meant for their analysis and synthesis [1].

In [2, 3] there is an approach for studying various systems based on pipeline model of scheduling theory and on the mathematical apparatus of continuous logic and logical determinants, which makes it possible to derive an observable and easily calculated expression for system performance and to carry out qualitative analysis of the effect of system parameters on its performance and on its optimal synthesis according to the criterion of best performance. In this case the time parameters are assumed to be deterministic. In practice, these parameters are in many cases nondeterministic, which substantially hampers the study of system.

We consider an extension of the general approach for optimal synthesis of engineering systems with uncertain (interval) type time parameters to nondeterministic case [4]. Under application of this approach to optimal synthesis of engineering systems with interval time parameters this problem reduces to solving similar problems for two systems with deterministic time parameters equal to the upper and lower bounds of the corresponding intervals.

2. Problem Statement

Consider a system operating in batch mode and let the batch contain n different jobs $1,...,n$. We employ simplest two-phase model of system. So, in first phase performed by first system unit first operation, namely inputing of the initial data, is carried out; further, in other phase which performed by the second unit of system the second operation is carried out – transformation and processing of these

data in various functional units of the system (processor, main memory, and external memory) and the output the result. The units are assumed to operate consecutively. Each job i ($i = \overline{1,n}$) firstly goes to the first unit, where first operation is full performed, and after that goes to second unit, where the second operation is carried out completely.

The time of execution of the first operation on arbitrary job i is assumed to be known inexactly and to be determined by a closed interval $\widetilde{a}_i = [a_{i1}, a_{i2}]$ of all possible values of this time. In similar way the time of execution of the second operation on job i is set: $\widetilde{b}_i = [b_{i1}, b_{i2}]$. So, the first unit starts the execution of the current job immediately after end of the previous job, i.e., it operates without idle times, whereas the second unit starts the execution of the current job j only after the job j leaves the first unit, i.e., in the general case it operates with idle times. It is required to choose an order of jobs in the system under which its best performance is ensured, i.e., ttotal execution time of all jobs is minimum.

As in determiniscic case [5, 6], the optimal order of jobs can be assumed to be permutable, i.e., jobs must pass through two units in same order. Assume that execution times of first and second operations on an arbitrary job i are exact and are equal to a_i and b_i, respectively. Let for a pair of jobs (i, j) the order of passage through the first unit be $i \to j$, and the order of passage through second unit be opposite: $j \to i$. Let us change the order of jobs passing in the first unit by placing job i after j and moving job j (together with the jobs located earlier between i and j) to the left by length of freed time interval a_i. In this case the interval of the execution of one of the jobs i, which are subject to permutation, is moved to the right. However, it then ends at the time of completion of the execution of the job j in the first unit (before permutation, i.e., as previously, before the time of beginning of the execution of job in the second unit). Hence, a change in the order of jobs in the first unit does not affect the sequence of jobs in the second unit. Therefore, the same order of passage of jobs through the two units can be chosen without changing the resultant time of execution of all jobs. It means that for deterministic execution times of operations the optimal order in the sequence of jobs passing can be sought within the set of permutational orders of jobs. This conclusion is true for arbitrary deterministic execution times a_i and b_i of operations inside given intervals $\widetilde{a}_i = [a_{i1}, a_{i2}]$ and $\widetilde{b}_i = [b_{i1}, b_{i2}]$. Consequently, in accordance with the conditions of the problem, it remains valid if times of operations are assumed to be equal to the indicated interval values.

Thus, the solution of the stated problem reduces to finding an external permutation

$$P_n = (i_1, i_2, .., i_n), \quad i_k \in \{1,2,..., n\}, \tag{1}$$

of n given jobs that determines the order of jobs in the system, which is the same for its two units. The symbol i_k in expression (1) is the index of the job occupying the k-th place in the ordered sequence.

3. Logic Algebra of Nondeterministic Quantities and their Comparison

The problem solution requires some facts of the logic of nondeterministic interval quantities and of comparison theory for these quantities [4]. We shall proceed from continuous logic for deterministic (point) quantities [7]. The basic logical operations on these points are disjunction $\vee$ and conjunction $\wedge$ that are defined in following formulas:

$$\begin{aligned} a \vee b &= \max(a,b), \\ a \wedge b &= \min(a,b) \end{aligned} \tag{2}$$

Here $a, b \in C$, and the set C is an arbitrary interval of real numbers. Operations (2) obey the majority of laws of discrete logic, namely

$$a \vee a = a, \quad a \wedge a = a \qquad \text{(tautology)} \qquad (3)$$

$$a \vee b = b \vee a, \quad a \wedge b = b \wedge a \qquad \text{(commutative law)} \qquad (4)$$

$$(a \vee b) \vee c = a \vee (b \vee c), \quad (a \wedge b) \wedge c = a \wedge (b \wedge c) \qquad \text{(associative law)} \qquad (5)$$

$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c), \quad a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c) \qquad \text{(distributive law)} \qquad (6)$$

$$a \vee (a \wedge b) = a, \ a \wedge (a \vee b) = a \qquad (7)$$

$$a + (b \underset{\wedge}{\overset{\vee}{}} c) = (a+b) \underset{\wedge}{\overset{\vee}{}} (a+c), \qquad (8)$$

$$a - (b \underset{\wedge}{\overset{\vee}{}} c) = (a-b) \underset{\vee}{\overset{\wedge}{}} (a-c), \qquad (9)$$

$$a \cdot (b \underset{\wedge}{\overset{\vee}{}} c) = (a \cdot b) \underset{\wedge}{\overset{\vee}{}} (a \cdot c), \quad a,b,c > 0, \qquad (10)$$

$$-a \cdot (b \underset{\wedge}{\overset{\vee}{}} c) = (-a \cdot b) \underset{\vee}{\overset{\wedge}{}} (-a \cdot c), \quad a,b,c > 0, \qquad (11)$$

A special partial case of the equation (11) for $a = 1$ is the following law:

$$-(b \underset{\wedge}{\overset{\vee}{}} c) = (-b) \underset{\vee}{\overset{\wedge}{}} (-c), \qquad (12)$$

We now pass to continuous logic for interval quantities. In this case the continuous-logical operations of disjunction and conjunction (2) are generalized as set-theoretic constructions:

$$\begin{aligned} &\widetilde{a} \vee \widetilde{b} = \{a \vee b \mid a \in \widetilde{a}, b \in \widetilde{b}\}; \\ &\widetilde{a} \wedge \widetilde{b} = \{a \wedge b \mid a \in \widetilde{a}, b \in \widetilde{b}\}. \end{aligned} \qquad (13)$$

Here $\widetilde{a} = [a_1, a_2]$ and $\widetilde{b} = [b_1, b_2]$ are intervals regarded as the corresponding sets of points (values) belonging to them. According to [4], operations on intervals (13) obey the same laws (3)–(12) as the operations on point quantities (2). In particular, distributive laws (8) and the law (12) take form:

$$\widetilde{a} + (\widetilde{b} \underset{\wedge}{\overset{\vee}{}} \widetilde{c}) = (\widetilde{a} + \widetilde{b}) \underset{\wedge}{\overset{\vee}{}} (\widetilde{a} + \widetilde{c}), \qquad (14)$$

$$-(\widetilde{b} \underset{\wedge}{\overset{\vee}{}} \widetilde{c}) = (-\widetilde{b}) \underset{\vee}{\overset{\wedge}{}} (-\widetilde{c}). \qquad (15)$$

Due to [4] the results of the logical operations of disjunction and conjunction on intervals (13) are calculated by the formulas

$$\widetilde{a} \vee \widetilde{b} = [a_1, a_2] \vee [b_1, b_2] = [a_1 \vee b_1, a_2 \vee b_2], \qquad (16)$$

$$\widetilde{a} \wedge \widetilde{b} = [a_1, a_2] \wedge [b_1, b_2] = [a_1 \wedge b_1, a_2 \wedge b_2]. \qquad (17)$$

We briefly present some important facts of comparison theory for intervals. [4]

1. For any pair of intervals $\widetilde{a} = [a_1, a_2]$ and $\widetilde{b} = [b_1, b_2]$ the equivalence relation

$$(\widetilde{a} \vee \widetilde{b} = \widetilde{a}) \Leftrightarrow (\widetilde{a} \wedge \widetilde{b} = \widetilde{b}), \qquad (18)$$

holds, i.e., like point quantities, the intervals are compatible (in the sense that if one of the two quantities is maximal, then the other is minimal and vice versa).

2. Pairs of intervals $\tilde{a} = [a_1, a_2]$ and $\tilde{b} = [b_1, b_2]$ can be in relations «greater than» and «smaller than» defined in the same way as in the case of point quantities by the such equivalence:

$$(\tilde{a} \geq \tilde{b}) \Leftrightarrow (\tilde{a} \vee \tilde{b} = \tilde{a}, \tilde{a} \wedge \tilde{b} = \tilde{b}). \tag{19}$$

3. In accordance with (19), any two intervals $\tilde{a}$ and $\tilde{b}$ that are in relation $\tilde{a} \geq \tilde{b}$ or $\tilde{a} \leq \tilde{b}$ are said to be comparable. Otherwise $\tilde{a}$ and $\tilde{b}$ are incomparable.

4. For intervals $\tilde{a} = [a_1, a_2]$ and $\tilde{b} = [b_1, b_2]$ to be comparable and satisfy the relation $\tilde{a} \geq \tilde{b}$ it is necessary and sufficient that system of inequalities $(a_1 \geq b_1, a_2 \geq b_2)$ holds, and for $\tilde{a}$ and $\tilde{b}$ to be incomparable it is necessary and sufficient that at least one of systems of inequalities $(a_1 < b_1, a_2 > b_2)$ or $(b_1 < a_1, b_2 > a_2)$ are true. Thus, only the intervals displaced relative to each other along number axis are comparable; in this case interval displaced to the right is greater. If one of intervals overlaps other the intervals are incomparable.

5. In a system of intervals $\tilde{a}_1, \tilde{a}_2, ..., \tilde{a}_k$ the interval $\tilde{a}_1$ is said to be maximal (minimal) interval if it is comparable with other intervals $a_2, ..., a_k$ and is in relations $\tilde{a}_1 \geq \tilde{a}_2, ..., \tilde{a}_1 \geq \tilde{a}_k$ $(\tilde{a}_1 \leq \tilde{a}_2, ..., \tilde{a}_1 \leq \tilde{a}_k)$ with them.

6. It is necessary and sufficient in system of intervals $\tilde{a}_1 = [a_{11}, a_{12}], \tilde{a}_2 = [a_{21}, a_{22}], ..., \tilde{a}_k = [a_{k1}, a_{k2}]$ for interval $\tilde{a}_1$ be maximal that the system of the relations holds:

$$a_{11} = \bigvee_{i=1}^{k} a_{i1}, \quad a_{12} = \bigvee_{i=1}^{k} a_{i2}, \tag{20}$$

and for $\tilde{a}_1$ to be minimal it is necessary and sufficient that following equations is true:

$$a_{11} = \bigwedge_{i=1}^{k} a_{i1}, \quad a_{12} = \bigwedge_{i=1}^{k} a_{i2}, \tag{21}$$

4. Derivation of Optimality Conditions

In the previous case we define a relationship between the execution times $\tilde{a}_i, \tilde{b}_i, \tilde{a}_j, \tilde{b}_j$ of two arbitrary jobs (i, j) under which they must be executed in order $i \to j$ in optimal sequence of jobs $P(n)$ (1). Let $P_k = (i_1, ..., i_k); k \leq n$, be initial section of P_n and let $\tilde{t}_1(P_k)$ and $\tilde{t}_2(P_k)$ be time intervals containing all possible times of completion of sequence P_k in 1st and 2nd units. Because $P_{k+1} = (P_k, i_{k+1})$, we can write

$$\tilde{t}(P_{k+1}) = \tilde{t}_1(P_k) + \tilde{a}_{i_{k+1}}, \quad \tilde{t}_2(P_{k+1}) = [\tilde{t}_1(P_{k+1}) \vee \tilde{t}(P_k)] + \tilde{b}_{i_{k+1}}. \tag{22}$$

Here $\vee$ is disjunction of type (13). The recurrence relations (22) make it possible to calculate the total time of execution for any order of the sequence of jobs P_n in form of a time interval $\tilde{T}(2, n) = \tilde{t}_2(P_n)$. Let $P_n^1 = (i_1, ..., i_k, i, j_1, ..., j_n)$ and $P_n^2 = (i_1, ..., i_k, j, j_1, ..., j_n)$ be two sequences of jobs passing through the system that differ only in order of execution of jobs i and j occupying the $(k+1)$-th and $(k+2)$-th positions in sequence. Let us find out when P_n^1 is more preferable than P_n^2, i.e.

when jobs i and j must be executed in order $i \to j$ (and not vice versa). The corresponding condition is written as

$$\tilde{t}_2(P_{k+2}^1) \le \tilde{t}_2(P_{k+2}^2). \tag{23}$$

According to (22), the sequence P_n^1 is more preferable than P_n^2 if the time or passage of its regulated subsequence P_{k+2}^1 through two units is less than that of P_{k+2}^2. To write preference condition in explicit form we must express $\tilde{t}_2(P_{k+2})$ via the time parameters $\tilde{a}_i$ and $\tilde{b}_i$ of jobs. Let $\tilde{t}_1(P_k) = \tilde{t}$. Then $\tilde{t}_2(P_k) = \tilde{t} + \tilde{\Delta}$, where $\tilde{\Delta} = \tilde{b}_{i_k}$. By the fact that $P_{k+1}^1 = (P_k, i)$ and $P_{k+2}^1 = (P_{k+1}^1, j) = (P_k, i, j)$, on applying twice recurrence relations (22) we obtain

$$\tilde{t}_1(P_{k+1}^1) = \tilde{t} + \tilde{a}_i;\ \tilde{t}_2(P_{k+1}^1) = ((\tilde{t} + \tilde{a}_i) \vee (\tilde{t} + \tilde{\Delta})) + \tilde{b}_i;$$
$$\tilde{t}_1(P_{k+2}^1) = \tilde{t} + \tilde{a}_i + \tilde{a}_j;$$
$$\tilde{t}_2(P_{k+2}^1) = \{(\tilde{t} + \tilde{a}_i + \tilde{a}_j) \vee [((\tilde{t} + \tilde{a}_i) \vee (\tilde{t} + \tilde{\Delta})) + \tilde{b}_i]\}.$$

We similarly determine haracteristics P_{k+1}^2 and P_{k+2}^2; in this case we have

$$\tilde{t}_2(P_{k+2}^2) = \{(\tilde{t} + \tilde{a}_i + \tilde{a}_j) \vee [((\tilde{t} + \tilde{a}_i) \vee (\tilde{t} + \tilde{\Delta})) + \tilde{b}_j]\} + \tilde{b}_i.$$

The substitution of the above expressions into the formula (23) yields explicit form of the condition under which the jobs i and j in the optimal sequence must follow in the order $i \to j$:

$$\{(\tilde{t} + \tilde{a}_i + \tilde{a}_j) \vee [((\tilde{t} + \tilde{a}_i) \vee (\tilde{t} + \tilde{\Delta})) + \tilde{b}_i]\} + \tilde{b}_i \le \{(\tilde{t} + \tilde{a}_i + \tilde{a}_j) \vee [((\tilde{t} + \tilde{a}_j) \vee (\tilde{t} + \tilde{\Delta})) + \tilde{b}_j]\} + \tilde{b}_i. \tag{24}$$

To simplify inequalities (24) we apply the laws (8), (12) and we can take by (8) the term $\tilde{t}$ outside the parentheses on both sides of (24). On canceling it, we find

$$\{(\tilde{a}_i + \tilde{a}_j) \vee [(\tilde{a}_i \vee \tilde{\Delta}) + \tilde{b}_i]\} + \tilde{b}_j \le \{(\tilde{a}_i + \tilde{a}_j) \vee [(\tilde{a}_i \vee \tilde{\Delta}) + b_j]\} + \tilde{b}_i.$$

We now take the terms $\tilde{a}_i, \tilde{a}_j, \tilde{b}_i$ and $\tilde{a}_i, \tilde{a}_j, \tilde{b}_j$ outside the curly brackets on left- and right-hand sides of the new inequality, respectively. On canceling the common terms on the two sides we write

$$(-\tilde{b}_i) \vee (\tilde{\Delta} - \tilde{a}_i - \tilde{a}_j) \vee (-\tilde{a}_j) \le (-\tilde{b}_j) \vee (\tilde{\Delta} - \tilde{a}_i - \tilde{a}_j) \vee (-\tilde{a}_i).$$

Based on law (12), we take the minus sign outside all brackets in the last inequality and multiply its left- and right-hand sides by -1, which results in

$$\tilde{a}_i \wedge \tilde{b}_j \wedge (\tilde{a}_i + \tilde{a}_j - \tilde{\Delta}) \le \tilde{a}_j \wedge \tilde{b}_i \wedge (\tilde{a}_i + \tilde{a}_j - \tilde{\Delta}). \tag{25}$$

The symbol $\wedge$ in (25) is conjunction (13). Let us solve inequality (25). We rewrite it in the form

$$\widetilde{L} \wedge \widetilde{D} \le \widetilde{M} \wedge \widetilde{D}, \tag{26}$$

where $\widetilde{L} = \widetilde{a}_i \wedge \widetilde{b}_j$, $\widetilde{M} = \widetilde{a}_j \wedge \widetilde{b}_i$, $\widetilde{D} = \widetilde{a}_i + \widetilde{a}_j - \widetilde{\Delta}$. The logical inequality (26) for interval quantities is solved by the same separation method as for point quantities [7]. We obtain $\widetilde{L} \le \widetilde{M}$ (always), $\widetilde{L} > \widetilde{M}$ (for $\widetilde{D} \le \widetilde{M}$) for (26), and, on returning to the original quantities, we derive the following solutions to (25):

$$\widetilde{a}_i \wedge \widetilde{b}_j \le \widetilde{a}_j \wedge \widetilde{b}_i, \tag{27}$$

$$\widetilde{a}_i + \widetilde{a}_j - \widetilde{\Delta} \le \widetilde{a}_j \wedge \widetilde{b}_i < \widetilde{a}_i \wedge \widetilde{b}_j, \tag{28}$$

The inequality (27) involves only time characteristics of jobs i and j. If (27) holds then jobs i, j in the optimal sequence P_n follow in the order $i \to j$ irrespective of the order of the other jobs. Besides the characteristics of i and j, inequality (28) contains the parameter Δ depending on subsequence P_k preceding i and j. Fulfillment of condition (28) means that jobs i and j in the optimal sequence P_n for execution of jobs follow in order $i \to j$ only in the case when the preceding subsequence P_k has the corresponding value of the parameter Δ. It is clear that for optimal scheduling of jobs it is more advisable to use condition (27) stated as the following independent theorem.

Theorem 1. For jobs i and j in optimal sequence of execution of all n jobs in a two-unit nondetermined system with execution times of first and second operations of job i in form of intervals $\widetilde{a}_i = [a_{i1}, a_{i2}]$ and $\widetilde{b}_i = [b_{i1}, b_{i2}]$ to follow in the order $i \to j$ irrespective of the order of execution ofother jobs it is necessary and sufficient that the time parameters i and j satisfy condition (27).

5. Reduction to Deterministic Problems

We will reduce the optimality conditions for the order of execution of jobs in the nondeterministic engineering system in question that are established in Theorem 1 to the well-known optimality conditions for the order of execution of jobs in different deterministic systems [4]. Consider two two-unit deterministic systems. Let the execution times of the first and second operations on an arbitrary job i in the first system be equal to the lower bounds a_{i1} and b_{i1} of the times $\widetilde{a}_i$ and $\widetilde{b}_i$ of execution of these operations in given nondeterministic system, respectively, and let in other systems these times be equal to the lower and upper bounds a_{i2} and b_{i2} of the times $\widetilde{a}_i$ and $\widetilde{b}_i$. We will call these systems accordingly the lower and the upper deterministic boundary systems relative to the nondeterministic system.

Theorem 2. For jobs i and j in optimal sequence of execution of all n jobs in two-unit nondetermined system with execution times of first and second operations of job i in form of the intervals $\widetilde{a}_i = [a_{i1}, a_{i2}]$ and $\widetilde{b}_i = [b_{i1}, b_{i2}]$ to be carried out in the order $i \to j$ irrespective of the order of execution of the other jobs it is necessary and sufficient that jobs i and j be carried out in same order irrespective of execution of other jobs, i.e. in order of execution in the optimal sequences for execution of all jobs in two deterministic two-unit systems, namely in lower and upper boundary systems. Theorem 2 implies following theorem.

Theorem 3. For a type (1) permutation $P_n = (i_1,...,i_n)$ be an optimal sequence of execution of n jobs in a nondeterministic two-unit engineering system with execution times of the first and second operations on job i in form of intervals $\widetilde{a}_i = [a_{i1}, a_{i2}]$ and $\widetilde{b}_i = [b_{i1}, b_{i2}]$ it is necessary and sufficient that P_n be also the optimal sequence of the execution of n operations in the lower and upper boundary systems. Theorem 3 implies the two theorems below.

Theorem 4. The set M of all optimal sequences of n jobs in a nondeterministic two-unit computing system with execution times of the first and second operations of job i in form of intervals $\widetilde{a}_i = [a_{i1}, a_{i2}]$ and $\widetilde{b}_i = [b_{i1}, b_{i2}]$ is the intersection of the sets M_l and M_u of the all optimal sequences of n jobs in its lower and upper deterministic boundary systems.

Theorem 5. For an optimal secuence $P_n = (i_1,...,i_n)$ of execution of all n jobs to exist in a nondeterministic two-unit computing system with execution times of the first and second operations of job i in form of intervals $\widetilde{a}_i = [a_{i1}, a_{i2}]$ and $\widetilde{b}_i = [b_{i1}, b_{i2}]$ it is necessary and sufficient that the intersection of the sets M_l and M_u of all optimal sequences of the execution of n jobs in its lower and upper deterministic boundary systems be nonempty.

Theorems 4 and 5 imply the following direct solution algorithm for the stated problem, i.e. for finding an optimal sequence $P_n = (i_1,...,i_n)$ of execution of n jobs in a nondeterministic two-unit system with execution times of first and second operations of job i in the form of intervals $\widetilde{a}_i = [a_{i1}, a_{i2}]$ and $\widetilde{b}_i = [b_{i1}, b_{i2}]$.

Step 1. Finding the set M_l of all optimal sequences of execution of n jobs in lower boundary system of original system with execution times $a_i = a_{i1}$ and $b_i = b_{i1}$, which are the times of 1st and 2nd operations of job i. The well-known solution methods for deterministic two-stage problem of scheduling in industrial systems are used [2, 3, 5, 6].

Step 2. Finding the set M_u of all optimal sequences of execution of n jobs in upper boundary system of the original system with execution times $a_i = a_{i2}$ and $b_i = b_{i2}$, which are the times of 1st and 2nd operations of job i, using the same methods as in Step 1.

Step 3. Finding the intersection $M_l \cap M_u$ of the sets, which is the set M of all optimal sequences of execution of n jobs in the given nondeterministic two-unit system. If $M \neq \varnothing$ then any sequence $P_n \in M$ is desired optimal sequence of execution of n jobs. If $M = \varnothing$ then there are no such sequences.

The suggested direct solution algorithm for the problem requires exhaustion when determining the intersection of the sets M_l and M_u, and therefore it is efficient only for $|M_l| = |M_u| = 1$ or for $|M_l|$ and $|M_u|$ close to 1. In case $|M_l|$ or $|M_u|$ is large, the direct algorithm is ineffective, and it is necessary to pass to the application of decision rules making it possible to find an optimal sequence of execution of jobs in a nondeterministic computing system without exhaustion.

6. Construction of Decision Rules

Consider an arbitrary two-unit deterministic computing system with the times of execution a_i and b_i of the first and second operations of job i in the first and second units respectively. We split the set of jobs into first, second and third classes of jobs: $(a_i < b_i), (a_i > b_i)$ and $(a_i = b_i)$. Then the decision rules for finding optimal sequences of execution of all jobs in a system are based on the schedule

presented in the Table 1. An arbitrary cell (p,q) of the table contains a condition under which two arbitrary jobs i and j (belonging to the p-th and q-th classes respectively) are placed in order $i \to j$ in optimal sequence. The schedule makes it possible to state a non-exhaustive decision rule for finding all optimal sequences of jobs for any set of jobs.

Table 1

Class of job	Order of execution		
	1	2	3
1	$a_i \le a_j$	Always	always
2	never	$b_i \ge b_j$	$b_i \ge b_j$
3	$a_i \le a_j$	Always	always

For example, the cell (1,1) shows that for the set of jobs of the first class the optimal execution sequence is obtained by arranging job i in increasing (more precisely, nondecreasing) order relative to parameter a_i.

Let us apply a similar approach to a given nondeterministic two-unit computing system with execution times of the first and second operations of job i in the form of intervals $\widetilde{a}_i = [a_{i1}, a_{i2}]$ and $\widetilde{b}_i = [b_{i1}, b_{i2}]$. Along with this system consider its lower and upper deterministic boundary processing systems (Table 1). The former has execution times a_{i1} and b_{i1} of the 1st and 2nd operations of job i, and for the latter these values are a_{i2} and b_{i2}. By Theorem 3 an optimal sequence of execution of jobs in a nondeterministic system is also an optimal sequence of the execution of jobs in its lower and upper deterministic boundary systems. Therefore, the optimality condition for a sequence of jobs in a nondeterministic system is the intersection of similar conditions for its lower and upper boundary systems.

Consider lower boundary system. In accordance with presented technique we split its set of n jobs into jobs of the first, second and third classes: $(a_{i1} < b_{i1}), (a_{i1} > b_{i1})$ and $(a_{i1} = b_{i1})$ respectively. Let us compile the schedule of execution for this system (see Table 2).

We now consider the upper boundary system. By the same technique we split its set of n jobs into jobs of the first, second and third classes: $(a_{i2} < b_{i2}), (a_{i2} > b_{i2})$ and $(a_{i2} = b_{i2})$. We thus obtain Table 3 of the schedule of operation of this system.

The schedule for a nondeterministic processing system is intersection of schedules of its lower (Table 2) and upper (Table 3) deterministic boundary systems of the original system. This table is compiled in the following way. Using the combination of some cells (p_l, q_l) and (p_u, q_u) of Tables 2 and 3 respectively we form the cell $((p_l, p_u),(q_l, q_u))$ of the desired table into which the condition equal to the intersection of the conditions in the cells (p_l, q_l) and (p_u, q_u) of Tables 2 and 3 respectively is inserted.

If the inserted condition in the cell contains the words «always» and «never» it is simplified in the following way: $A \cap \text{always} = A$, $A \cap \text{never} = \text{never}$, A is arbitrary.

Table 2

Class of job	Order of execution		
	$1l$	$2l$	$3l$
$1l$	$a_{i1} \le a_{j1}$	Always	always
$2l$	never	$b_{i1} \ge b_{j1}$	$b_{i1} \ge b_{j1}$
$3l$	$a_{i1} \le a_{j1}$	Always	always

Table 3

Class of job	Order of execution		
	$1u$	$2u$	$3u$
$1u$	$a_{i2} \le a_{j2}$	Always	always
$2u$	never	$b_{i2} \ge b_{j2}$	$b_{i2} \ge b_{j2}$
$3u$	$a_{i2} \le a_{j2}$	Always	always

The presented procedure is carried out for all possible combinations of cells in Tables 2 and 3. As a result schedule for nondeterministic processing system (Table 4) is constructed. In each cell $((p_l, p_u),(q_l, q_u))$ of the Table 4 the complex condition is presented under which the arbitrary jobs i and j (where the job i belongs to the p_l-th class of the lower boundary system and to the p_u-th class of the upper boundary system and job j belongs to the q_l-th class of the lower boundary system and to the q_u-th class of the upper boundary system) are placed in an optimal sequence of execution of jobs in the order $i \to j$. The conditions in Table 4 are given in the form of inequalities for the boundaries of intervals determining the execution times of jobs and, when possible, in the form of inequalities for the indicated intervals.

For construction of non-exhaustive decision rules for determining all optimal sequences of executions of jobs in nondeterministic systems we use Table 4. In contrast to deterministic systems an optimal sequence of execution of jobs in nondeterministic systems may not exist. This is due to the fact that different intervals (execution times of jobs) may not be compaprble and may not have minimal and maximal intervals. The decision rules for each set of classes of jobs forming the set of jobs performed in the nondeterministic system are constructed separately.

7. Example

We will construct the decision rule for finding the optimal sequences of the execution of jobs belonging to the single class $(1_l,1_u)$. The condition in the cell $((1_l,1_u),(1_l,1_u))$ of Table 4 shows that the jobs i in the desired sequences must follow in nondecressing order of the interval parameter $\tilde{a}_i = [a_{i1}, a_{i2}]$ or, which is the same, in nondecreasing order of the two parameters: a_{i1} and a_{i2}. What has been said implies the following rule: arrange all jobs i in nondecreasing order relative to the parameter a_{i1} and thus obtain the corresponding set M_1 of ordered sequences of jobs; arrange all jobs i in nondecreasing order relative to the parameter a_{i2} and thus obtain a similar set of sequences M_2; take the intersection of the sets M_1 and M_2 which gives the desired set of optimal sequences of jobs.

Table 4

Class of job	Order of execution								
	1*l*1*u*	1*l*2*u*	1*l*3*u*	2*l*1*u*	2*l*2*u*	2*l*3*u*	3*l*1*u*	3*l*2*u*	3*l*3*u*
1*l*1*u*	$\widetilde{a}_i \le \widetilde{a}_j$	$a_{i1} \le a_{j1}$	$a_{i1} \le a_{j1}$	$a_{i2} \le a_{j2}$	always	Always	$a_{i2} \le a_{j2}$	always	always
1*l*2*u*	never	$a_{i1} \le a_{j1}$ $b_{i2} \le b_{j2}$	$a_{i1} \le a_{j1}$ $b_{i2} \le b_{j2}$	never	$b_{i2} \ge b_{j2}$	$b_{i2} \ge b_{j2}$	never	$b_{i2} \ge b_{j2}$	$b_{i2} \ge b_{j2}$
1*l*3*u*	$\widetilde{a}_i \le \widetilde{a}_j$	$a_{i1} \le a_{j1}$	$a_{i1} \le a_{j1}$	$a_{i2} \le a_{j2}$	always	Always	$a_{i2} \le a_{j2}$	always	always
2*l*1*u*	never	never	never	$b_{i1} \ge b_{j1}$ $a_{i2} \le a_{j2}$	$b_{i1} \ge b_{j1}$	$b_{i1} \ge b_{j1}$	$b_{i1} \le b_{j1}$ $a_{i2} \le a_{j2}$	$b_{i1} \ge b_{j1}$	$b_{i1} \ge b_{j1}$
2*l*2*u*	never	never	never	never	$\widetilde{b}_i \ge \widetilde{b}_j$	$\widetilde{b}_i \ge \widetilde{b}_j$	never	$\widetilde{b}_i \ge \widetilde{b}_j$	$\widetilde{b}_i \ge \widetilde{b}_j$
2*l*3*u*	never	never	never	$b_{i1} \le b_{j1}$ $a_{i2} \le a_{j2}$	$b_{i1} \ge b_{j1}$	$b_{i1} \ge b_{j1}$	$b_{i1} \le b_{j1}$ $a_{i2} \le a_{j2}$	$b_{i1} \ge b_{j1}$	$b_{i1} \ge b_{j1}$
3*l*1*u*	$\widetilde{a}_i \le \widetilde{a}_j$	$a_{i1} \le a_{j1}$	$a_{i1} \le a_{j1}$	$a_{i2} \le a_{j2}$	always	Always	$a_{i2} \le a_{j2}$	always	always
3*l*2*u*	never	$a_{i1} \le a_{j1}$ $b_{i2} \le b_{j2}$	$a_{i1} \le a_{j1}$ $b_{i2} \le b_{j2}$	never	$b_{i2} \ge b_{j2}$	$b_{i2} \ge b_{j2}$	never	$b_{i2} \ge b_{j2}$	$b_{i2} \ge b_{j2}$
3*l*3*u*	$\widetilde{a}_i \le \widetilde{a}_j$	$a_{i1} \le a_{j1}$	$a_{i1} \le a_{j1}$	$a_{i2} \le a_{j2}$	always	Always	$a_{i2} \le a_{j2}$	always	always

8. Conclusion

In this article we give some theoretical facts -in the field of jobs sequences in the systems. They are touch some problems connected with uncertainty of time parameters of systems. It is shown that the problems can be reduce to complete determined case.

References

1. Drummond M. E., *Evaluation and Measurement Techniques for Digital Computer Systems*. N.J.: Prentice-Hall, 1973.
2. Levin V. I., "Scheduling of Computer System Action. The Analysis", *Automatic Control and Computer Sciences* 2, 1983, pp. 64–72.
3. Levin V. I., "Scheduling of Computer System Action. The Synthesis", *Automatic Control and Computer Sciences* 3, 1983, pp. 64–70.
4. Levin V. I., "Discrete Optimization in Conditions of the Interval Indeterminacy", *Automation and Remote Control* 7, 1992, pp. 97–101.
5. Johnson S.M., "Optimal Two-and-Three-Stage Production Schedules with Set-up Times Included", *Naval Research Logistic Quaterly* 1, 1954, pp. 61–68.
6. Muth J. F., Thompson, Gerald L. Industrial Scheduling. N. J.: Prentice-Hall, 1963.
7. Levin V.I., *Strukturno-Logicheskie Metody Issledovaniya Slozhnykh Sistem (Structural-Logi-cal Investigation Methods for Complex Systems).* Moscow: Nauka, 1987 (in Russian).