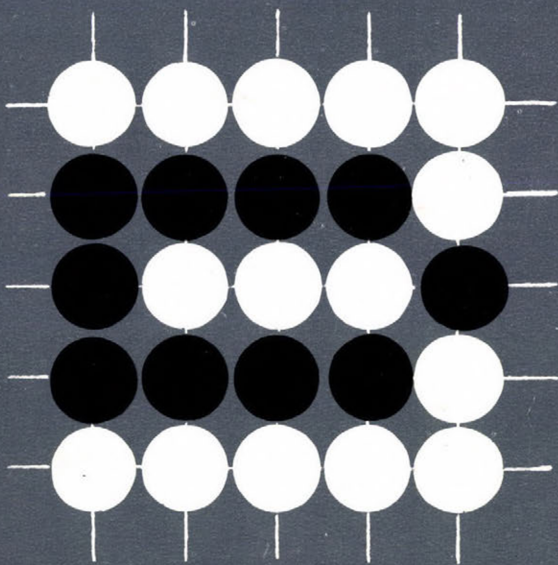
közlemények

27/1982

MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest



MAGYAR TUDOMÁNYOS AKADÉMIA
SZÁMÍTÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZETE

KÖZLEMÉNYEK

1982. JUNIUS

Felelős kiadó:

DR VÁMOS TIBOR
igazgató

ISBN 963 311 141 2
ISSN 0133-7459

82.5033/3 – MSZH Nyomda

TARTALOMJEGYZÉK

CONTENT

Л И Т Е Р А Т У Р А

MTA Számítástechnikai és Automatizálási Kutató Intézete, Közlemények 27/1982

DATA BASE AGGREGATION I

JACEK BAŃKOWSKI

Institute for Scientific, Technical and
Economic Information
Warsaw

1. Introduction

The rapid development of data bases in the last decade yielded quite a variety of data base models as well as models of the "real world". Rather unfortunately, in author's opinion, the latter are very closely related to particular data models, as if their adherents tried to advocate data base designers to look at the real world through the keyhole of data model. In spite of its obvious practical advantages such approach has equally obvious methodological shortcomings: one "real world" model expressed in terms connected with a particular data model can hardly be implemented using another data model. Moreover, and what perhaps is more important, having some real world situation it is almost impossible to argue sensibly that one data model is advantageous to another. The operations of modification, updating and, more generally, of aggregation of data bases need in particular data models (e.g. relational, hierarchical, network-like etc.) some extra information from outside of the model to secure semantical integrity of the result of those operations. Thus, in author's opinion, the claims concerning algebraical properties of query languages underlying particular data models are vastly exaggerated if preserving of semantical correctness is taken into account. E.g. not every formally admitted: join in relational model, inserting a subnet into a network etc. leads to a result still reflecting real world situation. To form a basis for query language design it is necessary to build a model of real world and process of accumulating the knowledge about it; such model enables to describe not only the "syntactical" result of an operation (i.e. the contents of the record(s) representing the result) but also the "semantical" interpretation of input and output data. This paper is an attempt to build such a model.

The following "program postulates" were assumed:

(a) the model of the "real world" should be, as far as it is possible, independent of any particular data structure and/or model

(b) the "real world" consists of a finite number of discrete objects which can be distinguished one from another within the model

(c) the "real world" stays in exactly one "state" (at given moment of time); such "state", which formally is a set of objects is interpreted as the description of the real world; in the process of knowledge accumulation, however, many states are admitted as potential descriptions of the world

(d) the acts of cognition in which only incomplete information on objects is aquired should be representable in the model.

The representation of knowledge in the proposed model is divided into two parts, viz. so called "theory" and "experiment". The first represents the information on restrictions valid in every instance of the real world and the latter the information on properties of some particular "real world" instance. Roughly speaking, they correspond to "a priori" and "a posteriori" knowledge about the world.

2. Definitions and preliminaries

Let X be a finite set of objects of the real world. The powerset of X will be denoted by $\mathcal{P}_o(X)$ and $\mathcal{P}(X)$ will denote the powerset of X minus the empty subset. Any $s \in \mathcal{P}_o(X)$ is referred to as a state. The state $s=[x_{i_1}, x_{i_2}, \ldots, x_{i_n}]$ is interpreted as the possibility of simultaneous occurence of the real world objects $x_{i_1}, x_{i_2}, \ldots, x_{i_n}$. In particular, $s=\emptyset$ represents the possibility of an empty occurence of objects.

Let $\mathbf{S}(X) = \mathcal{P}_o(\mathcal{P}_o(X))$. The knowledge about real world is represented by $S(X) \in \mathbf{S}(X)$. According to the remarks in the previous section any representation of knowledge is composed of two parts, viz. a theory $\overline{S}(X) \in \mathbf{S}(X)$ and an experiment $\underline{S}(X) \in \mathbf{S}(X)$. The fact that $\bar{s} \in \overline{S}(X)$ $(\underline{s} \in \underline{S}(X))$ means that the state $\bar{s}$ ($\underline{s}$) is admitted from the point of view of theory $\overline{S}(X)$ (experiment $\underline{S}(X)$). Accordingly, if $\bar{s} \notin \overline{S}(X)$ ($\underline{s} \notin \underline{S}(X)$) then the state $\bar{s}$ ($\underline{s}$) is forbidden from the point of view of theory $\overline{S}(X)$ (experiment $\underline{S}(X)$); the theory $\overline{S}(X)$ (experiment $\underline{S}(X)$) is equivalent to the statement: "exactly one state $\bar{s}$ ($\underline{s}$) of all $\bar{s} \in \overline{S}(X)$ ($\underline{s} \in \underline{S}(X)$) may correspond to the actual situation in the real world". It therefore follows that the theory $\overline{S}(X) = \emptyset$ (experiment $\underline{S}(X) = \emptyset$) is self-contradictory, as there are no states at all. Such theories (experiments) may occur as the result of aggregation (see below) and indicate that the descriptions of the real world in the form of theories (experiments) are contradictory. The total knowledge about real world $S^c(X) \in \mathbf{S}(X)$, represented by the theory $\overline{S}(X)$ and experiment $\underline{S}(X)$ is computed by means of conclusion operation:

$$S^c(X) = C(\overline{S}(X), \underline{S}(X)) = \overline{S}(X).\underline{S}(X)$$

("." denotes the cross-section of families of subsets)

It should be noted that the conclusion of nonempty theory and experiment may be empty family. Such result occurs iff the theory and experiment are mutually contradictory.

In the sequel the symbol $S(X)$ with no superscript or bar is used to denote either theory or experiment or conclusion when given formulae are valid in either case.

Definition 2.1.

The projection of a state s on the set X, denoted by $s\restriction_X$ is equal to

$$s\restriction_X = s\cap X$$

The projection of a theory (experiment, conclusion accordingly) $S(X_1)$ on the set X_2, denoted by $S(X_1)\restriction_{X_2}$ is equal to

$$S(X_1)\restriction_{X_2} = \sum_{s\in S(X_1)} [\, s\restriction_{X_2}]$$

("$\sum$" denotes the union of families of subsets).

Definition 2.2.

The deductive aggregation of theories (experiments, conclusions) $S_1(X_1)$ and $S_2(X_2)$ is a theory (experiment, conclusion) equal to

$$D(S_1(X_1),\ S_2(X_2)) = [\, s: (\exists s_1\in S_1(X_1))(\exists s_2\in S_2(X_2))((s=s_1\cup s_2)\wedge(s_1=s_2))] =$$
$$= S_1(X_1).S_2(X_2) \qquad (1)$$

Informally, the definition is based upon the policy "admitted is only that what is explicitly admitted by both theories (experiments, conclusions)". Clearly

$$D(S_1(X_1),\ S_2(X_2))\subseteq \mathcal{P}_0(X_1\cap X_2)$$

Definition 2.3.

The inductive aggregation of theories (experiments, conclusions) $S_1(X_1)$ and $S_2(X_2)$ is a theory (experiment, conclusion) equal to

$$I(S_1(X_1),\ S_2(X_2)) =$$

$$= [\ s:\ (\exists s_1 \in S_1(X_1))(\exists s_2 \in S_2(X_2))((s = s_1 \cup s_2) \wedge (s_1|X_1 = s_2|X_2))] \qquad (2)$$

Informally, the definition is based upon the policy "everything is admitted what is not explicitly forbidden by any of theories (experiments, conclusions)". Clearly

$$I(S_1(X_1),\ S_2(X_2)) \subseteq \mathcal{P}_o(X_1 \cup X_2)$$

From the definitions 2.1. - 2.3. the following relationships can be derived:

$$D(S_1(X_1),\ S_2(X_2)) \subseteq I(S_1(X_1),\ S_2(X_2))$$

$$D(S_1(X),\ S_2(X)) = I(S_1(X),\ S_2(X))$$

$$D(S_1(X_1),\ S_2(X_2)) = D(S_2(X_2),\ S_1(X_1))$$

$$I(S_1(X_1),\ S_2(X_2)) = I(S_2(X_2),\ S_1(X_1))$$

$$D(S_1(X_1),\ S_2(X_2))|X_1 \subseteq S_1(X_1)$$

$$I(S_1(X_1),\ S_2(X_2))|X_1 \subseteq S_1(X_1)$$

$$D(D(S_1(X_1),\ S_2(X_2))|X_1,\ D(S_1(X_1),\ S_2(X_2))|X_2) =$$

$$= D(S_1(X_1),\ S_2(X_2))$$

$$I(I(S_1(X_1),\ S_2(X_2))|X_1,\ I(S_1(X_1),\ S_2(X_2))|X_2) =$$

$$= I(S_1(X_1),\ S_2(X_2))$$

$$D(C(\overline{S}_1(X_1),\ \underline{S}_1(X_1)),\ C(\overline{S}_2(X_2),\ \underline{S}_2(X_2))) =$$

$$= C(D(\overline{S}_1(X_1),\ \overline{S}_2(X_2)),\ D(\underline{S}_1(X_1),\ \underline{S}_2(X_2)))$$

$$I(C(\overline{S}_1(X_1), \underline{S}_1(X_1)), C(\overline{S}_2(X_2), \underline{S}_2(X_2))) =$$
$$= C(I(\overline{S}_1(X_1), \overline{S}_2(X_2)), I(\underline{S}_1(X_1), \underline{S}_2(X_2)))$$

The last two equalities constitute in fact the laws of commutativity between deductive (inductive) aggregation and conclusion. Those laws are valuable in practice, as they allow to aggregate theories and experiments separately which is in many cases simpler than aggregating of their conclusions. Indeed, for more "regular" theories and experiments, presented in the next sections, simpler rules of aggregation may be given, while aggregation of conclusions requires, in general, the direct use of (1) or (2).

3. Contineous theories and experiments

So far, no restrictions has been imposed on theories and experiments. Having a $S(X) \subseteq \mathcal{P}_0(X)$ it is impossible to tell wheather it is a theory or an experiment (or a conclusion). The differences between these notions lie in the source of knowledge and have yet no impact on their formal properties. The notions of contineous theory and contineous experiment will be defined after some auxillary definitions and lemmas.

Definition 3.1.

The set of all boundaries is a nonempty subset of families $\mathbb{B} \subset \mathcal{P}(\mathcal{P}_0(X))$ such that

$$(\forall B \in \mathbb{B})(\forall b_1, b_2 \in B)(b_1 \not\subset b_2)$$

i.e. no two elements of a boundary are in the relation of inclusion. The symbol denoting the set of objects is omitted if it leads to no confusion.

Lemma 3.1.

There exists a function $f: \mathbb{S} \dashrightarrow \mathbb{B} \cup [\emptyset]$ such that

$$f(S) \subseteq S$$

and

$$(\forall S \in \mathbb{S})(\forall s \in S)(\exists b \in f(S))(s \subseteq b)$$

(if $S = \emptyset$ then $f(S) = \emptyset$)

i.e. every element of S is covered by an element of f(S).

The value f(S) is referred to as the upper boundary of S.

Lemma 3.2.

There exists a function $g: \mathbb{S} \longrightarrow \mathbb{B} \cup [\emptyset]$ such that

$$g(S) \subseteq S$$

and

$$(\forall S \in \mathbb{S})(\forall s \in S)(\exists b \in g(S))(b \subseteq s)$$

(if $S = \emptyset$ then $g(S) = \emptyset$)

i.e. every element of S covers an element of g(S).

The value g(S) is referred to as the lower boundary of S.

Let $s \in S(X)$; then $Sub(s) \triangleq [\, t: t \subseteq s \,]$ and $Sup(s) \triangleq [\, t: s \subseteq t \subseteq X \,]$

Definition 3.2.

A nonempty theory $\overline{S}(X)$ is contineous iff

$$(\forall \overline{s} \in \overline{S})(\forall t \in Sub(\overline{s}))(t \in \overline{S})$$

Intuitively, if a contineous theory admits simultaneous occurence of objects $x_{i_1}, x_{i_2}, \ldots, x_{i_n}$ then any subset (including empty) of these objects may also occur. Contineous theories are sufficient to deal with many practical applications.

Definition 3.3.

A nonempty experiment $\underline{S}(X)$ is contineous iff

$$(\forall \underline{s} \in \underline{S})(\forall t \in Sup(\underline{s}))(t \in \underline{S})$$

Intuitively, if a contineous experiment admits simultaneous occurence of objects $x_{i_1}, x_{i_2}, \ldots, x_{i_n}$ then any broader subset of X may also occur. It is rather difficult to give an example of experimental procedure that could not be represented by a contineous experiment.

Lemma 3.3.

There exists inverse function f^{-1} (g^{-1}) iff the theory $\overline{S}$ (experiment $\underline{S}$) is contineous. The inverse functions are given by

$$f^{-1}(B) = \sum_{b \in B} \mathrm{Sub}(b)$$

and

$$g^{-1}(B) = \sum_{b \in B} \mathrm{Sup}(b)$$

Corollary

Upper (lower) boundary $\overline{B} = f(\overline{S})$ ($\underline{B} = g(\underline{S})$) is the unique representative of contineous theory $\overline{S}$ (experiment $\underline{S}$). The proof follows from lemmas 3.1. (3.2.) and 3.3..

As the theory $\overline{S}$ and experiment $\underline{S}$ can be represented by the corresponding boundaries $\overline{B}$ and $\underline{B}$ the conclusion $C(\overline{S}, \underline{S}) = C^{B}(\overline{B}, \underline{B})$ is equal to

$$C^{B}(\overline{B}, \underline{B}) = [s: (\exists \overline{b} \in \overline{B})(\exists \underline{b} \in \underline{B})(\underline{b} \subseteq s \subseteq \overline{b})]$$

Theorem 3.1.

The deductive (inductive) aggregation of contineous theories (experiments) is a contineous theory (experiment).

The theorem follows from the definitions of aggregation (2.2. or 2.3.) and contineous theories (3.2.) or experiments (3.3.). The results of aggregation may be summarized in the following way:

$$D^{B}(\overline{B}_1(X_1), \overline{B}_2(X_2)) = f\Big(\sum_{s \in \overline{B}_1(X_1)} \sum_{t \in \overline{B}_2(X_2)} [s \cap t]\Big)$$

$$D^{B}(\underline{B}_1(X_1), \underline{B}_2(X_2)) = g\Big(\sum_{s \in \underline{B}_1(X_1)} \sum_{t \in \underline{B}_2(X_2)} [s|X_2 \cup t|X_1]\Big)$$

$$I^{\mathcal{B}}(\overline{B}_1(X_1), \overline{B}_2(X_2)) = f(\sum_{s\in\overline{B}_1(X_1)} \sum_{t\in\overline{B}_2(X_2)} [s\setminus X_2 \cup s\cap t \cup t\setminus X_1])$$

$$I^{\mathcal{B}}(\underline{B}_1(X_1), \underline{B}_2(X_2)) = g(\sum_{s\in\underline{B}_1(X_1)} \sum_{t\in\underline{B}_2(X_2)} [s\cup t])$$

4. Theories and experiments of existential type (E-type)

In this section is proposed yet another form of representation of theories and experiments. It is shown that some contineous (and only contineous) theories and all contineous (and only contineous) experiments can be represented in that form.

Definition 4.1.

A theory $\overline{S}(X)$ is of existential type (E-type) iff

$$(\exists \overline{E}\subset\mathcal{P}(X))(\overline{S}(X) = [\,\overline{s}: (((\forall\overline{e}\in\overline{E})(\mathrm{card}(\overline{e}\cap\overline{s})\leq 1))\wedge\overline{s}\subseteq Z)])$$

where $Z = \bigcup_{\overline{e}\in\overline{E}} \overline{e}$.

Remark 4.1.

Every subset $\overline{e}\in\overline{E}$ may be interpreted as a statement:"there may exist at most one object from subset $\overline{e}$ ". The family $\overline{E}$ may be interpreted as a conjuntion of such statements. The objects from X that do not belong to Z cannot exist.

Lemma 4.1.

It is sufficient to consider only such families $\overline{E}$ for which

$$(\forall\overline{e}_1, \overline{e}_2\in \overline{E})(\overline{e}_1\not\subset\overline{e}_2) \qquad (3)$$

and

$$(\forall\overline{e}\in\overline{E})(1\leq\mathrm{card}(\overline{e})\leq 2) \qquad (4)$$

Indeed, if e.g. $\overline{e}_1\subset\overline{e}_2$ then

$$(\forall\overline{s}\in\overline{S})(\mathrm{card}(\overline{e}_2\cap\overline{s})\leq 1 \Rightarrow \mathrm{card}(\overline{e}_1\cap\overline{s})\leq 1)$$

and (3) is proved, i.e. adding $\overline{e}_1$ to $\overline{E}$ would not change $\overline{S}$. To prove the validity of (4) one should note that if there exists $\overline{e}_o \in \overline{E}$ such that $card(\overline{e}_o)=m>2$ then there exists family

$$\overline{E}' = \overline{E} \setminus [\overline{e}_o] + [\overline{e}_o \setminus [x_1]] + \sum_{x_2 \in \overline{e}_o \setminus x_1} [[x_1, x_2]]$$

("+" denotes union of families) and

$$W1 = (\forall \overline{e} \in \overline{E})(card(\overline{e} \cap \overline{s}) \leq 1) \equiv ((\forall \overline{e} \in \overline{E} \setminus [\overline{e}_o])(card(\overline{e} \cap \overline{s}) \leq 1)) \wedge$$
$$\wedge (card(\overline{e}_o \cap \overline{s}) \leq 1)$$

$$W2 = (\forall \overline{e} \in \overline{E}')(card(\overline{e} \cap \overline{S}) \leq 1) \equiv ((\forall \overline{e} \in \overline{E} \setminus [\overline{e}_o])(card(\overline{e} \cap \overline{s}) \leq 1)) \wedge$$
$$\wedge (card((\overline{e}_o \setminus [x_1]) \cap \overline{s}) \leq 1) \wedge ((\forall x_2 \in (\overline{e}_o \setminus [x_1]))(card([x_1, x_2] \cap \overline{s}) \leq 1))$$

For every $x_1 \in \overline{e}_o$ the last two terms are equivalent to $card(\overline{e}_o \cap \overline{s}) \leq 1$ and, finally, $W1 \equiv W2$ for every $\overline{s} \in \overline{S}$, i.e. every subset of cardinality $m>2$ may be replaced by a subset of cardinality m-1 (and several subsets of cardinality 2).

Remark 4.2.

According to lemma 4.1. every E-type theory can be represented by a family $\overline{E}$ of the form

$$\overline{E} = \overline{E}_1 + \overline{E}_2$$

where

$$\overline{E}_1 = \sum_{x \in X_1} [[x]] \quad ; \quad X_1 \subseteq X$$

and

$$\overline{E}_2 \subseteq \sum_{x_1 \in X_2} \sum_{x_2 \in X_2 \setminus x_1} [[x_1, x_2]] \quad ; \quad X_2 = X \setminus X_1$$

If $X_1 = X$ then the E-type theory is the least restrictive, i.e. it allows the existence of every combination of objects from X. If $X_2 = X$ and in the family $\overline{E}_2$ occur all two-element subsets of X then the E-type theory is the most restrictive, i.e. it allows the existence of at most one object from X.

Theorem 4.1.

Every theory of E-type is contineous but not conversely.

Indeed, if a theory $\overline{S}$ is of E-type and e.g. $\overline{s}_2 \in \overline{S}$ then $\overline{s}_2 \in Z$ and $(\forall \overline{e} \in \overline{E})(\mathrm{card}(\overline{e} \cap \overline{s}_2) \leq 1)$; it therefore follows that for every $\overline{s}_1 \subseteq \overline{s}_2$ $\overline{s}_1 \subseteq Z$, $(\forall \overline{e} \in \overline{E})(\mathrm{card}(\overline{e} \cap \overline{s}_1) \leq 1)$ and, consequently, $\overline{s}_1 \in \overline{S}$.

The converse theorem is obviously not true. The cardinality of different contineous theories is equal to the cardinality of the set of upper boundaries, which is equal to the cardinality of the set of families $\overline{E}$ satisfying condition (3) in lemma 4.1.. But, according to (4) in lemma 4.1. , every theory of E-type which is represented by a family $\overline{E}$ containing subsets of cardinality greater than 2 can be represented equivalently by a family $\overline{E}$ containing subsets of cardinality not greater than 2 , only.

For example, contineous theory represented by the upper boundary $\overline{B} = [[1, 2], [1, 3], [2, 3]]$ for the set of objects $X = [1, 2, 3]$ is not of E-type. The only possible (without symmetric modifications) E-type theories in this case, according to remark 4.2. , are listed below with corresponding upper boundaries:

$\overline{E} = [[1], [2], [3]]$	$\overline{B} = [[1, 2, 3]]$
$\overline{E} = [[1], [2, 3]]$	$\overline{B} = [[1, 2], [1, 3]]$
$\overline{E} = [[1, 2], [1, 3]]$	$\overline{B} = [[1], [2, 3]]$
$\overline{E} = [[1, 2], [1, 3], [2, 3]]$ $[[1, 2, 3]]$	$\overline{B} = [[1], [2], [3]]$

Intuitively, one may note that the contineous theory represented by $\overline{B} = [[1, 2], [1, 3], [2, 3]]$ is equivalent to the statement: "there may exist at most two objects from the set [1, 2, 3]" which, as it occurs, cannot be represented equivalently by a conjunction of statements of the type: "there may exist at most one object from ...".

Definition 4.2.

An experiment $\underline{S}(X)$ is of E-type iff

$$(\exists \underline{E} \subset \mathcal{P}(X))(\underline{S}(X) = [\underline{s}: (((\forall \underline{e} \in \underline{E})(\text{card}(\underline{e} \cap \underline{s}) \geqslant 1)) \wedge (\underline{s} \subseteq X))])$$

Remark 4.3.

Every subset $\underline{e} \in \underline{E}$ may be interpreted as a statement: "there exists at least one object from the subset $\underline{e}$". The family $\underline{E}$ may be interpreted as a conjunction of such statements.

Lemma 4.2.

It is sufficient to consider only such families $\underline{E}$ for which

$$(\forall \underline{e}_1, \underline{e}_2 \in \underline{E})(\underline{e}_1 \not\subset \underline{e}_2) \qquad (5)$$

Indeed, if e.g. $\underline{e}_1 \subset \underline{e}_2$ then

$$(\forall \underline{s}\ \underline{S})(\text{card}(\underline{e}_1 \cap \underline{s}) \geqslant 1 \Rightarrow \text{card}(\underline{e}_2 \cap \underline{s}) \geqslant 1)$$

and (5) is proved, i.e. adding $\underline{e}_2$ to $\underline{E}$ would not change $\underline{S}$.

Theorem 4.2.

Every experiment of E-type is contineous and conversely.

Indeed, if experiment $\underline{S}(X)$ is of E-type and e.g. $\underline{s}_1 \in \underline{S}$ then $\underline{s}_1 \subseteq X$ and $(\forall \underline{e} \in \underline{E})(\text{card}(\underline{e} \cap \underline{s}_1) \geqslant 1)$; it therefore follows that for every $\underline{s}_1 \subseteq \underline{s}_2 \subseteq X$ $\underline{s}_2 \subseteq X$, $(\forall \underline{e} \in \underline{E})(\text{card}(\underline{e} \cap \underline{s}_2) \geqslant 1)$ and, consequently, $\underline{s}_2 \in \underline{S}(X)$.

The converse theorem is also true. Consider a function:

$h: \mathbb{B} \rightarrow \mathcal{P}(X)$

$$h(\underline{B}) = g([p: ((\forall \underline{b} \in \underline{B})(\mathrm{card}(p \cap \underline{b}) \geqslant 1)) \wedge (p \subseteq X)]) \qquad (6)$$

It can be seen that the function h is determined for every $\underline{B} \in \mathbb{B}$. Moreover, $(\forall p_1, p_2 \in h(\underline{B}))(p_1 \not\subset p_2)$ what coincides with condition (5) in lemma 4.2.. Finally, if $\underline{B}$ represents $\underline{S}$ i.e. $\underline{S}(X) = \bigcup_{\underline{b} \in \underline{B}} \mathrm{Sup}(\underline{b})$ and

$$\underline{S1}(X) = [\underline{s}_1: ((\forall p \in h(\underline{B}))(\mathrm{card}(p \cap \underline{s}_1) \geqslant 1) \wedge (\underline{s}_1 \subseteq X)]$$

then for $\underline{B} = [\underline{b}]$ and $\underline{b} = \emptyset$, $h(\underline{B}) = \emptyset$ and $\underline{S1}(X) = \mathcal{P}_o(X) = \mathrm{Sup}(\underline{b}) = \underline{S}(X)$

Otherwise (from (6))

$$\underline{S1}(X) = [\underline{s}_1: (\exists \underline{b} \in \underline{B})(\underline{s}_1 \in \mathrm{Sup}(\underline{b})] = \bigcup_{\underline{b} \in \underline{B}} \mathrm{Sup}(\underline{b}) = \underline{S}(X)$$

From the proof it follows that $h(\underline{B})$ is equal to the family $\underline{E}$ representing the experiment described by $\underline{B}$. To compute $h(\underline{B})$ the following iterative procedure can be used:

1. If $\mathrm{card}(\underline{B}) = 1$ (i.e. $\underline{B} = [\underline{b}]$) then
   $$h(\underline{B}) = \sum_{x \in \underline{b}} [[x]]$$
2. If $\mathrm{card}(\underline{B}) > 1$ then
   $$h(\underline{B}) = g(\sum_{p \in h(\underline{B} \setminus [\underline{b}])} \ \sum_{q \in h([\underline{b}])} [p \cup q])$$

It is worth noting that $h^{-1} = h$ i.e. $\underline{B} = h(h(\underline{B}))$ $(\underline{B} \neq [\emptyset])$.

Theorem 4.3.

The deductive (inductive) aggregation of E-type theories (experiments) is an E-type theory (experiment).

The theorem follows from the definitions of aggregation (2.2 or 2.3.) and E-type theories (experiments). The results of aggregation may be sunnarized in the following way:

$$D(\overline{E}_1(X_1), \overline{E}_2(X_2)) = f((\overline{E}_1(X_1)|_{Z_2} + \overline{E}_2(X_2)|_{Z_1}) \setminus [\emptyset])$$

where $Z_1 = \bigcup_{\overline{e} \in \overline{E}_1} \overline{e}$, $Z_2 = \bigcup_{\overline{e} \in \overline{E}_2} \overline{e}$; the family containing empty subset is subtracted as the result cannot contain such element (see def. 4.1.) and such families may appear during projections of $\overline{E}_1$ and $\overline{E}_2$.

$$D(\underline{E}_1(X_1), \underline{E}_2(X_2)) = g((\underline{E}_1(X_1)|_{X_2} + \underline{E}_2(X_2)|_{X_1}) \setminus [\emptyset])$$

$$I(\overline{E}_1(X_1), \overline{E}_2(X_2)) = f(\overline{E}_1(X_1) + \overline{E}_2(X_2))$$

$$I(\underline{E}_1(X_1), \underline{E}_2(X_2)) = g(\underline{E}_1(X_1) + \underline{E}_2(X_2))$$

ÖSSZEFOGLALÁS

AGGREGÁTUMOK ADATBÁZISOKBAN

A dolgozat a "valós világ" egy, az adatbázisok tervezése során használható modelljére tesz javaslatot. A közölt modell a következő tulajdonságokkal bir:

a/ minden konkrét adat modelltől független
b/ véges számu objektumot tartalmaz
c/ a "valós világ"-nak egy állapota van. Ez az állapot a valós világ leirásaként értelmezhető.
d/ a modell alkalmas nem teljes információ ábrázolására.

Az ismeretek reprezentálása két részre osztható /"elmélet" és "gyakorlat"/. Ezek a fogalmak nagyjából az "a priori" és "postpriori" tudásnak felelnek meg.

Р е з ю м е

В статье представляется модель " реального мира " для пользования в строении баз данных, которая обладает следующими стройствами:

а/ модель " реального мира " независима от какой-то определенной модели данных
б/ " реальный мир " состоит из конечного числа объектов
в/ " реальный мир " находится в точно одном " состоянии ", которое интерпретуется как описание " реального мира "
г/ неполная информация имеет отображение в модели

Представление знания разделяется на так называемые " теорию " и " эксперимент ", которые вообще говоря соответствуют " априоричному " и " апостериоричному " знаниям.

MTA Számítástechnikai és Automatizálási Kutató Intézete, Közlemények 27/1982

LOGICAL AND STRUCTURAL INVESTIGATIONS OF THE RELATIONAL DATA MODEL

J. DEMETROVICS

The concepts data base and data base management system have a central role in the computer aided information service and retrieval. Large masses of data representing complicated relationships are impossible to view on the machine level. The user may have but a generalized view of the wars of data stored by the computer and of its structure. The database management system is the software tool that establishes the link between the machine and the user's "general" level and thus permits to generate and conduct the storage, updating and retrieval of data on a logically higher level.

There are several ways to classify database management systems, the most generally accepted going by the way data and links between them are represented for the user. From the several data models proposed up to now three have had a relatively general acceptance and practical use: the hierarchical, the net and the relational models. With respect to its possiblities in future use the relational one (introduced by E.F.Codd) looks like one of the most promising data management tools.

In the relational approach data links are represented in the n-tuples of data. The relational model's advantage is not making prefound use of the machine representation ways, but representing data in a user conceivable form. It is a means apt to describe the logical structure of data bases as well. Data in it are stored in matrix form. A matrix stands for a unit of a Codd's third normal form relational data base, with its rows being the data records and its columns the data attributes. So this unit of the relational data base can be stored as a flat file.

The exact definition is the following:
Let Ω be a nonvoid finite set ($\Omega=\{a_1,a_2,\dots,a_n\}$). A finite set of unary functions over this set is called a relation. These are depicted as two

dimensional arrays: if R is a relation over Ω and $R=\{h_1,h_2,\dots,h_k\}$ where every h_i has arity n, the table of this relation is

	a_1	a_2	...	a_n
h_1	$h_1(a_1)$	$h_1(a_2)$	...	$h_1(a_n)$
h_2	$h_2(a_1)$	$h_2(a_2)$	...	$h_2(a_n)$
..	...	...	...	...
..	...	...	...	...
h_k	$h_k(a_1)$	$h_k(a_2)$	...	$h_k(a_n)$

Ω in this table is the set of attributes: the elements of R (i.e. the rows of the table) are essentially the records of data. These can have no repetitions in the relation, as the latter is defined as a set.

The relational data model has two main theoretical aspects. One of them is finding ways to discover and maintain links between data which are adequate to the structure of the relational data model. The two principal methods to describe links of data in an abstract way that these investigations have yielded use the concept of functional dependences and that of intersection dependences resp. The other theoretical aspect concerns the investigation of query methods obtainable to data bases constructed according to the relational data model. It is well known, that (commercially available) relational data bases are incomparably slower in queries than hierarchical or net data bases, especially if the forms of future queries can be obtained previously. Still, query forms cannot always be preassigned to data bases which gives to the relational data base good future chances if an effective and to the user demands well compatible query language is defined and the relational data base management system is properly organised which are the two basic tasks for the immediate future.

In the present paper the functional dependency and three analogous concepts are treated then the problems of the query operations [1], [5], [24] are considered.

For an effective data retrieval interconnections among data have to be dealt with properly. Functional dependencies (as introduced by E.F. Codd [9]) is an important tool for taking interconnections among data into consideration in relational data bases [3].

<u>Definition 1:</u> Let Ω be a set of attributes and R a relation over it. $A, A \subseteq \Omega$ <u>functionally depends</u> on $B, B \subseteq \Omega$ iff

$$(\forall h,g \in R)((\forall a \in A)\ (h(a) = g(a)) \Rightarrow (\forall b \in B)\ (h(b) = g(b))),$$

This is denoted by $A \overset{f}{\underset{R}{\rightarrow}} B$ and heuristically means that determining the attribute values on A leaves no choice as to the attribute values on B.

Let R be a relation over Ω. We denote by F_R the set of its functional dependencies, i.e.

$$F_R = \{(A,B) : A \subseteq \Omega,\ B \subseteq \Omega,\ A \overset{f}{\underset{R}{\rightarrow}} B\}.$$

The set F_R is called the full family of functional dependencies in the relation R and much investigated because knowing only this of a relation permits to design its structuring in a relational data base in a concise and (in memory requirements) economical way.

Other types of data dependencies in a relation R are quite possible, of which we mention but three the dual, strong and weak dependencies, [10] of which the full families will be denoted by D_R, S_R and W_R

<u>Definition 2:</u> Let R be a relation over the attribute set let A and B be subsets of Ω.
B depends on A <u>dually</u> in R iff

$$(\forall h,g \in R)\ ((\exists a \in A)\ (h(a) = g(a)) \Rightarrow (\exists b \in B)\ (h(b) = g(b))),$$

B depends on A <u>strongly</u> in R iff

$$(\forall h,g \in R)\ ((\exists a \in A)\ (h(a) = g(a)) \Rightarrow (\forall b \in B)\ (h(b) = g(b))).$$

B depends on A <u>weakly</u> in R iff

$$(\forall h,g\in R)\ ((\forall a\in A)\ (h(a) = g(a)) \Rightarrow (\exists b\in B)\ (h(b) = g(b))).$$

By the logical structure of a relation we shall mean the full families F_R, D_R, S_R, W_R.

Knowing its dual and weak dependencies may greatly increase the efficiency of data retrieval from a relational data base when only partial information is required or when the user doesn't know all the attribute values necessary for the retrieval.

The knowledge of the strong dependencies is useful in the decomposition of big relations into smaller ones and thus helps in the reduction of the space requirements of the storage. The way of decomposition is analogous to how in the case of functional dependencies it is made [9], [12].

A basic problem in the theory of relational data bases is to characterize functional dependencies in a self contained way, i.e. to axiomatize them. A basic result of W.W. Armstrong [3] is to give an axiomatization of full f-families. This system of axioms is however, a little unpractical in handling important combinatorical problems of full f-families ([5], [6], [7]).

In [6] one can find a modification of Armstrong's axioms which faciliates handling certain types of combinatorical problems ([5], [19]). One of the purposes of the present paper is to give an axiomatization of full f-families based on their combinatorial properties. Next two theoretical problems of full families are discussed [15] and linear relations described [14]. A description of a factual system is given by the relational data model. A duality principle is stated between the functional and dual full families, the systems of axioms (with similar structures) are given for the full f-, d- and s-families and full w-families with no subset of attributes dependent on the void set (Theorem 1).

For the sake of completeness we give the axiom systems given for full f-, d- and s-families in papers [3] and [10]: let $Z \subseteq 2^{\Omega} \times 2^{\Omega}$ and $A,B,C,D \subseteq \Omega$.

The φ axioms are:

(F1) $(A,A) \in Z$;

(F2) if $(A,B) \in Z$ and $(B,C) \in Z$ then $(A,C) \in Z$;

(F3) if $(A,B) \in Z$ and $C \supseteq A$, $D \subseteq B$ then $(C,D) \in Z$;

(F4) if $(A,B) \in Z$ and $(C,D) \in Z$ then $(A \cup C, B \cup D) \in Z$.

The ν axioms are:

(D1) $(A,A) \in Z$;

(D2) if $(A,B) \in Z$ and $(B,C) \in Z$ then $(A,C) \in Z$;

(D3) if $(A,B) \in Z$ and $C \subseteq A$, $B \subseteq D$ then $(C,D) \in Z$;

(D4) if $(A,B) \in Z$ and $(C,D) \in Z$ then $(A \cup C, B \cup D) \in Z$;

(D5) if $(A,\emptyset) \in Z$ then $A \neq \emptyset$.

The γ axioms are:

(S1) $(\forall a \in \Omega)$ $(\{a\}, \{a\}) \in Z$;

(S2) if $(A,B) \in Z$ and $(B,C) \in Z$ and $B \neq \emptyset$ then $(A,C) \in Z$;

(S3) if $(A,B) \in Z$ and $C \subseteq A$, $D \subseteq B$ then $(C,D) \in Z$;

(S4) if $(A,B) \in Z$ and $(C,D) \in Z$ then $(A \cap C, B \cup D) \in Z$;

(S5) if $(A,B) \in Z$ and $(C,D) \in Z$ then $(A \cup C, B \cap D) \in Z$;

Next we give new systems of axioms of a new pattern for the full s-, d-, and f-families, then their analogous for weakly dependent families containing no subset of the attributes dependent on the void set (see Theorem 3). Let $Z \subseteq 2^{\Omega} \times 2^{\Omega}$ then Z satisfies the corresponding systems of axioms iff the following conditions hold:

The F axioms are:

$\forall(X,Y)\in(P(\Omega)\times P(\Omega)\setminus Z)\ \ \exists E\subseteq\Omega$ such that

(i) $X\subseteq E$ and $Y\not\subseteq E$;

(ii) if $(A,B)\in Z$ and $A\subseteq E$ then $B\subseteq E$ holds.

The D axioms are:

$\forall(X,Y)\in(P(\Omega)\times P(\Omega)\setminus Z)\ \ \exists E\subseteq\Omega$ such that

(i) $X\cap E\neq\emptyset$ and $Y\cap E=\emptyset$;

(ii) if $(A,B)\in Z$ and $A\cap E\neq\emptyset$ then $B\cap E\neq\emptyset$ holds.

The S axioms are:

$\forall(X,Y)\in(P(\Omega)\times P(\Omega)\setminus Z)\ \ \exists E\subseteq\Omega$ such that

(i) $X\cap E\neq\emptyset\quad Y\not\subseteq E$;

(ii) $(A,B)\in Z\quad A\cap E\neq\emptyset\ ,\ B\subseteq E$

The W axioms are:

$\forall(X,Y)\subseteq(P(\Omega)\times P(\Omega)\setminus Z)\ \ \exists E\subseteq\Omega$ such that

(i) $X\subseteq E$ and $Y\cap E=\emptyset$;

(ii) $(A,B)\in Z$ and $A\subseteq E$ then $B\cap E\neq\emptyset$ holds.

Theorem 1: The φ, ν and γ systems of axioms are equivalent with the F, D and S systems, respectively.

Nest we shall define equality sets of matrices and show them to be characterized by the property that the 3 equality sets determined by 3 rows are a Δ system. (Theorem 2). Then we give some reason why the F,D,S and W axiom systems have such similar forms.

Definition 3: Let g,h be two rows of a relation R over Ω. The equality set of the rows g and h is

$$E(h,g)=\{a\in\Omega : h(a)=g(a)\}.$$

The equality set of the relation R (i.e. the matrix R) is defined as

$\varepsilon(h,g) = \{a\in\Omega : h(a) = g(a)\}$.

Definition 4: A class of sets is said to be a Δ-system if for any $A\neq B$, $C\neq D$ of its sets $A\cap B=C\cap D$.

Theorem 2: Let f,g,h be rows of the relation R. Then the class of sets $\{E(f,g), E(g,h), E(f,h)\}$ is a Δ-system.

Let $\varepsilon = \{E_{i,j} : 1 \leq i < j \leq k\}$ a class of subsets of Ω for which $\{E_{i,j}, E_{i,l}, E_{j,l}\}$ is a Δ-system for any $1 \leq i < j < l \leq k$. Then a relation R over Ω can be constructed with $\varepsilon_R=\varepsilon$. Theorem 2 permits a new formulation of the F,D,S and W-axioms which are equivalent with the old ones except for the W case (Theorem 3).

Let $Z \subseteq 2^\Omega \times 2^\Omega$ and E be an arbitrary class of sets

$\{E_{i,j} : 1 \leq i < j \leq k, E_{i,j} \subseteq \Omega\}$.

The F' axiom is:

for Z there are such k and E that

(i) if $(X,Y)\in P(\Omega)\times P(\Omega)\setminus Z$ then there are such $i,j (1 \leq i < j \leq k)$ that $X \subseteq E_{i,j}$ and $Y \not\subseteq E_{i,j}$;

(ii) if $(A,B)\in Z$ and $A \subseteq E_{i,j}$ then $B \subseteq E_{i,j}$ with $1 \leq i < j \leq k$;

(iii) if for any i,j,ℓ $1 \leq i < j < \ell \leq k$ $\{E_{i,j},E_{i,l},E_{j,l}\}$ is Δ-system.

The D' axiom is:

for Z there are such k and E that

(i) if $(X,Y)\in P(\Omega) \times P(\Omega)\setminus Z$ then there are such i,j $(1 \leq i < j \leq k)$, if $X \cap E_{i,j} \neq \emptyset$ and $Y \cap E_{i,j} = \emptyset$;

(ii) if $(A,B)\in Z$ and $A \cap E_{i,j} \neq \emptyset$ then $B \cap E_{i,j} \neq \emptyset$ with $1\leq i<j\leq k$;

(iii) if for any i,j,ℓ $(1 \leq i < j < \ell \leq k)$ $\{E_{i,j},E_{i,l},E_{j,l}\}$ is Δ-system.

<u>The S' axiom is:</u>

for Z there are such k and E that

(i) if $(X,Y)\in P(\Omega) \times P(\Omega)\setminus Z$ then there are such i,j $(1 \leq i < j \leq k)$ that $X \cap E_{i,j} \neq \emptyset$ and $Y \nsubseteq E_{i,j}$;

(ii) if $(A,B)\in Z$ and $A \cap E_{i,j} \neq \emptyset$ then $B \subseteq E_{i,j}$ with $1 \leq i < j \leq k$;

(iii) for any i,j,ℓ $(1 \leq i < j < \ell \leq k)$ $\{E_{i,j}, E_{i,l}, E_{j,l}\}$ is Δ-system.

<u>The W' axiom is:</u>

for Z there are such k and E that

(i) if $(X,Y)\in P(\Omega) \times P(\Omega)\setminus Z$ then there are such i,j $(1 \leq i < j \leq k)$ that $X \subseteq E_{i,j}$ and $Y \cap E_{i,j} = \emptyset$;

(ii) if $(A,B)\in Z$ and $A \subseteq E_{i,j}$ then $B \cap E_{i,j} \neq \emptyset$ with $1 \leq i < j \leq k$;

(iii) for any i,j,ℓ $(1 \leq i < j < \ell \leq k)$ $\{E_{i,j}, E_{i,l}, E_{j,l}\}$ is Δ-system.

<u>Theorem 3:</u> The F', D', S' axioms are equivalents to the F, D and S axioms respectively. The W' axioms are definitely stronger then the W axioms.

The cause of the last statement is that W-axioms are meaningless for set pairs of the form $(\emptyset,B)$.

Theorem 4 states, that F',D',S' and W'-axioms characterize f-, d-, s- and w-families.

<u>Theorem 4:</u> Let $Z \in 2^{\Omega} \times 2^{\Omega}$ have one of the properties F,D,S,W and let Y' denote the corresponding set of axioms F',D',S' or W'. Then Z statisfies the Y'-axioms iff a relation R over Ω exists for which $Y_R = Z$ holds. (Y_R is the set of the dependencies of the kind in point in the relation.)

Next we mention two combinatorial problems.

The first is to give the minimal number of rows in a relation which represents any given full f-family (or antichain) of an n-element set as its set of functional dependencies (or minimal keys, respectively) ([15], [18]). These minimal numbers of rows are denoted by s(n) and S(n) for the two problems above. In [15] the bounds

$$\sqrt{2\binom{n}{[n/2]}} \leq s(n) \leq 2\binom{n}{[n/2]}$$

were proved; a more recent result of the author is

Theorem 5:

$$\frac{1}{n^2}\binom{n}{[n/2]} \leq s(n) \leq \binom{n}{[n/2]} + 1 \quad \text{and}$$

$$\frac{1}{n^2}\binom{n}{[n/2]} \leq S(n) \leq \frac{3}{2}\binom{n}{[n/2]} .$$

The upper bound can be found as a byproduct of characterising generator sets of full f-families by their maximal dependent attribute subsets' intersection irreducible sets.

Next we metion theorems about linear relations. These are relations with rational attribute values, the rows of which are a linearly closed subset of the vector field $Q^{|\Omega|}$.
Of course a linear relation is always characterized by a finite subset of its rows.

Let $R \subseteq Q^{|\Omega|}$ be a linear relation; then

Theorem 6: Every dependence $(A,B) \in F_R$ in R is linear, i.e. is given by a linear operator in $Q^{|\Omega|}$.

Theorem 7: All the minimal keys in R have the same cardinality k where k can take an arbitrary value between 1 and $|\Omega|$.

The problem of axiomatizing full f-families of linear relations is equivalent with the (internal) characterization of coordinatable matroids over

Q and is therefore open.

A practical example for illustrating the efficiency of the relational data base structure is the two major parts of the Maze and Industial Plants Producing Branch of the Nâdudvar Red Star Co-op data base which deals with stock and demand registering with special tasks for most of the subplants.

The investigation of but the functional dependencies proved sufficient to reduce storage area requirements of the system by about 40%. (Dual and weak dependencies were not found in this system.)

This task put forth the problem of efficient queries in the system which were interpreted as special tables (see [2], [24]). In [1] an efficient simplification algorithm for a large class of queries is given which improves the response time of the system which can in some cases quite long due to the difficulties that lie in the execution of the relation join operation.

R E F E R E N C E S

[1] Aho, A.V., Saigv, Y., Ullman, J.D.: Efficient Optimization of a Class of Relational Expressions, ACM Trans. Database Systems 4. (1979) 4, 435-454.

[2] Aho, A.V., Sagiv, V., Ullman, J.D.: Equivalences among relational expressions, SIAM J. Comput., 8 (1979), 218-246.

[3] Armstrong, W.W.: Dependency structures of data base relationship, Information Processing 74, North-Holland Publ. Co. (1974), 580-585.

[4] Armstrong, W.W.: On the generation of dependency structures of relational data bases, Publication 272, Universite de Montreal (1977)

[5] Beeri, C., Berstein, P.A.: Computational Problems Related to the Design of Normal Form Relational Schemas, ACM Trans. on Database Systems, 4, (1979) 1, 30-59.

[6] Békéssy, A., Demetrovics, J.: Contribution to the theory of data base relations, Discrete Math. 27 (1979), 1-10.

[7] Békéssy, A., Demetrovics, J., Hannák, L., Frank, P., Katona, Gy.: On the number of maximal dependencies in a data base relation of fixed order, Discrete Math. 30 (1980) 83-88.

[8] Codd, E.F.: A relational model for large shared data banks, Comm. ACM 13 (1970) 377-387.

[9] Codd, E.F.: Further normalization of the data base relational model, Data Base Systems, R. Rustin, ed. Prentice-Hall, Englewood Cliffs, NJ, (1972) 33-64.

[10] Czédli, G.: Függőségek relációs adatbázis modellben, Alk. Mat. Lapok (1980)

[11] Demetrovics, J.: On the number of candidate keys, Informations Processing Letters 7 (1978) 6, 266-269.

[12] Demetrovics, J.: Relációs adatbázis modell, MTA SZTAKI Közlemények 20 (1978) 21-23.

[13] Demetrovics, J.: Homogén file kulcsairól, Alkalmazott Matematikai Lapok 3 (1977) 185-191.

[14] Demetrovics, J.: On the equivalence of candidate keys with Sperner systems, Acta Cybernetica 4 (1979) 3, 247-252.

[15] Demetrovics, J.: Candidate keys and antichains, SIAM J. Alg. Disc. Meth. 1 (1980) 1,92.

[16] Demetrovics, J.: O klucsah odnonarodnüh fãjlov, Kibernetika (Kiev) (to appear).

[17] Demetrovics, J., Gyepesi, Gy.: On the Functional Dependency and Some Generalization of it, Acta Cybernetica (to appear)

[18] Demetrovics, J., Gyepesi, Gy.: A note on candidate keys and full families, SIAM J. Alg. Disc. Meth. (to appear)

[19] Fagin, R.: Functional dependencies in a relational database and propositional logic, IBM J. Res. and Develop. 21 (1977) 6, 534-544.

[20] Kleitman, D.: On a combinatorial problem of Erdős, Proc. AMS (1966) 139-141.

[21] Maier, D., Mendelzon, A.O., Sagiv, Y.: Testing Implications of Data Dependencies, ACM Trans. Database Systems 4 (1974) 4, 455-469.

[22] Rissanen, J.: Independent components of relations, ACM Trans. Database Syst. 2 (1977) 4, 317-325.

[23] Sperner, E.: Ein Satz über Untermengen einer endlichen Menge, Mathematische Zeitschrift 27 (1928) 544-548.

[24] Zloof, M.M.: Query-byexample: A data base language, IBM Syst.J. 16 (1977) 4 324-343.

Ö s s z e f o g l a l ó

A relációs adatmodel logikai és strukturális vizsgálata

Demetrovics János

Ebben a cikkben a relációs adatmodellben definiálható fügésekkel foglalkozunk, pontosabban a funkcinális függéssel és három analogonjával: a duális, erős és gyenge függésekkel. Adott tipusú függés vizsgálatakor az első feladatot az ún. teljes családok axiomatizálásá – a cikk első részében ezt végezzük. Kétféle axiómasémát adunk; az első csak a funkcionális, duális és erős függések teljes családjainak axiomatizálására alkalmas, mig a második séma a gyenge függésekére is.

Vizsgálunk még két, funkcionális függőségek teljes családjainak generálására, illetve kulcsrendszerekre vonatkozó probémát.

Végül megemlitjük azokat a lineáris relációkra vonatkozó tételeket, melyekből kiderül, hogy milyen következményei vannak a linearitásnak a reláció funkcionális függéseire.

Р Е З Ю М Е

Логическое и структуральное исследование в реальной базе данных

Я. Деметрович

В настоящей работе мы изучаем обобщения функциональных зависимостей. Кроме этого занимаемся линейными функциональными зависимостями и изучаем некоторые комбинаторные вопросы, связанные с реляционными базами данных.

MTA Számítástechnikai és Automatizálási Kutató Intézete, Közlemények 27/1982

MODELLING THE SOCIO-ECONOMIC DEVELOPMENT OF A NEW AGRICULTURAL REGION

Perfecto Dipotet
Institute of Mathematic, Cybernetic
and Computing Sciences
Habana, Cuba

In this paper are presented:
The main problems appearing in the organization, planning and management of Research Programs in the particular conditions of a developing country, Cuba.
Some mathematical models applied to the socio-economical-development of a new agricultural region.
Algorithms and programs to collect and to process information coming from experts. These algorithms are applied - to the microlocalization of socio-economical objects.

INTRODUCTION

The developing countries face great problems in the efficient exploitation of their resources. In order to solve the top-priority problems related- to social and economic development, i is necesary to concentrate to the- maximum the efforts of all the organizations, mainly those of research -- institutions. The solving of each one of these problems requires the - implementation of complex, long-range research programs, with well-defi-- ned aims and the participation of several research and production organizations. Programs concerning the development of new economic regions and mainly those related to agroindustrial regions are extremely necessary, - but only seldom carried out in developing countries.

The main objective of the research program is to develop the policy for - the long range socio-economical development of a region, and mainly to:

- establish the correct rythm and proportions for the development of the region;
- reach the stability in the development and management of natural, hu-- man and material resources;
- develop the social infraestructure allowing to stabilize the quantity-structure, settlement and reproduction of labor forces,
- to maximize the net integral effect (profit) of the economical activities in the region;

A retrospective analysis of the economic development of the region was performed, in order to have a clear picture of its current situation and trends

The main results of the analysis were:

- the need of increasing the efficiency of the organization and planning of the program, due to its complexity and to the lack of experience in our country;
- the need of using formal methods to process the qualitative information reported by the experts, due to the lack of reliable statistical data;
- the need of developing some mathematical models for planning long range agricultural production;
- to characterize the social factors affecting the economic development of the region.

Some of the experiences achieved in the organization and planning of a Research Program are offered, as well as, some implemented mathematical models, and the approach used to microlocalize the socio-economic objects in a new agroindustrial region.

MAIN FACTORS OF THE PROGRAM

When we analize the Reseach Program (DIPOTET-79), we observe there are, in -- our country, external factors, that we must take as compulsory, and internal-factors, particular for the Region, determining, to some extent, its development.

The external factor "policy for long range socio-economical development of -- the country" determine:

- global requirements in products, raw materials and services from the Re--- gion;
- the external resources to be allocated in the Region;
- indicators for social and institutional infraestructure to be developed in the Region.

In short, the economic development of the Region mainly depends on the effi- - cient management and exploitation of the external resources in the sense of - satisfying the requirements. Thus, it is possible to distinguish the follo-- wing strongly related general aspects:

- To develop and stabilize the population of the Region. Therefore, it is - needed to derive demographic models, to characterize the social factors -- affecting the economic development and to control the migration.
- To characterize the natural resources of the Region. Therefore, it is ----- needed to do a very complex work, the inventory and evaluation of these -- resources.
- To know and to evaluate the material resources, the available infraestructure and the tradition and experiences of the inhabitants of the Region.-- Therefore, it is needed to perform a retrospective analysis of the socio-- economic development of the Region, in order to derivea diagnosis of its - present situation.
- Conservation and, if possible, amelioration of the environment, mainly taking care of the consequences of human economical activities on natural re sources, Therefore, it is needed to estimate local and global limits in - the exploitation of these resources.
- The organization and management of the former four factors in order to --- maximize the net effect (global profit) of the socio-economical activities of the Region.

Each one of the former points may be considered, due to its compexity, as a-research subprogram where several organizations must participate in.

Being a developing region, it was needed to apply a dynamic approach in the works of the Program (Fig. 1). This approach allowed us to improve, frequently, the available information about the object, and to use the new information in decision making.

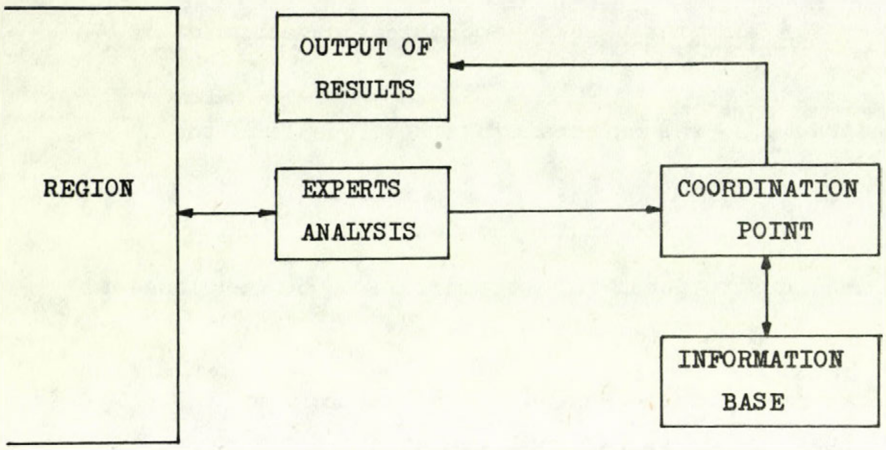


Fig. 1 Preliminary system to study the Region

With the improvement of the information base and, therefore, the new knowledge about the object and its environment it is possible to develop an information-system for planning long range economic development of the Region (DIPOTET-79) In other words, it is needed to perform the planning of long range investments projects for the Region based on some kind of man-machine system helping the -

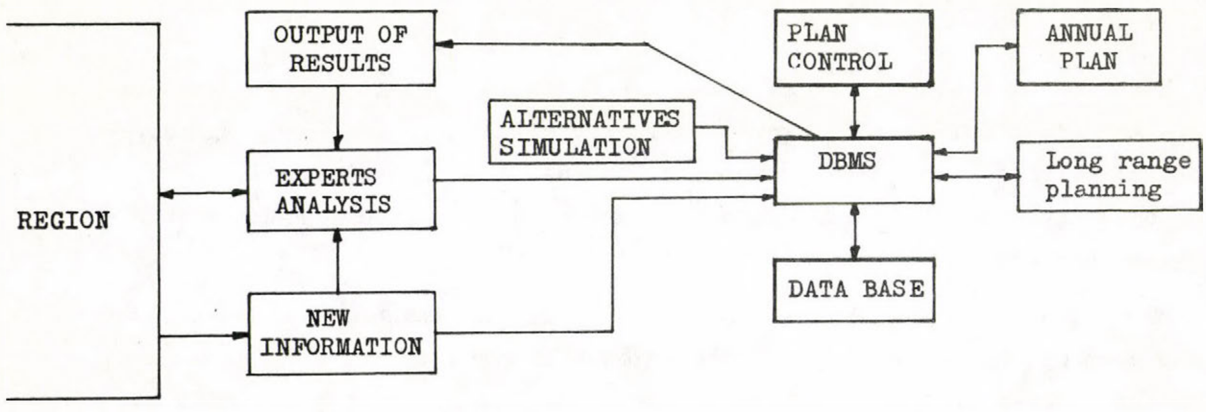


Fig. 2 System for planning long range economic development of the developing Region

"man", responsible for decision making, to evaluate the consequences of different plan alternations and to control, if necessary, the deviations of actual-plans.

Some mathematical tasks to be solved within the Research Program are:

1- To derive formal methods for the management of Research Programs in our - particular conditions.

2- To develop models for rational land exploitation.

3- To develop models for planning long range agricultural production.

4- To derive models for the microlocalization of socio-economical objects.

5- To develop models and MIS for the main enterprises of the Region.

Next we present some results about the Programs management, and the first version of the mathematical models implemented in the developing Region.

ORGANIZATION AND PLANNING OF THE PROGRAM

Production and service enterprises deal with concrete and well-defined tasks and subtasks. Scientific institutions deal with research themes. Thus, ---- themes must be formulated for each institution from the activities and jobs - belonging to the tasks of the program assigned to them.

Next we will present the procedures that must be carried out for collecting - and processing the data that will enable us to derive the reseach plan.

Let a set $J = (1, \ldots, n)$ of research institutions belonging to one organiza tion which must carry out a research program P in a given time T.

The Scientific Council of the Organization divides the Program P into several sets of important tasks $P_1, P_2, \ldots, P_m$.

Then, $P = (P_1, \ldots, P_m)$.

We use the form given in Fig. 1 to obtain the listing of the institutions of- the organization vs the tasks that they are going to undertake, respectively. For each $P_i \in P$; $i \in I = (1, \ldots, m)$ the Scientific Council establishes the deadline time $t_i \leq T$.

This deadline time t_i depends on several factors, but mainly on the will of - the user and the domestic requirement of the Organization.

The performance of each task is divided into r subtasks, for example in the- following 12 subtasks:

1) description of the tasks
2) formalization of the task
3) selection of methods and techniques to solve the task
4) collection and filtering of data
5) development of algorithms
6) programming
7) implementation of programs
8) analysis of results
9) improvement of the description of the task
10) modelling improvement
11) implementation of the solution of the task
12) drafting of reports, handbooks and user instructions.

The form shown in Fig. 2 offers the listing of all institutions vs the ---- subtasks where they will participate, respecively.

Tasks	INSTITUTES									ENTERPRISES						
	ACC	Bot.	Geog.	IMACC	Met.	Ocea.	Suel.	Zool.	P.P.	MINAG	CEATM	CONST.	Mat. Const.	Hid. Econ.	Educ.	Transp.
1		+	+	+	+					+						
2				+						+	+	+	+	+	+	+
.																
.																
10	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+

Fig. 1 Institutions vs Tasks

Subtasks / Tasks	1	2	3	4	5	6	7	8	9	10	11	12	Name of participants
1													
2													
.													
.													
10													

Fig. 2 Institutions vs subtasks

The Research Council builds up, using these forms (see Fig. 2) the matrix $A^j = \left\| a^j_{il} \right\|_{mxr}$ where $a^j_{il} \in (0,1)$; $i \in I = (1, \ldots, m)$

$j \in J = (1, \ldots, n)$

$l \in L = (1, \ldots, r)$

$a^j_{il} = 1$ means that institute j participates in the carrying out of task i in- the subtask l. The institution j lists the jobs and activities to be perfor- med within the time interval t_i, for each case $a^j_{il} = 1$. The research themes- are elaborated with the former list and the unification and generalization of other activities. The resources needed are established as well as the onset - and completion dates.

In those cases when $\sum a^j_{il} = 0$; $j \in J$; $i \in I$; $l \in L$; in other words, when none of the n institutions participate in the solution of one subtask a_{il}, it is - necessary to find other institutions that would open new themes concerning -- subtask a_{il}.

The form shown in Fig. 3 in used to list the research themes of the Program - vs the subtask where they will take part, respectively. These three forms -- are the additional blanks that must be filled out in the organization and --- planning (ACC-80) of the Program.

Program P is then formed by a set W = (1, ..., s) of research themes.

Let $a^w_{il} \in (0,1)$ denote each element (subtask a_{il} related to theme w) in Fig.3 $w \in W$; $i \in I$; $l \in L$ each theme w, $w \in W$ is then related to a set A^w of sub---- tasks a^w_{il}.

Then, $A^k \cap A^0 = A^{k0}$; k, $0 \in W$; is the set of subtasks where both themes k, 0- participate in simultaneously, cardinal N_{k0} of set A^{k0} is considered to indi- cate some relationship between themes k and 0; k, $0 \in W$.

The graph shown in Fig. 4 is the matrix $N = \left\| N_{k0} \right\|_{sxs}$ formed by cardinals- of the intersections sets (see Fig. 4) will, of course, be symmetric in res-- pect to the main diagonal.

In our case, we separate from the graph a subgraph, the maximun linked tree.- Each node of the tree will be a theme. The value of the links will be given- by their correspondent elements in matrix N, indicating some degree of rela- tionship among the themes.

The procedure to construct the tree is the following:

1) Selection of the maximum element N_{pp} in the main diagonal of matrix N. Node (theme) p is the root of the tree hierarchical structure -

Tasks / Themes	1	2	3	4	5	6	7	8	9	10
10180230 (1,1)	1,4, 8 9,12		1,4,8 9, 12	1,4,8 9,12	1,4,8 9,12	1,4,8 9,3		1,4,8 9,3	1,4,8 9,3	4,11
10280222 (1,2)	1,2,4, 8,9,12		1,2,4, 8,9,12		1,2,4, 8,9,12	1,2,4, 8,9,12		1,2,4, 8,9,12	1,2,4, 8,9,12	4,11
10380143 (1,3)	1,4,5, 8,9,12	1,4,5, 8,9,12	1,4,5, 8,9,12						1,4,5, 8,9,12	4,11
10480535 (1,4)	1,2,3, 4,9,12		1,2,3, 4,9,12					1,2,3, 4,9,12	1,2,3 4,9,12	4,11
10580541 (1,5)	1,4						1,2, 4,7, 8,9, 10,12		1,4	4,11
10780521 (1,6)	1,2,4, 8,9,12		1,2,4, 8,9,12		1,2,4, 8,9,12				1,2,4, 3,8,9, 12	4,11
10781521 (1,7)	1,2,4, 8,9,12		1,2,4, 8,9,12						1,2,3, 4,8,9	4,11
10980222 (1,8)	1,2,4, 8,9,12		1,2,4, 8,9,12		1,2,4, 8,9,12			1,2,4, 8,9,12	1,2,4, 8,9,12	4,11

Fig. 3 Themes vs subtasks

Themes Themes	1	2	3	4	5	. . . .	25	26	27	28	29	30
1	37											
2	32	38										
3	17	17	26									
4	18	18	14	26								
5	6	6	6	6	14							
.		.			.							
.		.			.							
.		.			.							
.		.			.							
.		.			.							
.		.			.							
.		.			.							
25	17	17	26	14	6	. . . .	26					
26	7	10	8	14	2		8	35				
27	30	32	22	22	11		22	35	100			
28	2	2	2	2	2	. . . .	2	5	6	11		
29	2	2	2	2	2		2	2	2	3	12	
30	2	2	2	2	2		2	2	2	5	3	6

Fig. 4

of the tree.

2) If tree has not free nodes then go to 3 else, search the nodes[1] corresponding to the next level testing the following conditions;

2.1) Inf $N_{pi} \geq N_{ki}$ for each k; k, i, p $\in$ W; p $\neq$ i $\neq$ k $\neq$ p; then node-i is linked to node p in the former level and is a leaf of the --tree.

2.2) If $N_{pk} \geq N_{pi} \leq N_{ki}$ for each k; k, i, p $\in$ W; i $\neq$ p $\neq$ k $\neq$ i;- then node p is linked to node i in the former level.

2.3) If $N_{pk} > N_{pi} < N_{ki}$ for, at least, one k; k, i, p $\in$ W; ----- i $\neq$ p $\neq$ k $\neq$ i; then node i is not linked to node p. If there is-another node in former level, then DO p that node and go to 2.1,-else ADD 1 to the level counter and GO TO 2.

3) EXIT.

In Fig. 5 the maximun linked tree is shown.

In our example, this subgraph aids the leaders of the program in decision-making concerning the management of the research.

For example:

a) it is obvious that theme 27 is really a "bottleneck", and it is --absolutely necessary to assure its resource allocation;
b) the subtrees derived from nodes 2 and 3 respectively may be considered as subprograms to improve program management;
c) themes 4, 12, 5, 17, 26 are practically isolated and it may be ---possible that their works may begin in advance or be delayed -----(within time interval t_i), according to resource allocation pro---blems.

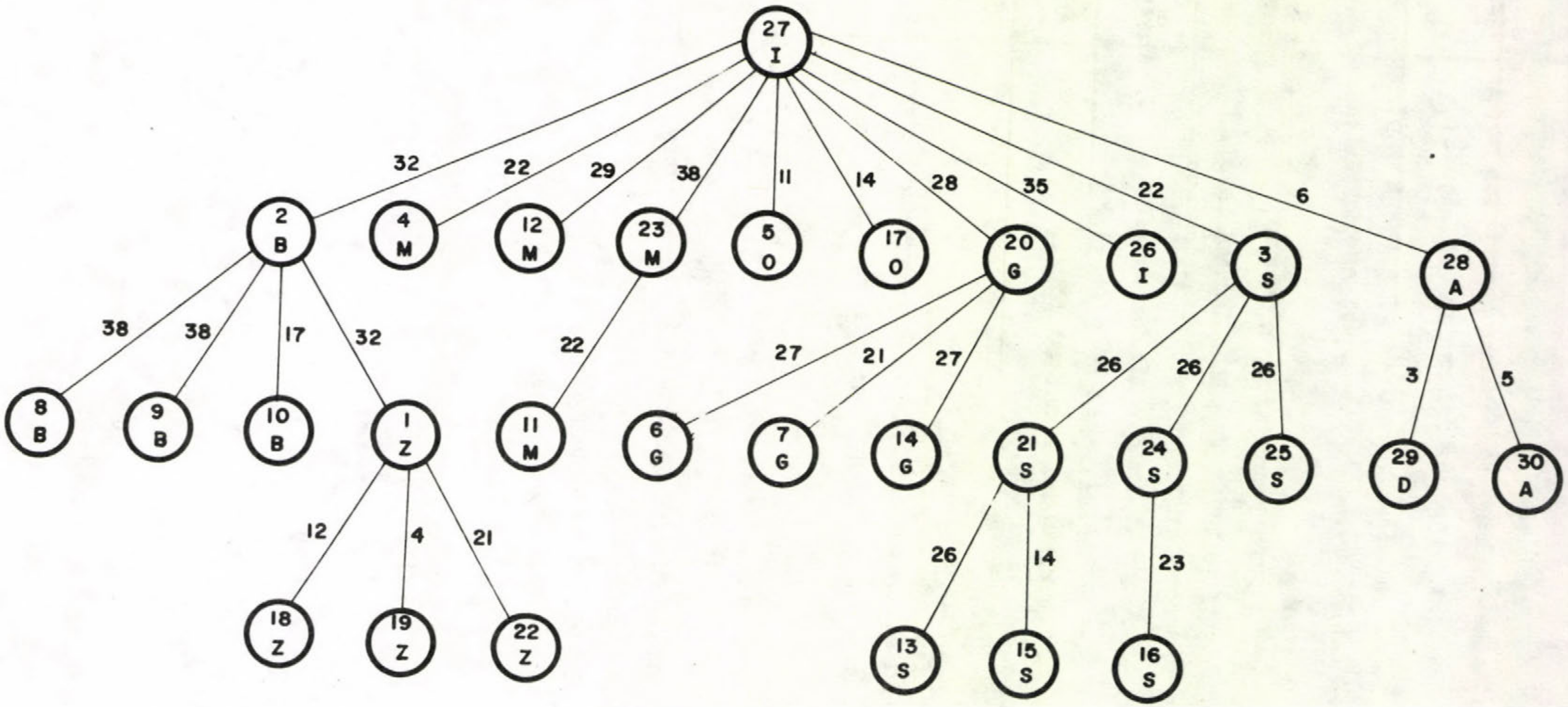
What we have presented above are only examples. There are many applications-of the tree and it is also possible (Dipotet, 1980) to derive other useful --subgraphs from the graph shown in Fig. 5.

RESOURCE ALLOCATION PROBLEMS

With the information received from the themes of the Program for each task -- P_i, $P_i \in P$; $i \in I$; we establish its working stages $(P_i(1), \ldots, P_i(t_i))$, ---- where $P_i(t)$, $i \in I$; $t \leq t_i$; is the working stage in time t.

For each $P_i(t)$ we determine its resource vector

1 See the detailed algorithm in (Dipotet-80)



MAXIMUM LINKED TREE

Fig. 5

$r_i(t) = (r_{i1}(t), \ldots, r_{ik}(t), \ldots, r_{iq}(t))$

where $k \in K = (1, \ldots, q)$ is the resource number.

Then $\sum_{i \in I} r_{ik}(t) = R_k(t)$, resource requirements **k** in time **t**.

The function $R_k(t)$ may vary (see Fig. 6). In some cases, the resource k, --- $k \in K$, is difficult to obtain, but generally the "rate of change" of the re-- source -maximum or minimum increase or decrease at the time unit- is known.

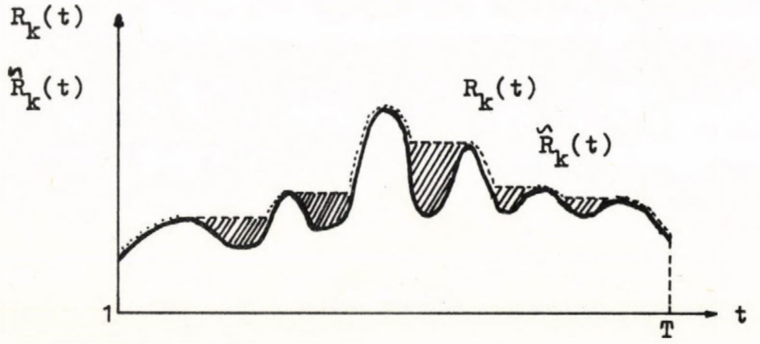


Fig. 6

In these cases, it is convenient to use function, (see Fig. 6) with only one maximum and $\max_t R_k(t) = \max_t \hat{R}_k(t)$.

However, when we have some surplus in a given resource, **k** for example, (see - Fig. 6), this surplus may be used in other research programs during the same- time interval **T**. When it is not possible to use them, it seems reasonable to reduce them so that $\max_t R_k(t)$ could be the minimun amount, and $R_k(t)$ will be almost parallel (Fig. 7) to axis **t**.

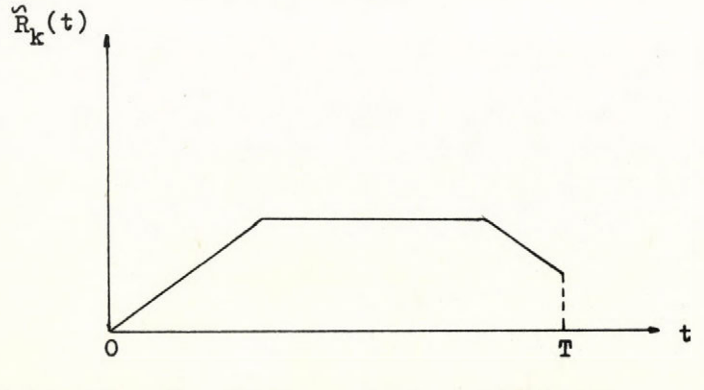


Fig. 7

We tried to solve the former problem as follows:

a) varying the content of stages $P_i(t)$ and, therefore, vector $r_i(t)$, -- but maintaining value t_i;

b) varying the set of tasks P_i, or participating in another set of ---- tasks;

c) varying $t_i \leq T$, but keeping the stage contents;

d) combining the former three approaches.

Next we describe one algorithm for building up the function $R_k(t)$. Suppose - we know the maximun rate of increasing (V_k) and decreasing (D_k) resource k. We find $R_k = \sum R_k(t)$ and $h_k = \frac{R_k}{T}$; if $h_k \leq V_k$ and D_k, then $\tilde{R}_k(T) = h_k$; if $D_k > h_k > V_k$, then $\tilde{R}_k(1) = V_k$ and $h_k^1 = \frac{R_k - V_k}{T - 1}$.

We then compare h_k^1 with 2 V_k and D_k.

If $h_k^1 \geq 2V_k$ then $\tilde{R}_k(2) = 2V_k$;

If $h_k^1 > D_k$ then $\tilde{R}_k(T) = D_k$.

Then, form $R_k - V_k$ we substract, respectively, $2V_k$ or D_k (or both) and divide by T - 2 or T - 3 to obtain h_k^2 .

The Procedure continues, until step r where $l_2 D_k \leq h_k^r \geq l_1 V_k$; $l_1, l_2 \in Z = (1, 2, \ldots, z)$. Thus in the former (r-1) steps the values of fuction $\tilde{R}_k(t)$ for $0 \leq t \leq l_1 - 1$ and $T - C_2 + 2 \leq t \leq T + 1$ ($\tilde{R}_k(0) = \tilde{R}_k(T+1) = 0$) were obtained.

For other t values, $l_1 \leq t \leq T - l_2 + 1$, we suppose $\tilde{R}_k(t) = h_k^r$.

Thus, $\tilde{R}_k(t)$ is derived for all t, $t \in (1, \ldots, T)$.

From the algorithm it follws that $\sum_{t=1}^{T} \tilde{R}_k(t) = \sum_{t=1}^{T} R_k(t)$.

However, it may be possible not to satisfy the resource constraints. It means $\sum_{i \in I} r_{ik}(t) \neq \tilde{R}_k(t)$.

It is possible to determine which resource distribution satisfies the cons--- traints for all tasks and all times <u>t</u> for $\tilde{R}_k(t)$ function.

We determine coefficient $a_k(t) = \frac{\tilde{R}_k(t)}{R_k(t)}$ and derive for each task i, $i \in I$, -- the resources vectors $r_i^1(t)$, $r_i^1(t) = (r_{i1}(t).a_1(t), \ldots, r_{iq}(t).a_q(t))$.

For it, $\sum_{i \in I} r_{ik}(t).a_k(t) = a_k(t) \sum_{i \in I} r_{ik}(t) = a_k(t).R_k(t) = \tilde{R}_k(t)$.

In this case, we assign $r_i^1(t)$ to each stage $P_i(t)$, $i \in I$; $t \in (1, \ldots, T)$; --- instead of $r_i(t)$, then the works must be updated within the stage.

It may happen that, for some tasks, resources are not enough for the whole -- time interval T and for other task they are in exceses. Then, experts must - analize and rectify the resource distribution plan. However it is possible - to apply mathematical programming methods for estimating the optimal, in a --

certain sense, resource distribution plan.

For example, the following quadratic programming problem is cosidered. To - determine X(t) values satisfying

$$\min \sum_{t=1}^{q} \sum_{k=1}^{q} \left[R_k(t).X(t) - \hat{R}_k(t) \right]^2$$, with the constraints

$$\sum_{t=1}^{T} r_{2k}(t).X(t) = \sum_{t=1}^{T} r_{ik}(t); \; X(t) \geq); \; i \in I.$$

In this work we have T variables and q x m type equality constraints.

Suppose we obtain the solution for X9TO = 1 for all t = (1, ..., T) using it as initial solution, it is possible to continue using, e.g., the gradient -- method.

This problem statement is not at all senseless. It means that the resource allocation of each stage of all tasks with the same proportionality coeffi-- cient must be changed, in order to obtain a resource distribution plan, for- time interval T, as near as possible to be best $R_k(t)$ (Fig. 7) and assurring for every task that the total requirement in T for every resource is obtai- ned.

PARCELS DISTRIBUTION IN REGIONAL PLANNING

For the solution of this task we used different linear programming models. - Let us present (DIPOTET-81) the simplest one.

Let:

J = (1, ..., n) be the set of towns (or users of the agricultural production);

K = (1, ..., k) be the set of differents productions;

I = (1. ..., m) be the set of parcels;

S_i = the area of parcel i, $i \in I$;

r_i^k = the production per unity of area of type k, $k \in K$, culture in parcel i;

d_i^k = costs associated to the production of k, $k \in K$, in parcel i, $i \in I$;

b_j^k = transportation cost for the unity of k, $k \in K$, form parcel i, $i \in I$, to consumer j, $j \in J$;

e_j^k = costs related to the consumption of the k product by the consumer j, $j \subset J$;

The task is how to distribute the parcels in order to satisfy the demands with - minimum costs.

Let us denote by x_{ij}^k the volumes of type k product transported form parcel i, -- $i \in I$, to consumer j, $j \in J$. Then, our task is to solve the following linear programming problem:

to determine:
$$\min_{x_{ij}} \quad {}_{i \in I} \quad {}_{j \in J} \quad {}_{k \in K} \; (c_{ij}^k + d_i^k + e_j) \, x_{ij}^k$$

with the constraints:

$$_{j \in J} \quad _{k \in K} \; \frac{x_{ij}^k}{r_i^k} = S_i$$

$$_{i \in I} \quad x_{ij}^k = b_j^k$$

$$x_{ij}^k = 0$$

Let us suppose the solution for this task is

$$x_{ij}^k = \underline{x}_{ij}^k$$

Then we obtain

$$\underline{x}_i^k = \sum_j \underline{x}_{ij}^k$$

The volumes of k products in parcel i, i I, and

$$\underline{s}_i^k = \frac{\underline{x}_{ij}^k}{r_i^k}$$

the areas of parcel i, $i \in I$, planted with k, k K.

This simple model may be complicated (DIPOTET-81) with other constraints related to resources, production technology, manpower and also the dynamic of - the Region. Then, it may be possible to use a block type linear programming problem (GOLSHTEIN-66).

In our case (the development of a new Region) it is needed to find mechanisms to intensify the production and to improve efficiency enterprises using this-resource (land).

For this reason we added to the former model the following elements:

= (1,2, ..., T) a set of time periods; is the planning period;

p_k^t = price (estimated) for product k, k K at time period t, t .;

f_i^t = cost associated to parcel i, $i \in I$, at time t, in order to be considered suitable for planting;

l_i^t = cost (tax) for 1 ha of parcel i, $i \in I$, at time period t ;

PLANNING PERENNIAL CROPS

For the solution of this task, it is possible to use different models ------- (DIPOTET-81). We think, in our particular, case, the most suitable approach- is the simulation optimization one (BEAUSOLEIL-80) but it presents some imple mentation problems. For this reason we have developed and implemented, with- good results, a linear programming model (ALEXEIEV-DIPOTET-1978), similar, in some sense, to Csaki's ideas (CSAKI-76). Next we present these ideas and afterwards our model.

Let

$J = (1, \ldots, j)$ be the set of hectareas in a given region;

$\mathcal{T} = (1, \ldots, T)$ be the set of time periods, T is the planning horizon;

$x_j(t) =$ the number of hectareas used for perennial crop j, $j \in J$, at period t, $t \in \mathcal{T}$;

$k_j^+(t) =$ the number of hectareas, used for new plantings of perennials of type j, $j \in J$, at time t, $t \in \mathcal{T}$;

$k_j^-(t) =$ the number of hectareas of perennial of type j removed at year t, --- $t \in \mathcal{T}$;

b_{jk} = proportion of lands of type k, $k \in J$,(i.e. with trees of type k) progres sing to type j, $j \in J$, in one year.

The state equations are then defined as

$$x_j(t+1) = \sum_{k=1}^{s} b_{jk}(t)\, x_k(t) + k_j^+(t) - k_j^-(t)$$

or in matrix form

$$x(t+1) = Bx(t) + k^+(t) - k^-(t)$$

where $x(t) = (x_1(t), \ldots, x_s(t))$ is the set state vector

and $k^+(t) = (k^+{}_1(t), \ldots, k_s^+(t))$; $k^-(t) = (k_1^-, \ldots, k_s^-(t))$ are the control vec tors.

We ilustrate the state equations por the perennial crop with an example of citrus fruit production. Consider de following production time periods:

	Age of trees Years
$x_1(t)$	0 - 1
$x_2(t)$	1 - 2
$x_3(t)$	2 - 3

$x_4(t)$ 3 - 4

$x_5(t)$ 4 . . . producing * or mature tress

The state equation for new plantings is

$$x_1(t+1) = k_1^+(t)$$

the trees in the second year

$$x_2(t+1) = b_{21}x_1(t)$$

$$\vdots \qquad \vdots$$

and trees in the fith and succeding years (producing or mature trees)

$$x_5(t+1) = b_{55}x_5(t) + b_{54}x_4(t)$$

With the given $b_{jk}(j = 1, ..., 5)$ $(k = 1, ..., 5)$

In the matrix form the state equations are written:

$$x(t+1) = Bx(t) + hk_1^+(t);$$

where

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ b_{21} & 0 & 0 & 0 & 0 \\ 0 & b_{32} & 0 & 0 & 0 \\ 0 & 0 & b_{43} & 0 & 0 \\ 0 & 0 & 0 & b_{54} & b_{55} \end{bmatrix} \qquad h = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

here we have 5 state variables $x(t) = (x_1(t), ..., x_5(t))$; one control variable $k_1^+(t)$ and $t = 1$ year.

The system of state variables can be simplified by succesive substitution. For example:

$$x_5(t+1) = b_{55}x_5(t) + bk_1^+(t-4)$$

where $b = b_{54}\, b_{43}\, b_{32}\, b_{21}$

Thus we have one state variable, one time delay and $t = 1$ year.

If we choose time period equals 5 years, then we can even eliminate time delay. The state equation then reduces to:

$$x_5(t+1) = \bar{b}_5 x_5(t) + \bar{\bar{k}}\,(t), \text{ where}$$

* Note: We consider different production of mature trees from 5 to 10 years (age of trees)

$\bar{k}(t)$ = the number of planting during 5 year period;

$\bar{b}_5$ = shows what proportion of trees, planted during a 5 years period, will be producing.

Next we present the implemented model

Let,

$\mathcal{T}$ = (78, ..., 78 + T) be the set of time periods, T is the planning horizon;

J = (79, 80, ..., 85) be the set of years when the parcels were conditioned to be planted;

I = (69, 70, ..., 85) be the set of years when trees planted;

K = (1, ..., 4) be the set of types of parcels, according to their degree of erosion;

Parameters:

C_{ij}^{kt} = additional investments at year t, $t \in \mathcal{T}$, related to reconstructions at year j, $j \in J$, for 1 ha. of parcel type k, $k \in K$, planted at year i, -- $i \in I$, and $t \geq j > i$

a_{ij}^{kt} = production at year t, $t \in \mathcal{T}$, for 1 ha. planted at year i, $i \in I$, for -- parcel type k, $k \in K$, reconstructed at j, $j \in J$;

c_{ij}^{kt} = processing cost at year t, $t \in \mathcal{T}$, for 1 ha. of parcel of type k, $k \in K$, planted at i, $i \in I$, and reconstructed at j, $j \in J$;

$c_{ij}^{k} = \sum_{t=79}^{95} c_{ij}^{rt}$ integrated cost related to processing 1 ha. of parcel of type k, $k \in K$, planted at i, $i \in I$, and reconstructed at j, - $j \in J$;

c_j^t = processing cost at year t, $t \in \mathcal{T}$, related to the utilization of 1 ha. of parcels planted at j, $j \in J$;

p^t = price for 1 Ton. purchased production at year t, $t \in \mathcal{T}$;

s^{ki} = area for k type parcels planted at i, $i \in I$;

s^j = area of new parcels ready to be planted at year j, $j \in J$;

Q^t = sinking funds at year t, $t \in \mathcal{T}$, for the factory processing the fruits;

q^T = sinkings funds (for unity of capacity) for the enlargement of the factory at year T, $T \in \mathcal{T}$;

M = actual processing capacity for the factory at the beginning of the --- planning period;

Decision variables:

x_{ij}^{k} = area for k type parcels, planted at year i, $i \in I$, and reconstructed - at j, $j \in J$;

x_j = area planted at year j, $j \in J$;

y^t = volume of fruits produced at year t, $t \in \mathcal{T}$;

z^T = enlargement of the capacity of the processing factory at T years;

v^t = profit obtained by fruit production and processing at year t, $t \in \mathcal{T}$;

N^t = investments at year t, $t \in \mathcal{T}$; for the enlargement of social infraestructure;

Constraints:

1.- $\sum_{j=78}^{85} x_{ij}^{k} = s^{ki}$, $i \in I$; $k \in K$;

condition constraining k type parcels planted at i, $i \in I$;

2.- $x_j = s^j$, $j \in J$;

condition constraining new parcels planted at j, $j \in J$;

3.- $p^t y^t - \sum_{j=79}^{t} \left(\sum_{i \in I} \sum_{k \in K} (c_{ij}^{kt} + c_j^t) x_j \right) - \sum_{t=79}^{t} q^{t'} z^{t'} - v^t = Q^t$

t', $t \in \mathcal{T}$;

balance conditions to form profits resulting from production, processing - and purchase of the fruits at year t, t ;

4.- $\sum_{j=79}^{t} \left(\sum_{i \in I} \sum_{k=1}^{4} a_{ij}^{kt} x_{ij}^{k} + a_j^t x_j \right) - y^t = 0$, t, $t \in \mathcal{T}$;

conditions to produce the volume of fruits at year t;

5.- $y^t - \sum_{t'=79}^{t} z^{t'} \leq M$; t', $t \in \mathcal{T}$;

conditions for the factory to process the volume of fruits produced;

It is demanded to maximize the general profit for the planning period. It --- means:

$$\sum_{t=79}^{95} p^t y^t - \sum_{j=79}^{85} \left(\sum_k \sum_i c_{ji}^k x_{ji}^k + c_j x_j \right) - \sum_{t'=79}^{85} q^{t'} z^{t'} - Q \longrightarrow \max$$

The model was succesfully implemented and is actually used, mainly, for crop-prediction and resource allocation problems within the citrus enterprise "Isla de la Juventud".

PLANNING LON RANGE LIVESTOCK

Next we present the general ideas (CSAKI-76) of a DLP model for a livestock production system, and the implemented version for our particular case (MORIN-DIPOTET-81).

Let

$I = (1, \dots, s)$ be the set of differents type of animals;

$\mathcal{T} = (1, \dots, T)$ be the set of time periods, T is the planning horizon;

$x_i(t)$ = the number of animals of type i, $i \in I$, at year (period) t, $t \in \mathcal{T}$;

$k_1^+(t)$ = the number of animals of type i, $i \in I$, purchased at period t, $t \in \mathcal{T}$;

$k_1^-(t)$ = the number of animals of type i, $i \in I$, sold at period t, $t \in \mathcal{T}$;

a_{ij} = the coefficient which shows what proportion of animals of type j, $j \in J$, will progress to type i, $i \in I$, in the succeding period.

Then we can write the state equations for the livestock subsystem as:

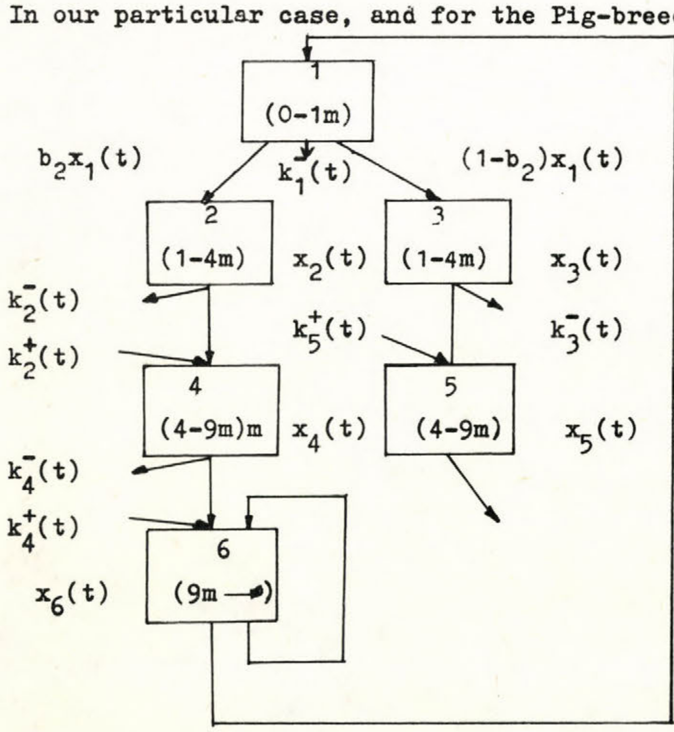
$$x_i(t+1) = \sum_{j=1}^{s} a_{ij}x_j(t) + k_i^+(t) - k_i^-(t) \qquad (1)$$

or in matrix form

$$x(t+1) = Ax(t) + k^+(t) - k^-(t)$$

Here $x(t) = (x_1(t), \dots, x_s(t))$ is the vector of state variables;
$k^+(t) = (k_1^+(t), \dots, k_s^+(t))$ and $k^-(t) = k_1^-(t), \dots, k_s^-(t))$ are vectors of control variables.

In our particular case, and for the Pig-breeding subsystem example we have:



t = 1 month (time unit)

$x_1(t+1) = a_{61}x_6(t)$

$x_2(t+1) = a_{12}x_1(t) - k_1^-(t)\, b_2 + (1-a_{24})x_2(t)$

$\vdots \qquad \vdots$

$x_6(t+1) = a_{46}x_4(t) + a_{66}x_6(t) + k_6^+(t+1) - k_4^-(t)$

<u>State variables:</u>

$x(t) = (x_1^t, \dots, x_6^t)$

Control variables

$$k^-(t) = (K_1^-(t), \ldots, k_4^-(t))$$
$$k^+(t) = (k_2^+(t)_1, k_3^+(t), k_4^+(t))$$

Values for a,b and constraints equations, depend on local conditions and production technology (MORIN-81).

In our case, we obtained the optimal flock structure at time to (for x_1 known) using natural, capacity, food, manpower and technological constraints and LP - program package.

With this structure as initial conditions we derived the time (year) recurrency $x(t) \longrightarrow x(t+1)$, $x(t+1) = Ax(t) + k^+(t) - k^-(t)$ (for fixed local conditions - and technology, it means constraints) trying to maximize the flock ---------- $(z = \sum_{\substack{i=1 \\ t \in \tau}}^{s} x_i(t))$ keeping its internal optimal structure.

For the time being, the accuracy of the results we obtained with this model is enough, nevertheless we are working on different type of models, for this same livestock system.

PROBLEMS ON SOCIO-ECONOMICAL OBJECTS MICROLOCALIZATION

The elements of this problem are:

- a set J = (1, ..., n) of raw materials sources;
- a set I = (1, ..., m) of points where it is possible to place the enterprises for processing the raw materials;
- a known function $b: J \longrightarrow R$, whose values b_j represent the volumes of raw materials coming from the sources j, $j \in J$;
- an unknown (a priori) function $X: I \longrightarrow R$ whose value X_i is the capacity of -- point i, $i \in I$, for processing raw material;
- a function $x: I \times J \longrightarrow R$ such that x_{ij} is the quantity of raw material from source j, $j \in J$, to be elaborated in point i, $i \in I$;
- a known family $(g_i : R \longrightarrow R)_{i \quad I}$ of functions such that

$$g_i(X_j) = \begin{cases} 0 \text{ if } i \neq j \\ \text{building and maintenance cost for the enterprise } i, i \in I, \text{ de-} \\ \text{pending on capacity } X_i. \end{cases}$$

- a known function $T: I \longrightarrow R$ whose value T_i is the building cost for object- i, $i \in I$;
- a known function $K: I \longrightarrow R$ for the processing cost of raw material unity in the object i, $i \in I$;
- a function $c: I \times J \longrightarrow R$ transportation cost of raw material from j, $j \in J$,- to i; if $d: I \times J \longrightarrow R$ is the distance matrix for points i and j, then, -- generally, the transportation point is proportional $(p: R \longrightarrow R)$ to the -- distance and $c_{ij} = p \cdot d_{ij}$;

- a function $P : 2^I \longrightarrow R$ defined on the set of parts $w \subset I$

$$P(w) = \sum_{\substack{i \in I \\ j \in J}} c_{ij} x_{ij} + \sum_{i \in w} g_i(X_i)$$

P(w) is interpreted as the cost of construction, maintenance and transport for objects at ponnts i, $i \in w$.
- the actual shape of g_i is $g_i(X_i) = (K_i X_i + T_i)$. Sign (X_i) it is possible to show (Figueroa-Jachaturov 78) that P(w) may be calculated in the following way:

$$P(w) = \sum_{j \in J} b_j \min_{i \in w} (C_{ij} + K_i) + \sum_{i \in w} T_i$$

The problem now is to calculate an $\alpha \subset I$ such that $P(\alpha) = \min_{w \in I} P(w)$ with the following natural constraints $b_j = \sum_{i\in\alpha} x_{ij}$ (all raw materials are distributed to the enterprises)

$$x_{ij} \geq 0$$

$$X_i = \sum_{j\in J} x_{ij}$$ (the capacity of point i is -- completely used)

There are algoritms (Jachaturov-78) for the exact solution of the former problem.

In our particular case, we must offer recomendations for building some socio-economical objects under the following conditions (Jachaturov-Figueroa 78):

$$P(w) = \sum_{j\in J} b_j \min_{i\in w} (l_{ij} \cdot p + K_i) + q \sum_{i\in w} T_i ;$$

Where p and q are some kind of fuzzy parameters with the following meaning:

p is the transportation cost for the production unity to the unity of distance;

q is the normative coefficient for efficiency in capital investments.

$\left| l_{ij} \right|$ is a known matrix giving the distances between points i, $i\in I$, and j,-- $j\in J$, respectively.

The available information about p and q may be graphically represented as -- fuzzy sets originated by statistical sources (or others)

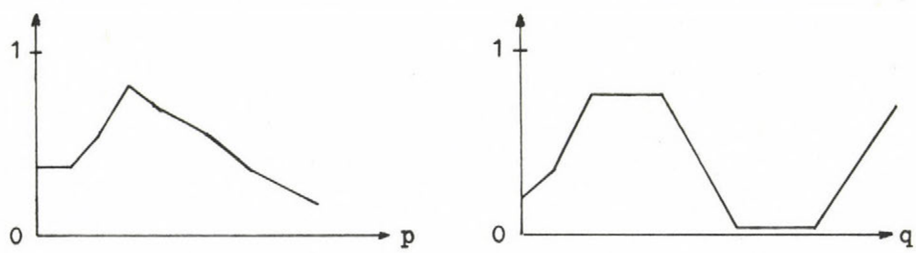


Fig.8

We also consider the case when there is no information about p or q. Then - the task of "more rational, instead of optimal, microlocalization of the objects", nust be solved.

We suppose, simplifying, $(\forall i \in I)$, $K_i = K$, $T_i = T$.

Then, $P(w) = p \sum_{j\in J} \min_{i\in w} l_{ij} + K \sum_{j\in J} b_j + q \left| w \right| T.$

We next investigate the task solution for one parameter.

We investigate for q fixed, for example, q = 1.

Then $P(w) = (\sum_{j \in J} \min_{i \in w} l_{ij})\, p + |w|\, T + K \sum_{j \in J} b_j$

- We first determine an interval for parameter p, $p \in [A,B]$, where:
  if $p < A$ the solution has cardinal minimum not null;
  if $p > B$ the solution has cardinal maximum (it is needed to build in all -- points).
- We solve several systems of linear equations to find subintervals (in p) -- where the solutions do not change within it. We obtain, for example

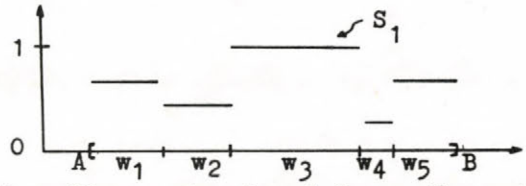


Fig.9

together with a "fuzzy structure" S_1 on the set of solutions $\{w_1, \ldots, w_5\}$ such that the maximum degree of pertenence belongs to the more stable --- (for parameter variation) set;
- A refinement of the former partition is calculated. The subinterval ----- (l_i, l_{i+1}) corresponding to the solution w_i is divided into subintervals - λ_j, λ_{j+1}, $(\lambda_i = l_i, \lambda_r = l_{i+1})$, $j = 1, \ldots, r$; where the raw material - supply from the sources to the processing plants do not change. We obtain also a fuzzy structure on the subintervals.

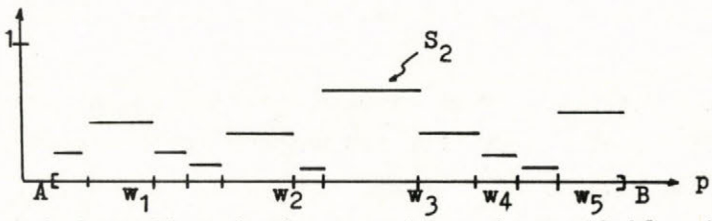


Fig. 10

- If some a priori information about parameter p is available, it may be a - fuzzy one.

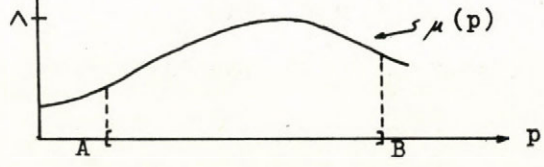


Fig. 11

We take its restriction (see Jachaturov-Figueroa 79) to $[A,B]$, $f \wedge \mathcal{U}_{[A,B]}$. We-may propose, for management decision making, the algebraic product of all fuzzy structures considered extended to 2^I

$$R(w) = \begin{cases} 0 \text{ if } w = w_i \text{ for } i = 1, \ldots, 5 \\ (S_1 . S_2 . \bar{\mu})(w_i) = S_1(w_i) . S_2(w_i) . \bar{\mu}(w_i) \end{cases}$$

where $\bar{\mu}(w_i) = \max_{p \in [l_i,\, l_{i+1}]} \bar{\mu}(p)$

It is possible to show that there is an interval $\left[\underline{p}, \bar{p}\right]$ such that, if $p \leq \underline{p}$- then only one object must be built and, if $p \geq \bar{p}$ then it is needed to build-- objects in all available places. We are then interested in the interval ---- $\left[\underline{p}, \bar{p}\right]$.

We cannot (due to the high dimensionality of the job) calculate a straight-- line for each $w \subset I$ such as

$$P(w) = \left(\sum_{j} \min_{i \in w} l_{ij}\right)p + |w| \; T + K \sum_{j} b_j$$

and to consider the envelope (convex, piecewise linear curve) of that family- of straight lines.

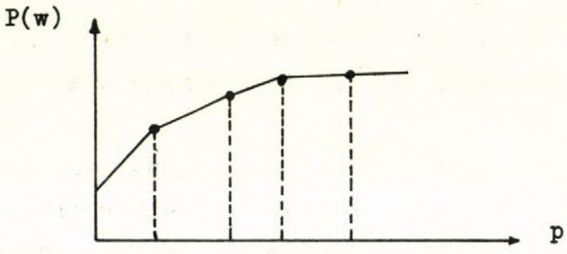


Fig. 12

With this approach we could obtain a partition for interval $\left[\underline{p}, \bar{p}\right]$ in subin-- tervals where the optimal solution w could be constant.

To avoid the former difficulties we applied a method which consisted in the-- a priori determination of a very thin and uniform partition of the subinter-- val $\left[\underline{p}, \bar{p}\right]$. For each value of p corresponding to the nodes of the partition we calculate (Jachaturov-76) the global minimum for function P(w). In this - case we developed (Jachaturov-Figueroa 78) an algorithm for searching local - minimum and, afterwards we select the "best" one.

In that way we obtain a partition of $\left[\underline{p}, \bar{p}\right]$ in stability subintervals of the minimum of p.

Supported by heuristical criteria we assign, in that way, to all $w \subset I$ one --- priority. For example, we can assign one priority proportional to the measu- re of the stability respecting the variation of p. The $w \subset I$ outside the par- tition receives priority o. Thus, one fuzzy set structure on $\Omega = 2^I$ is deri-- ved which solves our task.

We obtained the solution for q = 1. This allows us to determine a partition- of the first quadrant in the following way,

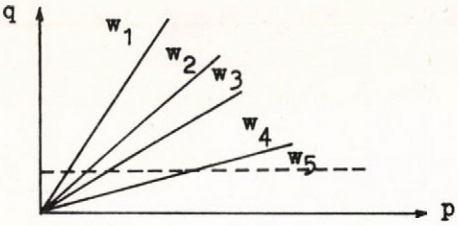


Fig. 13

Where each area of the plane is a stability area for the solution. Thus, - if a pair is known, we can derive the subset which allows us to obtain the minimum. When q diminishes and the socio-economical objects are cheaper (the building); the solution may change, increasing the number of objects (for p-fixed). If q is fixed and transport cost is reduced, we decrease the number of objects.

When we have the solution as a priority structure on the power set of I ---- (fuzzy subset of I), for each subinterval $\left[l_k, l_{k+1}\right] \subset \left[\underline{p}, \bar{p}\right]$ corresponding to a not null priority subset, we then obtain a partition of $\left[l_k, l_{k+1}\right]$ --- into stability subintervals for the assignation of elements of J to those of w_k.

If $\underline{C}_{ij} = l_{ij}.l_k$ and $\bar{C}_{ij} = l_{ij}.l_{k+1}$; then,

$$P(w_k) = \sum_{j \in J} b_j \min_{i \in w_k} \underline{C}_{ij} + \quad (\bar{C}_{ij} - \underline{C}_{ij}) \quad + \sum_{i \in w_k} T_i ;$$

Where, for each $j \in J$, the function $\min\limits_{i \in w_k} \left[\underline{C}_{ij} + \lambda\,(\bar{C}_{ij} - \underline{C}_{ij})\right]$ is the envelopment of the straigh lines of the following type

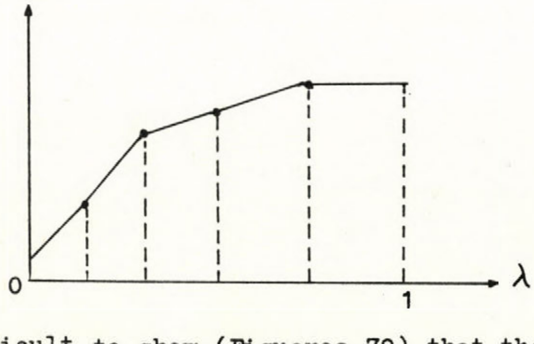


Fig. 14

It is not difficult to show (Figueroa-79) that the sum of all the former --- funcions, $P(w_k)$, is a function of the same type.

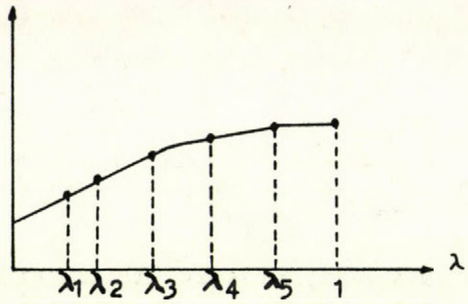


Fig.15

It is also possible to show that on each subinterval $\left[\lambda_k, \lambda_{k+1}\right]$ the assignations are constants.

Now we can determine (for each w⊂ I with its priority a priority structure- on the set of their possible assignations, as we have done before with the - subsets w⊂ I.

For simplifying the results, we list only the support of the fuzzy subset -- determined by the priority structure. In the listing, we present several -- (one for each w⊂ I with positive priority tables composed of two columns -- one for asignation and the other for priority and as many rows as there are- assignations with no null priority.

The former algorithms have been applied (Figueroa-79) to the microlocaliza-- tion fo some socio-economical objects in the Island of Pines.

DISCRETE MODELS IN REGIONAL PLANNING

The elements of the general problem are:

- a set J = (1, ..., n) of raw materials sources;
- a set I = (1, ..., m)of points where it is possible to place the enterprises for processing the raw materials;
- a set $\mathcal{T}$= (1,2, ..., T) of time periods, where T is the planning period;
- a known function b: J ⟶ R, whose values b_j^t represent the volumes of raw materials coming from the sources j, $j \in J$, at time $t \in \mathcal{T}$;
- an unknown (a priori) function X: I ⟶ R whose value X_i^t is the capacity - of point i, $i \in I$, for processing raw material al time $t \in \mathcal{T}$;
- a function x: I x J ⟶ such that x_{ij}^t is the quantity of raw material --- from source j, $j \in J$, to be elaborated in point i, $i \in I$, at time $t \in \mathcal{T}$;
- a function c: I x J ⟶R transportation cost of raw material from j, --- $j \in J$, to i, $i \in I$, at time $t \in \mathcal{T}$; if d: I x J ⟶ R is the distance matrix for points i and j, then, generally, the transportation point is proportional (p: R⟶R) to the distance and $c_{ij} = p.d_{ij}$;
- a function x: I x J ⟶R such that x_{ij}^t is the quantity of raw material - from source j, $j \in J$, to be elaborated in point i, $i \in I$, at time $t \in \mathcal{T}$;
- a known function T: I ⟶R whose value T_i is the building cost for object i, $i \in I$;
- a known function K: I ⟶R whose value K_i^t is the processing cost of raw-material unity in the object i, $i \in I$, at time $t \in \mathcal{T}$;
- $g_i(X_i^t)$ are the building and processing cost for the enterprise i, $i \in I$,-- at time t, $t \in \mathcal{T}$, depending on capacity X_i;

Let $X_i^t = \sum_{j \in J} x_{ij}^t$ then the task of optimal object micolocalization for a given region may be formulated as:

to determine, $\min\limits_{x_{ij}^t} \sum_{i \in I} \sum_{j \in J} \sum_{t \in \mathcal{T}} c_{ij}^t \, x_{ij}^t + \sum_{i \in I} \sum_{t \in \mathcal{T}} g_i(X_i^t)$;

the constraints are:

$$\sum_{i \in I} x_{ij}^t = b_j^t$$

$$X_i^t = \sum_{j \in J} x_{ij}^t \leq a_i^t$$

$$x_{ij}^t \geq 0$$

If functions $g_i(X_i^t)$ are linear, then we face a transport dynamic task and to solve it we apply the linear programming methods. Nevertheless, generally, these functions $g_i(X_i^t)$ are not linear, but discrete discontinuos ones, difficulting the solution for this task in the general case.

If $g_i(X_i^t) = (K_i X_i^t + C_i)$ Sign X_i^t (DIPOTET-81) then our task will be reduced to:

$$\min_{x_{ij}^t} \sum_{i \in I} \sum_{j \in J} \sum_{t \in \tau} (c_{ij}^t + K_i^t)\, x_{ij}^t + \sum_{i \in I} T_i \text{ Sign } X_i^t$$

$$\sum_{i \in I} x_{ij}^t = b_j^t$$

$$\sum_{j \in J} x_{ij}^t \geq a_i^t;\ x_{ij}^t \geq 0$$

This task is a multi-extremes one in non-linear programming. It may be solved using combinatorical methods, using the "branch and bound" method. However, for using these methods in medium size problems, powerful computers are needed.

For long range planning, in our case, it is possible to use a simplified model not considering capacity (X_i^t) constraints.

Then, our task will be the following:

to determine,

$$\min_{x_{ij}^t} \sum_{i \in I} \sum_{j \in J} \sum_{t \in \tau} d_{ij}^t\, x_{ij}^t + \sum_{i \in I} T_i \text{ Sign } X_i^t$$

$$\sum_{i \in I} x_{ij}^t = b_j^t,\quad x_{ij}^t \geq 0$$

were $d_{ij}^t = c_{ij}^t + K_i^t$

Next we present the combinatorical version for the former problem statement.

Let $w \subset I$ be a subset where we suppose must be buil the objects to microlocalize.

Then, on the set of all subsets $w \subset I$ it is possible to define function $P(w)$ in the following way:

$$P(w) = \min_{x_{ij}^t} \sum_{i \in w} \sum_{j \in J} \sum_{t \in \tau} d_{ij}^t\, x_{ij}^t + \sum_{i \in w} T_i;$$

$$\sum_{i \in w} x_{ij}^t = b_j^t$$

$$x_{ij}^t \geq 0,\ i \in w$$

There are not constraints linking the variables by t, then we may write:

$$P(w) = \sum_{t\in \mathcal{T}} \min_{x_{ij}^t} \sum_{i\in w} \sum_{j\in J} d_{ij}\, x_{ij}^t + \sum_{i\in w} T_i$$

There are not constraints of the type $X_i^t \leq a_i^t$ then it is possible to write:

$$P(w) = \sum_{t\in \mathcal{T}} \sum_{j\in J} b_j^t \min d_{ij}^t + \sum_{i\in w} T_i$$

$$P(w) = \sum_{t\in \mathcal{T}} \sum_{j\in J} \min_{i\in w} b_j^t\, d_{ij}^t + \sum_{i\in w} T_i$$

To solve this task it is needed: to buit T matrix $\left\| b_j^t\, d_{ij}^t \right\|_{mxn}$; then, to --- find $\sum_j \min b_j^t\, d_{ij}^t$ for every t, $t \in \mathcal{T}$; then to sum up (index t) and to add - $\sum_{i\in w} T_i$.

In the most simple case, when $d_{ij}^t = d_{ij}$ (it means $c_{ij}^t = c_{ij}$; $K_i^t = K_i$) it is - possible to write:

$$P(w) = \min_{x_{ij}^t} \sum_{i\in w} \sum_{j\in J} d_{ij} \sum_{t\in \mathcal{T}} x_{ij}^t + \sum_{i\in w} T_i$$

We have not constraints $X_i^t \leq a_i^t$, then x_{ij}^t for every t must take value 0 or - (exclusive) b_j^t, because of constraints.

$$\sum_{i\in w} x_{ij}^t = b_j^t$$

Then, to determine the P(w) value it is possible to use the following

$$P(w) = \sum_{j\in J} \min d_{ij} \left(\sum_{t\in \mathcal{T}} b_j^t\right) + \sum_{i\in w} T_i$$

Now, the computations to find P(w) are very simplified because the task is - now a not dynamic one and it is possible to calculate

$$P(w) = \sum_{j\in J} \min_{i\in w} d_{ij}\, \bar{b}_j + \sum_{i\in w} T_i; \qquad \text{(I)}$$

here $\bar{b}_j = \sum_{t\in \mathcal{T}} b_j^t$ is only once calculated before computing the P(w) values - for every $w \subset I$.

We remark that in this simple case P(w) is related to distance and building-cost and (I) fulfill (for the moment) the requirements of our problem (ob--- jects microlocalization a developing region).

REFERENCES

ALEXEIEV A.M., DIPOTET P. - 79: Modelo Economico-Matematico para la Obtencion de Variantes de Desarrollo Citricola.
Habana. ACC.

RIVERO O. - 80: Modelo para el Balance Alimentario del Ganado Vacuno.
Informe Tecnico. IMACC. Academia de Ciencias de Cuba.

BEAUSOLEK R. - 80: Modelode Simulacion para la Produccion a Largo Plazo de Cultivos Perennes.
Revista CCA. III. Habana.

CSÁKI C. - 79: A Long Range Project for Agricultural Production
IIASA Report. Laxenburg Austria.

DIPOTET P., JACHATUROV V.R. - 79: Niekotorie Vaprosi Razrabotki Informatsionnoi Sistemi pri Realizatsii Kompleksnih Program Isliedovanii
VIEN KHOA HOC VIETNAM. HANOI.

DIPOTET P. - 80: Elaboracion de un Sistema de Informacion para la Planificacion de las Actividades Cientifices de la ACC.
Informe Tecnico. IMACC. ACC.

DIPOTET P., JACHATUROV V.R. - 81: PLANNING A RESEARCH PROGRAM For the Development of an Agricultural Region.
IFIP WORKING CONFERENCE ON COMPUTERS APPLICATIONS in Agriculture and Food Production
Habana, October 1981.

DIPOTET P. - 81: Elaboracion de un Sistema Hombre-Máquina para la Proyeccion del Desarrollo de Nuevos Territorios.
IMACC. Academia de Ciencias. T-31904.
Estudio de Orientacion.

FIGUEROA M. - 78: Progremes para la Microlocalizacion de Objetos Socio-Economicos.
IMACC. Academia de Ciencias. Habana.

JACHATUROV V., FIGUEROA M.: Microlocalizacion de Objetos Socio-Economicos en Condiciones de Informacion Difusa.
Informa Technico. T-31904. IMACC. Habana.

ACC - 1980: Procedimientos para la Planificacion de Temas de Investigacion.
PACI. Academia de Ciencias de Cuba. Habana.

MORIN E., DIPOTET P. - 82: Modelo para la Planificacion de la Produccion a Largo Plezo de una Empresa Porcina.
Submitted to Journal CCA. Habana. Cuba.

VIESIELIOVSKI V. ET. AL. - 82: Algorithms and Programs for Object Microlocalization.
Technical Report. IMACC. ACC. HABANA. CUBA

Моделирование социально-экономического развития некоторого нового сельскохозяйственного района

Перфекто Дипотет

В работе дается общая концепция информационной системы поддерживающей комплексное развитие в отсталых областях развивающихся стран. Также описываются математическим программированием решаемые модели для некоторых конкретных подсистем /проектирование выращивания многолетных, размещение социально-экономических объектов, и т.д./.

Egy új mezőgazdasági terület társadalmi-gazdasági fejlődésének modellezése

A dolgozat egy, a fejlődő országok elmaradottabb területi egységeinek komplex gazdasági fejlesztését támogató információs rendszer koncepcióját vázolja fel. A szerző az általános leirás mellett több konkrét részrendszerre /évelőtermesztés tervezése, társadalmi-gazdasági objektumok elhelyezése, stb/ matematikai programozással megoldható modellt is közöl.

ITA Számítástechnikai és Automatizálási Kutató Intézete, Közlemények 27/1982

FORMALIZATION OF CONCURRENCY CONTROL IN DISTRIBUTED DATA SYSTEMS (1)

Serge M. MIRANDA

CERISS-INRIA
Universitě des Sciences Sociales
Place Anatole France
31040 Toulouse Cedex
France
Tel (61) 23 11 45 ext 395

ABSTRACT

Synchronization issues in distributed data bases were heavily invested n recent literature and many synchronization protocols have been designed.

However much work is still to be done in the areas of ROBUSTNESS and ECOVERY, FORMAL PERFORMANCE ANALYSIS, FORMAL SPECIFICATION AND VALIDATION, IGOROUS UNIFORMIZATION.

This paper is a contribution to the two latter points; we propose a ormal approach based on abstract data types (algebraic methodology) and evelop a uniform rigorous framework in which the synchronization protocols an be specified and validated.

We illustrate our approach on a basic protocol which is representative f a major class of solutions.

EY-WORDS: protocol formalization (specification and validation), synchronization protocol, distributed data bases, duplicated data, abstract data types.

1) This research is sponsored by INRIA-ADI (SIRIUS project) under contract # 80003.

1. INTRODUCTION

Many researchers have recently presented solutions to concurrency control for distributed data systems. These solutions take form of synchronization protocols which have been designed to coordinate the remote processes in charge of local data (called "controllers") during an update session.

Very few proposals have been made to formally specify and validate ("formalize") these protocols.

This paper is primarily concerned with this crucial aspect; we introduce a formal methodology based on algebraically-specified data types to formalize existing protocols.

This article encompasses two major sections:

- the first one presents a clear definition of mutual integrity which turns out to be the basic requirement which must be verified by a synchronization protocol and introduces our formalism.
  This concept is translated in terms of our model through mutual-integrity theorems which are recalled.

- the second section illustrates our approach with a synchronization protocol (for duplicated entities) which has been largely referenced in the literature (namely THOMAS' one).

2. INTEGRITY CONCEPTS

We shall in turn examine the integrity concept in centralized and distributed data base management systems (DBMS).

2.1 INTEGRITY IN A CENTRALIZED DBMS

A data base can be viewed as a collection of entities and constraints whose values define VALID states of the data base. The concept of integrity (consistency) of a DBMS is twofold (MIRA80-b):

- internal integrity.

- external integrity.

Internal integrity is associated with integrity constraints defined on data to meet real-world restricitons.
The concept of transaction (GRAY78) has been introduced in centralized DBMS (and naturally extended to distributed DBMS (GRAY79)) to represent the atomic interaction of the user with the data base which preserves internal integrity.

External integrity corresponds to the control of concurrent transactions which may conflict in sharing common data (problems of "lost update", "dirty read", ...) A serialization mechanism (locking is the one which has been almost exclusively elected) must be defined to ensure external integrity.

2.2 INTEGRITY IN A DISTRIBUTED DBMS

Internal integrity in a distributed DBMS is called MUTUAL INTEGRITY when remote entities are involved in a (global) transaction; identity is a particular case of an integrity constraint which leads to the well-studied problem of duplicated entities. There on, we shall mainly consider the latter aspect.

We say that a synchronization protocol ensures MUTUAL INTEGRITY when the manipulated entities converge to the same state should update activity cease. (THOM75)...

A distinction between STRONG and WEAK mutual integrity has been proposed by several authors (SEGU78), (LELA79)...

However this distinction has been rather vague or incorrect since the underlying concept was itself vague or incorrect; for example in (LELA79) or (SEGU78) the distinction is based upon the SIMULTANEITY concept among remote states and this turns out to be delicate since no site can ever know the state of the entire distributed system (MONT78),(GRAY79),...

We propose a distinction based on the AVAILABILITY concept; an entity is said to be available when it is stable (no modification in progress) and open to a retrieval access.

Mutual integrity is said to be WEAK whenever a synchronization protocol enables to have two TEMPORARILY different available versions of the same copy at a given time (a transaction may retrieve consistent entities which are not the most current); it is said to be STRONG otherwise (the retrieved data are the most current).

We can make a parallel between this definition of mutual integrity (strong and weak) and the levels of consistency defined in centralized DBMS like SYSTEM-R (GRAY75); in this latter case, strong integrity correspond to the third level, weak integrity to the second level while the first level of consistency can be considered as a "weaker" form of integrity (access to dirty data is possible at that level). Our definition of availability precludes this latter type of consistency.

External integrity refers to the control of concurrent confliciting transactions which can be initiated anywhere in the underlying network.

In centralized systems there exists a control locus where shared COMMON memory is used for coordinating concurrent conflicting transactions.

Let us consider the type of control we may have in distributed data systems.

In a distributed system, which can be defined as a collection of processes communicating only through message-passing, three types of CONTROL LOCI have been chosen for synchronization protocols:

(i) WITHIN A SITE; the concurrency control is said to be CENTRALIZED (or VERTICAL) by analogy with local systems (MENA77), (GARC79)...

(ii) WITHIN A PROCESS; there is no privileged site. Each site is functionnaly homogeneous. There exists a given process (which we call MASTER CONTROLLER) responsible for the whole synchronization session.
The control is said to be "PARTIALLY DECENTRALIZED"; the protocols of ELLIS (ELLI77)..., LE LANN (LELA76)..., BUSTA (BUST78), POPEK/MIRANDA(POPE79) ... are of this type.

(iii) WITHIN A MESSAGE; the initiator of the synchronization session and the initiator of the global update are (generally) different controllers. A special synchronization message propagates control data from controller to controller (like OK votes in THOMAS' protocol (THOM75)).
The control is said to be "FULLY DECENTRALIZED".

In the last two cases, the control is said to be HORIZONTAL; from point (i) to point (iii), the tendency is towards a reduction of the time for which a host has control over the progress of a protocol.

A "primary update token" that moves around the network and symbolizes control is a particular case of centralized control technique ("circulating centralized control") which has been proposed by several authors (WILM79-b),

The protocol we formalize in this paper presents the following characteristics:

INTEGRITY \ PROTOCOL	THOMAS'
Mutual integrity	WEAK
External integrity (type of control)	Fully-decentralized control
(type of consensus)	(majority)

Figure 1.
Major characteristics of THOMAS's protocol.

3. ABSTRACT DATA TYPES

3.1 ABSTRACT-DATA TYPE CONCEPT

A data abstraction is a behavioural representation-free description often using formal notation of a data object and the operations upon it.

Data abstractions are realized in programming languages by abstract data types (ADT) which isolate the representational detail from other programs units.

Although the ADT concept has largely been adopted by language designers (WULF76), (TARD77), (GUTT78),... the concept is really language-independent and can very naturally be carried over in layered systems (operating systems, data base management systems,...).

There are two properties of ADT which appear to be appealing for distributed systems:

(i) enclosure and implementation hiding (data independence; user transparency).

(ii) abstractional power (separability of functions; flexibility).

Two families of object-oriented languages (encompassing ADT's) have been proposed, the propositional one (MILN71), (HOAR72), (WULF76) and the algebraic one (BURST77), (GOGU78), (GUTT78).
We elected the algebraic approach which seems to be more adequate to complex structure formalization (PAOL77), (LOCK78), (MELK78),...

3.2 ALGEBRAIC APPROACH (GOGU76), (TARD77), (GUTT78).

We briefly recall the major features of the algebraic approach for ADT's as presented in (GOGU76).
An ADT can be defined as a MANY-SORTED ALGEBRA:

- an algebra of ONE SORT is roughly speaking a set of objects and a family of operators on the set; the set is called the carrier of the algebra.

- MANY-SORTED ALGEBRA extends this definition by allowing the carrier of the algebra to consist of several disjoint sets; each of these sets is said to have a SORT; the operators are sorted and typed but must be closed with respect to the carrier.

Two basic kinds of sorts are involved in an ADT specification: the sort being defined and any number of sorts assumed previously defined in a similar fashion (i.e. the Boolean sort includes the usuals constants T and F, and is assumed to have been defined in its own right with the same methodology; other built-in ADT's we may use are INTEGER, QUEUE,...)

Operators of the algebra are indexed by pairs (w,s) where $w \in S^*$ (sort of the operand) and $s \in S$ (sort of the results); the symbol $\underline{\Sigma w,s}$ will be used for the set of all operations with index (w,s); $\underline{\Sigma}$ is used for the union of all the sets $\Sigma w,s$ and is called the "signature" of the algebra.

A Σ-algebra A is a family of sets (As), $s \in S$, called CARRIERS of A which is determined by a triple $<S, \Sigma, E>$ where:

S is the set of "sorts", S : {s1, s2,...si,...sn} denoting the various types of objects which are required for that definition.

Σ is the set of operations $\Sigma = \{\Sigma,w,s\}$, whose operands and results are objects making up the sorts in S (SYNTAX description)

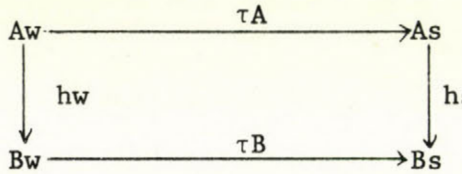
E is the set of equations which describe the semantics of each $\Sigma w,s$ of Σ; each algebraic equation (or axiom) defines the results of various combinations of operators applied upon various operands.

For every $(w,s) \in S^* \times S$ and every $\tau \in \Sigma w,s$ the function $\tau A : As1 \times As2 \times \times Asn \rightarrow As$ with w = s1,s2,...sn is called "A-operation named by τ"

Two many-sorted algebras are called Σ-algebras if they have the same signature.

A very important concept is the one of Σ-homomorphism; Definition: given two Σ-algebras A and B, a Σ-homomorphism, $h:A \rightarrow B$, is a family of functions $< hs:As \rightarrow Bs, s \in S >$ mapping each carrier in A into the corresponding carrier in B while preserving the operations, i.e.

$\forall \tau \in \Sigma w,s$ with $w=s1,s2,...sn$ and $(a1,...an) \in As1 \times As2 \times ..Asn$ we have $hs(\tau_A(a1,...an)) = \tau_B\ hs1(a1),...hsn(an))$; this can be visualized by the following diagram commutation:



We shall use the Σ-homomorphism concept to express:

(i) parallelism among remote operators (Σ-homomorphism)

(ii) layered abstractions between the distributed data base and the underlying transmission facility on one hand; between the distributed data base and the local DBMS on the other hand.

It is important to note that the ADT definitions of a particular type is not unique; however, it should meet the following goals:

(1) The operators should not be redundant.
(2) The equations must not be contradictory.
(3) The operators and axioms should be as simple as possible.
(4) The equations should be constructed in such a way that leads forcibly to unique reduction.
(5) We shall use a formalism close to OBJ-O (TARD77), (GOGU78), (GOGU76), (TARD79).

Which represents one of the basic language encompassing algebraically-specified data types; Guttag' systems for symbolic execution of ADT's seems to suffer some inadequacies mainly at the sintactic level (TARD79).

3.3 OBJ-O

OBJ-O is an object-oriented language defined by GOGUEN(GOGU76) and implemented by TARDO (TARD77); this language is very close to NPL language, now called HOPE defined in (BURS77-b). In OBJ-O, an algebra is a 4-tuple:

< SORTS, OPS, VARS, SPECS > where

SORTS, OPS, SPECS correspond respectively to S,Σ,E and VARS includes the definition of the working variables used in the axioms.

The error-operators ("ERROR-OPS") and their semantics ("ERROR-SPECS") may be naturally defined in this language; an extensive discussion of "error algebras" is presented in (GOGU77).

The general syntax of an operator is given by:

OP	:	S1,S2,.............SN $\longrightarrow$	Sp
Operator name (under bars are used for place-holders)		Operand sorts (N: arity of the OP operator)	results sort

Prefix, infix, postfix, distributed-fix declarations are possible in OBJ-O.

Each sort has an equality relation which is built-in with syntax:

-:= - : S   S$\longrightarrow$S

Hidden operators may be declared with the key-word "HIDDEN" placed after the result sort.

We shall often use conditional equations in the algebraic specification; a formal theory of conditional equations in ADT's is developed in (THAT76).

In the specifications of a synchronization controller using ADT's, we shall neither use place-holders in OPS nor consider ERROR operators.

4. FORMALIZATION OF A SYNCHRONIZATION PROTOCOL

The concept of ADT should be the lowest common denominator which maximizes every one's naturalness at some level of expression. In the area of sycnhronization protocols, the three basic operators (framework for uniformization/abstraction) which may receive a general consensus, are:

- PREPAREG; this operator corresponds;

  (i) to the local initialization which consists of all the actions needed for the transaction to start consistency enforcement: time-stamp acquisition (THOM75), new-value computation (THOM75), ticket allocation (LELA78), priority definition (ELLI77), (BUST78), (POPE79),...

  (ii) to the global initialization which includes the strategy to resolve conflicts: first revolution in the virtual ring (ELLI77), first step of synchronization (POPE79),..., to check security (POPE79), to perform temporary update (LELA78), (POPE79),....

- SETG; this operator consists of the propagation or broadcasting of the (permanent) update.

- UNSETG; this operator concerns the (robust) commitment phase (log synchronization (POPE79),...)

The work on reliability is much less developed than that on external consistency and in many proposals, UNSETG is embedded in SETG, as in (THOM75)... This corresponds to the fact that there exists a clear dichotomy between performance-oriented and robustness-oriented protocols.

The above operators apply to a "global virtual object"; the global virtual object consists of each local object semantically tied during a manipulation. A duplicated entity may be seen as a logical entity (the global virtual object) physically stored in several remote sites.

This "locality" concept receives a formal development in (BILL79).

The global virtual object is characterized by a set of (global) states in each participating controller. Each operator maps an input global state and a set of input messages into an output state and a set of output messages.

Protocols are said to be message-driven: reception of a synchronization message causes a series of actions (and eventually a message transmission) to be executed.

This concept of (global) STATE enables us to express the semantics of the attached manipulation operators in a simple way. A change of state is the result of the execution of an operator bound to the occurence of an event (here a synchronization-message).

These states enable us to specify the different protocol steps; they allow:

(i) The simplification of operator semantics.

(ii) The simplification of the validation process.

(iii) The setting of clear re-entry points for the recovery procedure.

(iv) The expression of parallelism among remote controllers by using homomorphisms among remote states.

Our model encapsulates the functionning of a controller with a Σ-algebra, called SYNCH, whose carrier is the global-entity state and operators, (PREPAREG, SETG, UNSETG), a reduced set of primitives attached to particular message receptions.

A synchronization protocol is depicted as a set of Σ-homomorphism ALGEBRAS.

This approach offers the same advantages as Petri-nets (representation independence...) with additional ones such as:

- simple rigorous formalism for inferring proofs of correctness.
- semantic framework from which correct implementations can be derived (automatically) (GUTT78), (TARD77), (NOUR79),...
- ROBUSTNESS integration in a natural way (in (POPE79), we define the semantics of a message reception indicating the failure of a cooperating controller).

4.1 GLOBAL ENTITY STATES

For each of the three generic operators indicated previously we attach an input and an output (global) state. Among these states two of them appear to be important;

- the available state, noted F, before the actual modification (initial state) an F' after (final state); the local entity (copy) can be retrieved.
  F represents the input state of PREPAREG and F' the output state of UNSETG.
- the unstable (unavailable) state noted U which corresponds to a current modification of the local entity which cannot be retrieved.

4.2 CANONICAL REDUCTION; SERIALIZATION

Every sequence of operators of SYNC is reduced to a sequence (PREPAREG)-(SETG)-(UNSETG), called canonical reduction.

A transaction is said to be ATOMIC if canonical reductions of this transaction are identical in each controller involved in the synchronization session.

Two confliciting transactions are said to be SERIALIZED if their respective canonical reductions do not interfere in the modification phase (SETG).

4.3 MUTUAL-INTEGRITY THEOREMS

Before recalling the two mutual-integrity theorems (whose proofs are given in (MIRA79)), let us give an extension of the Σ-homomorphism definition; we say we have a TOTAL Σ-homomorphism in a synchronization protocol whose each controller is specified by a Σ-algebra SYNC if:

$\forall i$, $\forall j$ SYNCi and SYNCj are Σ-homomorphic, $i,j \in (1,n)$ with n number of duplicated entities.

THEOREM 1: Strong-mutual integrity theorem

A synchronization protocol ensures strong-mutual integrity if:

(i) transactions are atomic and there exists a TOTAL Σ-homomorphism (on output states).

(ii) conflicting transactions are serialized.

THEOREM 2: Weak-mutual integrity theorem

A synchronisation protocol ensures weak-mutual integrity if:

(i) transactions are atomic and there exists a PARTIAL Σ-homomorphism (on input states).

(ii) for every unstable state, there is a transition $U \longrightarrow F'$ on each controller.

(iii) for the Σ-algebras SYNC which are not Σ-homomorphic there exists a morphism φs ensuring the following diagram commutation.

$$
\begin{array}{llcll}
\text{ith controller:} & (F)i & \longrightarrow & (U)i & [v\ (F)i] \\
 & \downarrow \varphi s & & \downarrow \varphi s & \\
\text{jth controller:} & (F)j & \longrightarrow & (U)i & \underbrace{[v\ (F)j]}_{\text{rejection case}}
\end{array}
$$

(iv) conflicting transactions are serialized

NOTE: A morphism φs between two global-entity states represents a FUNCTIONAL CORRESPONDENCE (not necessarily a simultaneity) between the operators generating these states; as a matter of fact, the morphism is associated with the synchronization message sent during the generation of the considered state. The successive morphisms represent an event-sequencing scheme.

5. FORMALIZATION OF THOMAS' PROTOCOL

Interested readers are urged to get a look at the quoted references concerning this protocol since we shall only sum up their basic properties.

5.1 PRINCIPLE OF THOMAS' SOLUTION (THOM75), (THOM77)

THOMAS' protocol is based on a MAJORITY CONSENSUS mechanism. Each Data Base Manager Process (DBMP) - there is one DBMP in every site - VOTES on the acceptability of update requests. For a request to be accepted and applied to all data base copies, only a majority of DBMP's need to aprove it. The request is said to be RESOLVED when a majority of DBMP's accepted or rejected it.

The initical step of this protocol concerns the acquisition of the BASE-VARIABLES (BV's) by the originator of the request (called application program or A.P.) from any DBMP (INTREQ message). The DBMP replies to the AP (BVTS message) by sending it the BV values along with the attached time-stamps (a time-stamp represents the last modification date of a given BV). The AP calculates the new values of the BV called the UPDATE-VARIABLES (UV's).

The global update session is initialized by the sending of the update request (EXTREQ message) which encompasses both the set of BV's with their time-stamps and the set of UV's.

The initial version of this protocol (THOM75) used a DAISY-CHAIN during the resolution phase: each DBMP votes when receiving an EXTREQ, and forwards the request along with the accumulated votes to another DBMP that hasn't voted yet if a consensus is not reached. This procedure continues until the request is resolved (a request is said to be "pending" till this resolution). The voting rules basically amount to voting:

O (OK): Each BV is current and there is no conflict;
If a majority consensus is attained (on OK votes) the request is globally accepted; the acceptation is notified to each DBMP by the sending of the UPD message.

R (Rejected): There is an obsolete BV. The rejection is notified to each DBMP with the REJ message (one reject vote is enough to globally reject a request which could be later resubmitted).

A weak form of rejection not considered here, is proposed in (THOM77).

P (pass): Each BV is current but there exists a conflict with a higher priority pending request (which received an OK vote).

A conflict between Ti and Tj correspond to $(BV's)_i \cap (UV's)_j \neq \emptyset$.

If a majority consensus is obtained on PASS votes the request is globally rejected.

D (defer): defer voting when:

- either each BV is current but there exists a conflict with a lower-prority pending request.

- or request-BV time-stamps are more current than the local ones which are obsolete (a previous update was not yet performed).
  This case corresponds to the weak integrity feature.

  These deferred request are queued (in Q).

5.2 THOMAS' PROTOCOL FORMALIZATION

A controller is formalized by an ADT named SYNC whose signature is indicated in the following figure.

OBJECT	OPERATORS	COMMENTS
Global entity (GE)	PREPAREG (M, GES); SETG (M, GES); UNSETG (GES); ID (GES);	M : Messages GES: Global entity state ID: Identity operator

Fig. 2.
SYNC Signature

OBJECTS	OPERATORS	COMMENTS
Message (M) M∈ (EXTREQ, UPD, REJ)	TRANSMIT (M, PR,DBMPL, I,J); BROADCAST (M, PR, LIST); RECEIVE (M,PR,DBMPL,I,J, GES, PR); WAIT (M,PR);	DBMPL: List of controllers which voted IJ: counters LIST: identification of receiving controllers (may be "ALL") GES,PR: local parameters PR: priority we indicate the parameters of importance in a message transmission/ reception
Copy status (CS) CS (STB,USTB)	ID (CS)	
Global Request Status (GRS) GRS∈(Λ,P,A,R)	ID(GRS)	Λ : none P : pending A : accepted R : rejected
Local Request Status (LRS) LRS∈ (OK,D,PS,RJ)	ID (LRS)	OK : OK vote D : deferred PS : pass RJ : rejected
Local entity (LE)	PREPAREL (LE); SETL (LE); UNSETL (LE);	
TRANSACTION (T)	PROCESS (T)	

Fig.3.

Signatures of other involved types

Other basic operators used in the specifications are those attached to:

- INT, BOOL, like TEST (A,B): = IF A = B THEN TRUE ELSE FALSE
  SUP (A,B): = IF A > B THEN TRUE ELSE FALSE
- LIST (L) like APPEND (i,L),...
- QUEUE (Q) like ENQ,DEQ,EMPTY?,...

A global entity state (GES) is a 3-tuple <CS, GRS, LRS> which may take 4 basic forms:

$$(STB,\Lambda,-),\ (STB,P,-),\ (STB,R,-),(USTB,A,-)$$

where " - " represents a non-specified local parameter (OK,RJ,D, or PS).

The specifications of SYNC for THOMAS' protocol are given in Annex 1.

Parallelism

We want to express the parallelism between two participating controllers namely SYNCi and SYNCj; parallelism between controllers i and j will be depicted by a Σ-homomorphism between SYNCi and SYNCj.

In order to do so we introduce the following morphism φ associated with synchronization messages which defines a correspondance between remote global states:

$$(STB,\Lambda,-) \xrightarrow{\varphi[\text{none}]} (STB,\Lambda,-)$$

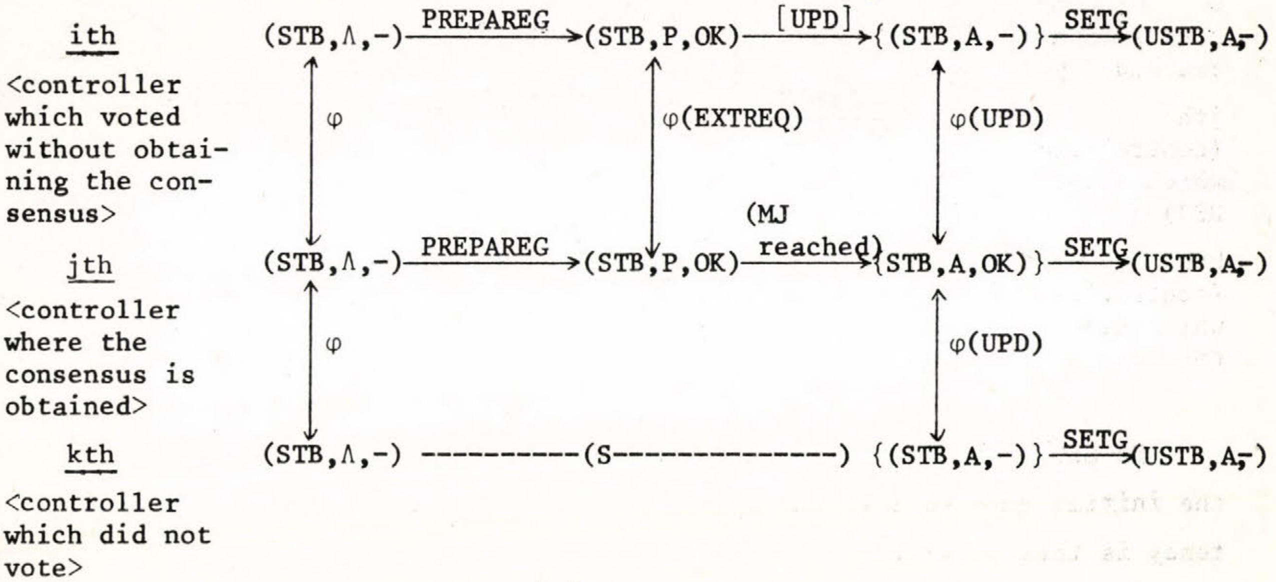
$$(STB,P,-) \xrightarrow{\varphi[\text{EXTREQ}]} (STB,P,-)$$

$$(USTB,A,-) \xrightarrow{\varphi[\text{UPD}]} (USTB,A,-)$$

Assertion 1: In an environment without concurrency conflicts, the protocol ensures weak mutual consistency

The verification of theorem 2 is straight-forward with $(STB,\Lambda,-) = F$ and $(USTB,A,-) = U$. As a matter of fact we get the following diagram commutations (attached to a given transaction) by making use of the Σ-algebras equations.

NOTE: We represent the three types of controllers which may exist in the distributed system.

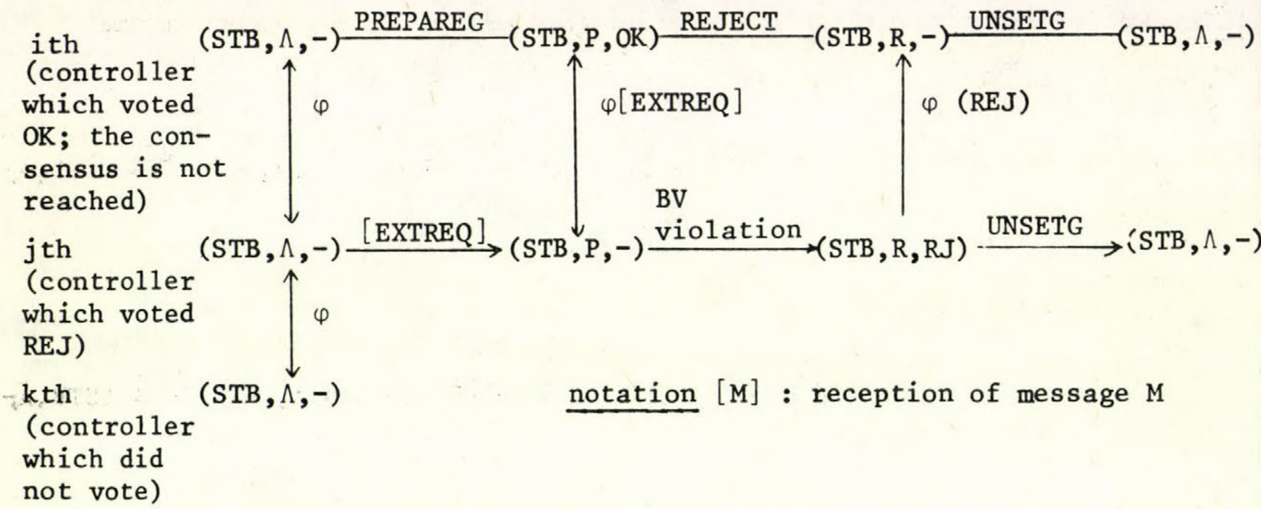


notation: [M] : reception of message M

In the following diagrams, we shall not represent the intermediate states (between {}) to alleviate the representations.

NOTE: If we integrate internal-integrity violation in this scheme we get the following commutation which ensures that the final global state (available for retrieval/update) is identical to the initial one.

We introduce the following morphism $(STB,R,-)\xrightarrow{\varphi(REJ)}(STB,R,-)$



No SETG has been performed; therefore, the final states correspond to the initial ones (before the synchronization session). The mutual consistency is then verified.

Assertion 2:

The protocol ensures weak mutual consistency when there is a finite set of concurrent conflicting transactions

Proof:

The proof of this assertion may be reduced to two concurrent transaction Ti and Tj (with PRi > PRj) since there is a total ordering of transactions.

Two major cases may occur depending on the fact that:

(i) The lower priority transaction (Tj) gets the majority consensus on OK before Ti.

(ii) Tj gets the majority consensus on PASS votes and is rejected.

We will use the indices i and j for φ to indicate the belonging of a correspondence to the ith or jth session. In ANNEX 2 Figure 4 (rejection of T and Figure 5 (rejection of Tj) depict the most general situations which can arise and show the weak mutual consistency.

6. CONCLUSION

The difference between this model and the others are a reflection of different goals; our model is an attempt to provide:

(i) a minimization of the primitive concepts: a framework for protocol uniformization/abstraction leading to the concept of synchronization-protocol transparency (this introduces a new degree of transparency to the four types of transparency presented in (TRAI79). Whatever the synchronization protocol is, we pointed out three generic primitive operators which represent the only knowledge of the inner and outer layers where the protocol is used; the operator semantics (depending on the protocol) is hidden and can be switched according to the suitability (strong or weak-mutual integrity,...) of the chosen protocol.

(ii) a way to express synchronization protocols clearly such that the effects of failures are formally specified (POPE79).

(iii) a basis from which SIMPLE (visualisable) and RIGOROUS proofs of correctness can be inferred.

(iv) a global architecture for a distributed-data-base integrity system; a synchronization protocol corresponds to a functional layer with a clear mapping to a local DBMS and to an END-to-END communication protocol.
The formalism is extendable to the functional-layer specification of a data-base-management system (models and manipulation languages) and a computer network (entities and protocols); our formalism is not constrained to synchronization protocols.
This represents a salient feature of the ADT-based approach.

(v) the expression of basic integrity concepts (mutual integrity theorems; serialization; atomicity,...).

Those points are not addresed in the other existing proposal made by ELLIS (ELLI77-b); ELLIS proposes a formalism based on L-SYSTEM for his protocol (in a fail-safe environment only); the specifications are visualizable and simple; however the proofs are very complex and dependent on the size of the network (small preferably).

There is another proposal made by WILMS (WILM79-a) with a formalism based on NUTT's networks but only concerned with the specification aspect. This article presents a new application of abstract data types; it aims at demonstrating that the ADT concept can be applied with a good profit to a typical distributed-data-base problem.

ACKNOWLEDGEMENTS

Special thanks are due to Gerry POPEK for invaluable discussions on this topic and to the numbers of the SIRIUS-INRIA group (animated by G. LE LANN) in which the ideas presented here were freely discussed and explored.

7. REFERENCES

(ALSB76) ALSBERG,P.A. "Multi-copy resiliency techniques" University of Illinois research report, CCTC-VAD-6-505,May 76.

(BILL79) BILLER,H., EBERHARD,L. "On the evaluation of architectures and applications of distributed data base management system". Seminar on distributed data sharing, Aix en Provence; France, May 15-17-1979.

(BURST77-a) BURSTALL,R.M., GOGUEN,J.A. "Putting theories together to make specifications". Proc. 5th Int. Conference on aritifical intelligence. Mit/1977, pp 1045-1058.

(BURST77-b) BURSTALL,R.M. "Design considerations for a functional programming language". Proc. Infotech Symposium, Copenhaguen, Oct. 1977.

(BUST78) BUSTA,J.M. "Integridad externa da una base de datos repertida en una red de ordenadores general heterogénea". PH-D Thesis, Univ.de Santiago de Compostela, Spain 1978. (CITEMA award of the best doctoral thesis in Spain.)

(ELLI77-a) ELLIS,C.A. "A robust algorithm for updating duplicated data bases". Workshop on distributed systems, Berkeley, 1977, pp.146-160.

(ELLI77-b) ELLIS,C.A. "Consistency and correctness of duplicate data systems". Proc. of the 6th Symposium on OS principles, 16-18.

(GARC79) GARCIA-MOLINA,H. "Performance of update algorithms for replicated data in a distributed data base". PH-D Thesis Univ. of Stanford, June 1977 (report n° STAN-CS, 79-744).

(GRAY75) GRAY,J. et al. "Granularity of locks and degrees of consistency in a shared data base".
Modelling in DBMS, G.M. Nijssen editor, North Holland, pp.365-394 (also IBM research report RJ 1606, 1975).

(GRAY78) GRAY,J. "Notes on data base operating systems"
Lecture notes in computer science, edited by J.Hartmanis, Springer Verlag, Berlin, New York, 1978, pp. 394-481.

(GRAY79) GRAY,J. "A discussion of distributed systems".
Proc. AICA, Bari, Oct. 10-13, 1979, pp. 204-211.

(GOGU76) GOGUEN,J.A. et al."An initial algebra approach to the specification, correctness and implementation of abstract data types".
IBM research report RC-6487, Oct. 1976.

(GOGU78) GOGUEN,J.A. "Some design principles and theory for OBJ-0 a language for expressing and executing algebraic specifications of programs".
Proc. Int. Conference on Mathematical studies of Information processing, Kyoto, Japan, 1978, pp. 429-475.

(GOGU78-a) GOGUEN,J.A., BURSTALL,R.M. "Some fundamental properties of algeraic theories: a tool for semantics of computation".
UCLA research report, July 1978..

(GOGU78-b) GOGUEN,J.A., TATCHER,J.W., WAGNER,E.G. "An initial algebra approach to the specification, correctness and implementation of ADT".
Current trends in Programming Methodology, Vol n° 4, Data Structuring (Ed. by R.Yeh) Prentice Hall 1978, pp.80-149.

(GUTT78) GUTTAG,J.V. et al. "Abstract data types and software validation"
CACM, Dec. 1978, Vol 21, number 12, pp. 1048-1064.

(HOAR72) HOARE,C.A.R. "Proof of correctness of data representations". Acta Informatica, 1,4 (1972) pp. 271-281.

(LELA76) Le LANN,G. "Introduction à l'analyse des systèmes multi-référentiels" Thèse d'Etat, Univ. de Rennes, France, Mai 1976, 198. p.

(LELA77) LE LANN,G. "A protocol to achieve distributed control in failure tolerant multi-computer systems". SIRIUS research report, IRIA, CTRI 002-1977.

(LELA78-a) LE LANN,G. "Algorithms for distributed data sharing systems which use tickets" SIRIUS research report, SYN 1003, 1978. (Proc. 3rd Berkeley Workshop, San Francisco, August 1978, pp. 259-272.

(LELA78-b) LE LANN,G. "Pseudo-dynamic resource allocation in distributed data bases" Proc. of the ICC, Kyoto, Japan, Sept. 1978

(LELA79) LE LANN,G. "Consistency and concurrency control in distributed data base systems". Presented at the ECC course on distributed data base Sheffield (U.K.) July 9-20, 1979 (SIRIUS Report, INT 1-1007.)

(MENA77) MENASCE,D., MUNTZ,R., POPEK,G. "A locking protocol for resource coordination in distributed systems" UCLA research report, 1977

(MILN71) MILNER,R. "An algebraic definition of simulation between programms". Stanford AI project, Memo AIM - 143, Feb 1971, 21 pp.

(MIRA79) MIRANDA,S., BUSTA,J. "Interference problem in distributed data systems". Proc.5th Conf. on the theory of operating systems Visegrád, Hungary, Jan 31. Feb 3, 1979.

(MIRA80-a) MIRANDA,S. "Specification and validation of two ring-structure synchronization protocols for distributed data bases". Proc.Int. Symposium on distributed data bases, Paris, March, 12-14, 1980.

(MIRA80-b) MIRANDA,S. "Aspects of data security in general-purpose data base management systems". Proc. 1980 Symp. on data security and privacy, IEEE, April 14-16, 1980, Berkeley, CA.

(MIRA80-c) MIRANDA,S. "Architecture formelle d'un système d'intégrité pour une base de données répartie". Thèse d'Etat, Univ. Paul Sabatier, Toulouse, France, Sept.1980.

(MIRA80-d) MIRANDA,S. "DLP:a fault-tolerant decentralized locking protocol for distributed data bases". Proc. FTCS,10,Kyoto, Japan. Oct. 1980.

(MONT78) MONTGOMERY,W.A. "Robust concurrency control for a distributed information system" PH-D Thesis, MIT/LOCS/TR-207, Dec 1978.

(NOUR79) NOURANI,F."Constructive extension and implementation of abstract data types and algorithms" PH-D Thesis, UCLA,ENG-7945, 172 p. August 1979.

(PAOL77-a) PAOLINI,P.,PELAGATTI,V. "Formal definition of mappings in a data base" Proc. ACM SIGMOND Conference, Toronto, 1977, pp. 40-46.

(PAOL77-b) PAOLINI,P.,PELAGATTI,V. "Formal definition of data models". Instituto d'Ellectrotecnica-Politecnico de Milano, Research report, n° 77-2, Feb. 1977.

(POPE79) POPEK,G., MIRANDA,S. "Formal solution to the external integrity problem in distributed systems using abstract data types". UCLA and CERISS research report,n° 79038, May 79 (revised version forthcoming).

(SEGU78) SEGUIN,S. "Traitement distribué d'informations réparties dans un réseau d'ordinateurs"
Thèse d.Etat, INPG, Marc 1980.

(TARD77) TARDO,J. "OBJ and the automatic implenetation of abstract data types".
PH-D Thesis, UCLA 1977

(TARD79) TARDO,J. et al. "A practical method for testing algebraic specifications".
UCLA quarterly report, Dept, of computer science, UCLA, pp. 59-71.

(THAT76) THATCHER,J.N. et al. "Specification of ADT using conditional axioms".
IBM research report, RC 6214. Sept. 1976.

(THOM75) THOMAS,R.H. "A solution to the update problem for multiple copy data bases which use distributed control"
BBN report ≠ ≠ 3340, July 1975.

(THOM77) THOMAS, R.H. "A majority consensus approach to concurrency control for multiple copy data bases"
BBN report,n° 3733, Dec. 1977, 51.p.

(TRAI79) TRAIGER,I.L. et al. "Transactions and consistency in distributeddata systems"
IBM research report, 1979, (to appear)

(WILM79-a) WILMS,P. "Etudes d'algorithmes de cohérence d'informations dupliquées et réparties" -Formalization à l'aide des rèseaux de Nutt".
Research report, INP Grenoble, RR n° 160, 160 bis, Feb. 79.

(WILM79-b) WILMS,P. et Al "A majority consensus algorithm for the consistendy of duplicated and distributed data" Research report, Univ. of Grenoble, March 79, (Proc.1st Int. Conf. on DCS, Huntsville, Alabama, Oct. 1-5, 1979).

(WULF76) WULF, W.A. et Al "Abstraction and verification in ALPHARD" Internal Report, 1976, CACM, Vol n° 20,8, pp. 553-563, August 1977.

ANNEX 1: Specifications of THOMAS' protocol in an OBJ- close language

```
OBJECT      SYNC (ith controller)
SORTS       GES,LE,INT,BOOL,T,M, PR,LIST,QUEUE,GRS,LRS,CS
OPS         PREPAREG    :  M × GES ────────→ GES
            SETG        :  M × GES ────────→ GES
            UNSETG      :  GES ────────────→ GES
            ID          :  GES ────────────→ GES

VARS        PR : INT  <priority (PR,i) will be noted Pi>
            i  : INT  <controller number>
            STB: CS ; USTB : CS ;
            P  : GRS;A      : GRS; R   :  GRS;        : GRS;
            OK : LRS;D      : LRS; PS  :  LRS;     RJ : LRS;
            EXTREQ  : M     < the EXTREQ is built by the AP with the BVTS
                              message>
            UPD     : M     < the updata message is broadcasted to each DBMP>
            REJ     : M     < the REJECTION message>
             We do not consider the messages exchanged with the AP
            PS#  : INT <counter of pass votes>
            OK#  : INT <counter of Ok votes>
            MJ   : INT <majority number>
            DMPL : LIST <list of DBMPs which voted>
            Q    : QUEUE <queue of deferred conflicting requests>

SPECS
<EXTREQ, reception>
RECEIVE (EXTREQ, PRk,DBMPL,PS #, OK#,; (STB,Λ,-),-):=ID(GRS)=P;ID(GES)=(STB,P,-);
                          IF <current base variables>
                          THEN PREPAREG((EXTREQ,PRk,PS#,OK#),(STB,P,-));
                          ELSE BROADCAST (REJ,PRk,DBMPL);
                               ID(LRS)=RJ; ID(GRS)=R; ID(GES)=(STB,R,RJ);
                               UNSETG (REJ,PRk; (STB,R,RJ));
```

```
RECEIVE (EXTREQ,PRk,DBMPL,PS≠,OK≠; (STB,P,-),PRj):=ID(GRS)=P;ID(GES)=(STB,P,-);
        IF <current base variables>
        THEN IF <conflicting updates>; <(BV's)k ∩ (UV's)j≠∅>
        THEN IF  SUP (PRk,PRj) = TRUE
        THEN ID(LRS) = D; ID(GES) = (STB,P,D);
             ENQ (Q,(EXTREQ,PRk));
             TRANSMIT (EXTREQ,PRk,DBMPL,PS≠,OK≠); WAIT (EXTREQyUPDyREJ,PRk);
        ELSE IF TEST (PRk,PRj) = TRUE  revote
             THEN PREPAREG ((ESTREQ,PRk,-,OK≠),(STB,P,-));
             ELSE ID(LRS) = PS; INCR(PS≠); APPEND (i,DBMPL);
                  IF TEST (PS≠,MJ) = TRUE
                  THEN ID(GRS) = R; ID(GES) = (STB,R,PS);
                       BROADCAST (REJ,PRk,DBMPL);
                       UNSETG ((EXTREQ,PRk), (STB,R,PS));
                  ELSE TRANSMIT (EXTREQ,PR,DBMPL,PS≠,OK≠),
                       ENQ(Q,(EXTREQ,PRk)); WAIT (UPDvREJ,PRk);
             ELSE PREPAREG((EXTREQ,PRk,PS≠,OK≠), (STB,P,-));
        ELSE ID (LRS) = RJ; ID(GRS) = R; ID(GES) = (STB,R,Rj);
             BROADCAST (REJ,PRk,DBMPL);
             UNSETG((EXTREQ,PRk),(STB,R,RJ));

RECEIVE (EXTREQ,PRk,DBMPL,PS≠,OK≠); (STBvUSTAB,A,-),PRJ):=ID(LRS)=RJ;
                                    ID(GES)=(STBvUST,R,-) ID(GRS)=R;
                                    BROADCAST (REJ,PRk,DBMPL);
                                    UNSETG((EXTREQ,PRk);(STBvUSTB,R,RJ));

<UPD reception>
RECEIVE (UPD,PRk,-,-,-; (STB,ΛvP,-),PRj) :=
       IF  TEST (GRS,Λ) = TRUE
           THEN PREPAREL(LE); <the DBMP did not participate in the voting>
           ELSE (DEQ(Q,(EXTREQ,PRk)); IF Test (PRj,PRk)=FALSE THEN (ID(GRS)=R;
                                           UNSETG((EXTREQ,PRj),(STB,R,-)));
```

```
      ID(GRS) = A; ID(GES) = (STB,A,-);
         SETG(UPD,(STB,A,-)) ;)
         LOOP WHILE EMPTY ? (Q) = FALSE
              DEQ (Q,(EXTREQ,PRi)); ID(GRS)=R; ID(GES) = (STB,R,-);
              UNSETG((EXTREQ,PRi),(STB,R,-)); <all conflicting transactions
                                                are rejected>

         ENDLOOP

<REJ reception>
RECEIVE (REJ,PRk,-,-,-,;(STB,P,-),PRk) : =
                         ID(GRS) = R; ID(GES) = (STB,R,-);
                         UNSETG (REJ,PRk;(STB,R,-));
                         IF EMPTY? (Q) = FALSE
                         THEN WAIT (EXTREQ,PRj); <for revote>
            <This set of operations will be referred in the diagrams as
             REJECT>
PREPAREG((EXTREQ,PRk,PS≠,OK≠),(STB,P,-)):=INCR(OK≠);ID(LRS)=OK;APPEND(i,DBMPL)
                         ID(GRS) = P; PREPAREL(LE);
                         IF TEST (OK≠,Mi) = TRUE
                         THEN ID(GRS) = A; ID(GES) = (STB,A,OK);
                              BROADCAST(UPD,PRk,ALL);
                              SETG (UPD,PRk;(STB,A,OK));
                         ELSE TRANSMIT (EXTREQ,PRk,DBMPL,PS≠,OK≠);
             ENQ(Q,(EXTREQ,PRk); WAIT (UPDvREJ,PRk);

SETG(UPD,PRk,(STB,-,-):= SETL(LE<PROCESS(T),...>; ID(CS)=USTB; ID(GRS)=A;
                                                  ID(GES) = (USTB,A,-);
                         UNSETG(-,-,;(USTB,A,-));

UNSETG(,-,-;(-,-)) : = UNSETL(LE); ID(CS) = STB; ID(GRS)=Λ; ID(GES)=(STB,Λ,-);
```

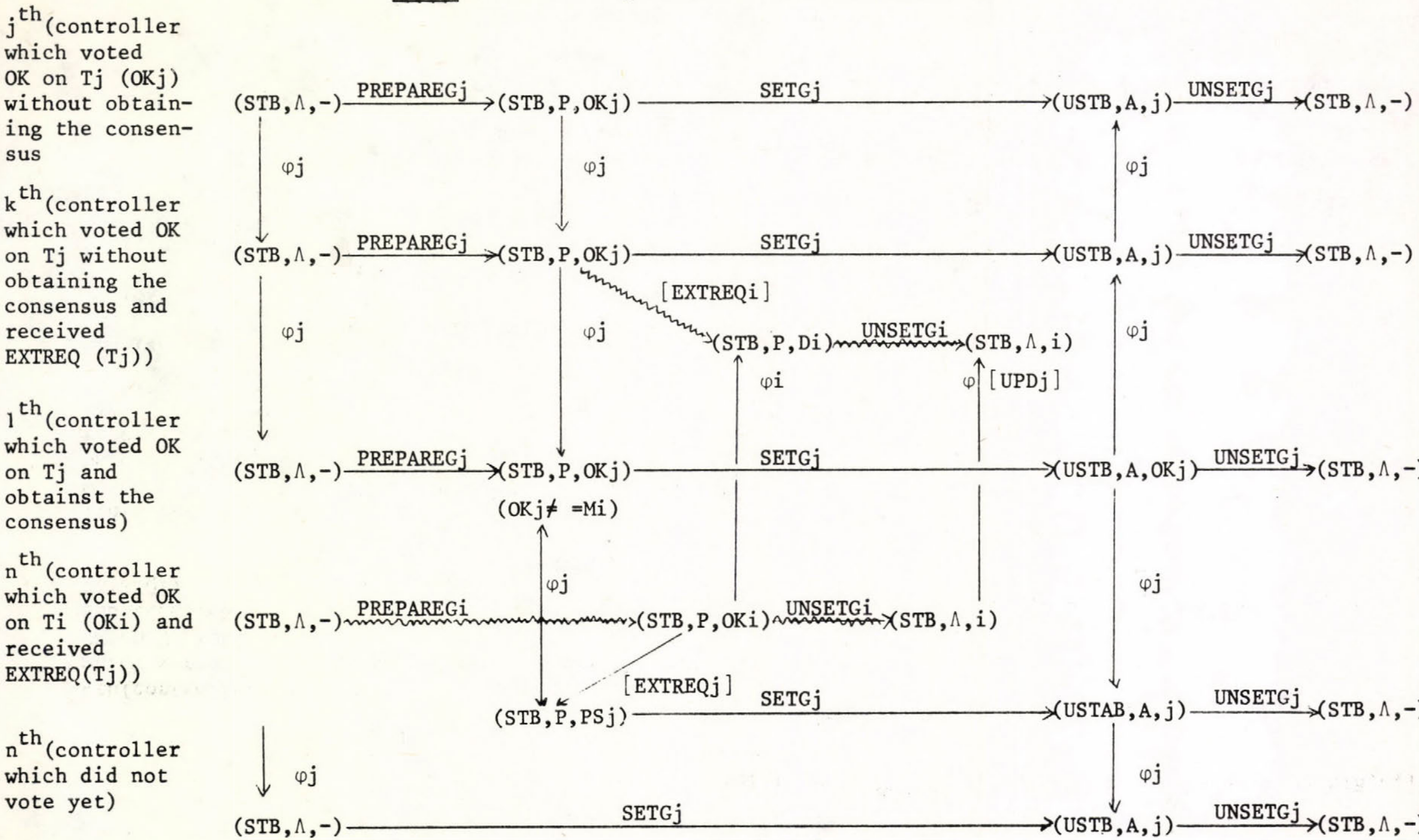
TCEJBO

Notes: (i) received messages are ignored in other configurations (in fact they could be treated like error conditions)

(ii) We do not consider local tests made on current BVs before local update is performed; these tests are related to robustness.

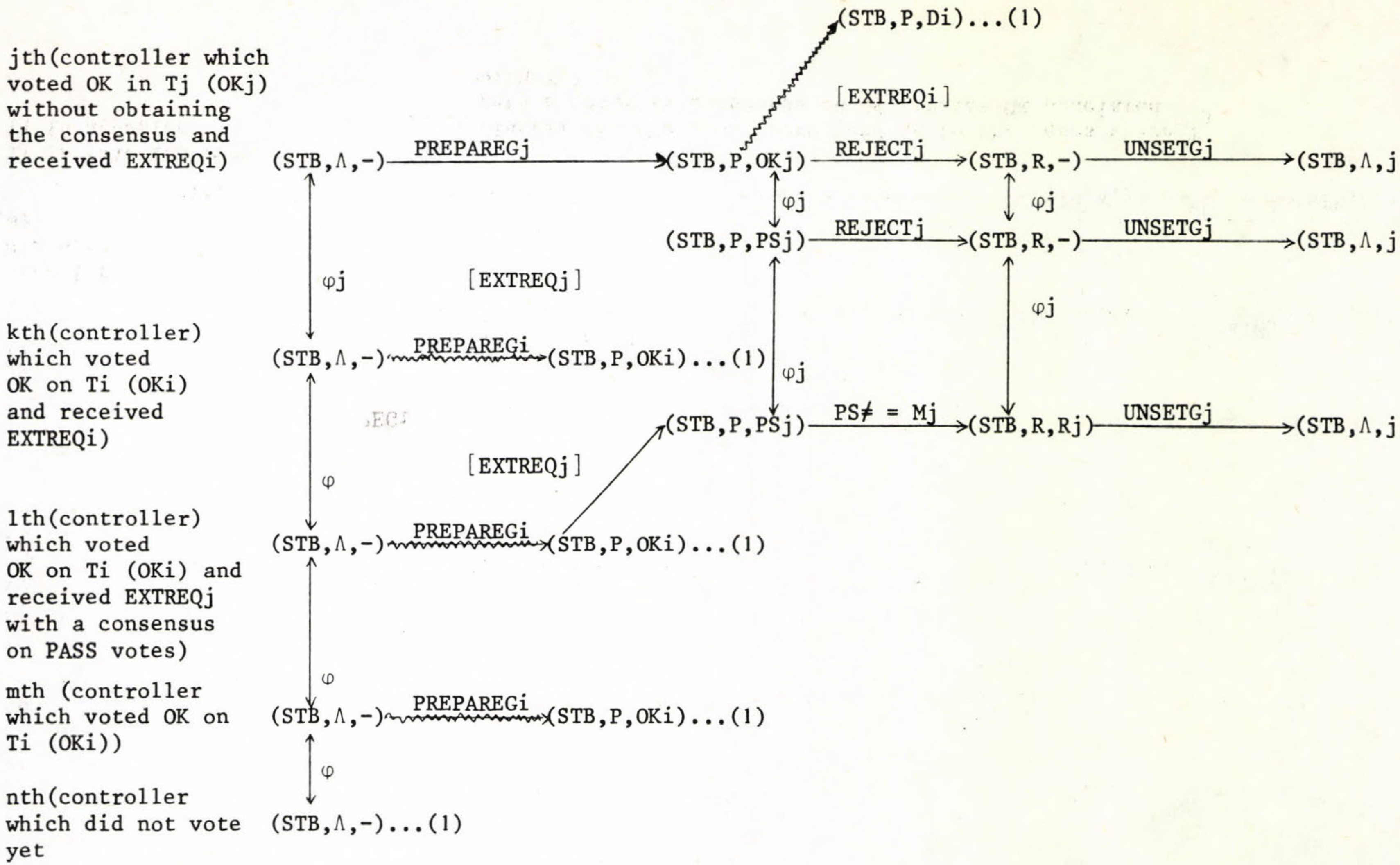
(iii) BROADCAST is here equivalent to TRANSMIT [(n-i) messages] The REj message is broadcasted to the DBMP's which voted (whose identification is in DBMPL) while the UPD message is broadcasted to every concerned DBMP.

Note: [Mi] : message attached to transaction Ti



Note: Ti is rejected when Tj is accepted

Fig.4. Diagram commutations corresponding to the cases where T_j gets a majority consensus on OKj (votes OK assciated with Tj)



Notes: (1) As soon as majority consensus is obtained on OK votes for Ti, we shall get the same diagram commutations as in the assertion 1.

(2) Here we introduced the morphism: (STB,R,-) $\xrightarrow{\varphi[\text{REJ}]}$ (STB,R,-) corresponding to the REJ message.

Figure 5. Diagram commutations corresponding to the case where Tj gets a majority consensus on PSj (Pass votes associated with Tj).

Összefoglalás

Az irodalom kiterjedten foglalkozik osztott adatbázisok szinkronizációs problémáival és sok szinkronizációs protokolt terveztek már. A cikkben absztrakt adattipusokon (algebrai módszeren) alapuló formális megközelítést mutatunk be és kidolgozunk egy egységes módszert a szinkronizációs protokolok leírására és ellenőrzésére. Az eredményeinket egy alapvető protokolon mutatjuk be, amely a megoldások egy széles osztályát reprezentálja.

Р Е З Ю М Е

Формальное описание синхронизации раздельных баз данных

В литературе обширно излагаются синхронизационные задачи разделенных баз данных. Многочисленные протоколы были проектированы. В настоящей работе предлагается формальный подход, основанный на абстрактных типах данных /на алгебраическом методе/ и разрабатывается единый метод для описания и проверки синхронизационных протоколов. Результаты иллюстрируются основным протоколом, представляющим широкий класс решений.

MTA Számítástechnikai és Automatizálási Kutató Intézete, Közlemények 27/1982

A NEW TIMING BASED ALGORITHM FOR CONCURRENCY CONTROL OF DISTRIBUTED DATABASES

ANDRÁS POGÁNY

Computer and Automation Institute
Hungarian Academy of Sciences

1. INTRODUCTION

In the last few years very high research effort has been devoted to the development of new concurrency control algorithms.

The goal of concurrency control is to ensure database consistency despite of paralel database accesses. The problem is presented by an example:

The database is composed of three records A,B,C.
The consistency criteria is A=B+C.
There are two accesses access1 and access2.
access1: A=A+1, B=B+1
built up from the following steps
a1/ read A
b1/ write A
c1/ read B
d1/ write B
access2: B=B-1, C=C+1
built up from the following steps
a2/ read B
b2/ write B
c2/ read C
d2/ write C

In spite of that each access executed alone, preserves database consistency but, the next paralel execution for example will destory that: a1, b1, a2, c1, d1, b2, c2, d2.

The concurrency control algorithm deal only with those accesses which executed alone preserve database consistency. These accesses are callaed transactions. The step of a transaction is an access to a database element (a read or a write).

The sequence of transaction steps built up from the steps of a given transaction set is called a log. Some logs preserve database consistency while others do not. The task of concurrency control is to avoid the execution of those logs which destroy consistency.

Algorithms can be classified with respect to the permitted logs. Algorithms which result in not strictly serial logs (logs in which the steps of a transaction are side by side) are quite sophisticated and always need some preliminary information about the transactions. A typical example of this type is the algorithm of SDD-1 [1,2,7].

To achieve a strictly serial log seems to be a very simple task. What it only needs is to lock the database elements which are to be accessed, before the execution of a transaction. Here we note that the term strictly serial log often means it is strictly serial only with respect to the conflicting transactions (two transactions are said to be conflicting if one of them reads or writes a data element which is to be written by the other one). Some distributed methods have been developed to solve the problem of mutual exclusion, i.e. Agrawala [4], E. Chang [5], but generally the concurrency control algorithms do not use them. The reason is the fact that they increase the number of messages needed to execute a transaction.

Most concurrency control methods are optimistic. Issuing the transaction, they assume that the system does not contain any conflicting transactions. If in spite of this expectation there is a conflict then the algorithm ensures that there is at least one node where the conflicting transaction meet. This meeting results in suspending one of the transactions. This suspension either means waiting for the end of the other one or causes the death of the suspended transaction (which should be rolled back and started again). If sufficiantly great number of nodes (not necessarily all) with data element written by the transaction has been visited then the so called synchronization phase is terminated. The number of nodes required to achieve synchronization depends on the concurrency control algorithm which has been used. We might say, a transaction must visit as many nodes as necessary to meet all the possible conflicting transactions. If concurrency control works correctly then one and only one of the concurrent transactions can finish its synchronization phase. As a result of this concurrency determination method, the nodes should not execute the updates of a transaction until its synchronization phase is terminated. The nodes are informed about the termination of the synchronization by a message. This message is often called confirmation.

There are many methods known from the literature which really result in serial logs i.e. Thomas [8], Rosenkrantz et. al. [6]. They differ in the solution of synchronization, but confirmation is resolved always by messages.

The algorithm described in this paper has a different solution for the realization of confirmation. It uses timing instead of messages. In case of reliable network, in this way, a transaction can be executed with minimal number of messages.

2. ENVIRONMENT CONDITIONS

The conditions our algorithm works among are quite strong, but the methods applied are very simple. Later it will be shown how the conditions can be weakened or even left out while adding new features to the algorithm.

Conditions:

1/ fully duplicated database.

2/ A clock to every node. If the clock in node <u>i</u> shows $C_i(t)$ at the moment <u>t</u> then

$$\forall_{i,j}\ C_i(t)=C_j(t)$$

3/ The transactions must be moment-like. This means that the time of the first read and last write must be at the same instant.

4/ The network, used by the database should be reliable. This means that every node always works correctly and each node can have access to all other nodes.

5/ A time interval τ can be defined in the network so that for every pair of nodes the transmission time of a message from one node to another is always less than τ.

3. THE PRINCIPLE OF THE ALGORITHM

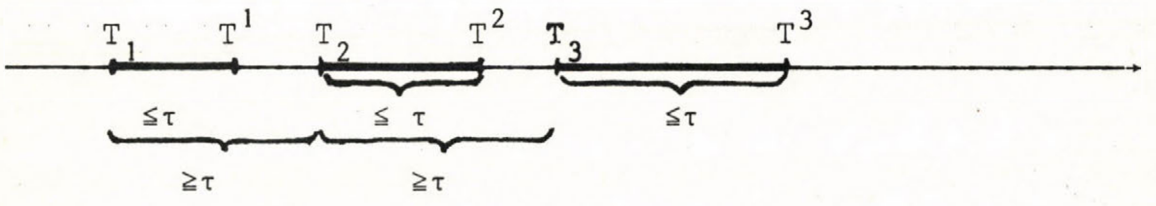
Independently from the concrete concurrency control method, to execute a transaction the minimum of one message per node is necessary. The time of a message exchange depends on the participant nodes, the type of message forwarding (broadcast, daisy-chain), the state of the network etc. There is no concurrency control algorithm which provides minimal execution time for every transaction on any network at any time. An optimal algorithm must be constructed in a way that permits an optimal implementation, that is the execution of a transaction should need at most one message per node and should not constraint the choice of the message forwarding method.

For algorithms which work by locking all the database elements accessed by a transaction, a sufficiant condition to achieve this locking is that the system executes only those transactions which are issued at a moment when no conflicting transaction is under execution. To decide whether there are transactions under execution conflicting with the transaction to be issued, the system has either to wait until the effect of conflicting transactions arrive at the node where the transaction is to be issued or to send inquires about transactions to every node. If we are to achieve an optimal solution then the former possebility is the better choice. The system has to wait before initiating a transaction as long as a message needs, in worst case, to arrive from the farthest node in time at the node where the transaction is to be initiated. This time is not more than τ, defined among the conditions.

An algorithm is to be constructed that matches the above mentioned considerations. This algorithm, in order to execute a transaction

a/ sends only one message to a node
b/ with arbitrary mode of message transferring, while
c/ the time interval between two conflicting transactions is at least τ.

The execution history of the transactions represented on a time axis will look like this:



T_i is the issue and
T^i is the execution time

To fulfil condition a/ the confirmation can not be done by explicit messages but can be done i.e. by timing. This timing is started at every node that receives a synchronization message.

To execute a transaction both of the following conditions must hold:

C1: Every node in the network is noticed of the transaction.

C2: At the time when the transaction is issued there are no conflicting transaction under execution.

If the node where the transaction is initiated, distributes the description of the transaction simultaneously with issue time (T_m) then C1 is fulfilled at the latest $T_m+\tau$. However, at time $T_m+\tau$ condition C2 is also hold because transactions conflicting T_m had to be issued in the interval $T_m-\tau$, T_m thus they arrived at each node till $T_m+\tau$.

To avoid concurrency let the strategy be the aborting and restarting of the transaction issued later.

Each node can check the concurrency independently from others. That is, if a node did not receive any transaction until $T_m+\tau$ that was concurrent with the transaction issued at T_m then conditions, C1, C2 are hold and the transaction of T_m can be executed.

With the terminology used heretofore, for a transaction issued at T_m, the synchronization will be finished at $T_m+\tau$ and the confirmation is the termination of timing τ started at T_m.

4. THE IMPLEMENTATION OF THE ALGORITHM

The issuing node assigns a timestamp (the value of the local clock at the local virtual execution of the transaction) to the transaction. Following this the transaction is compared, in the same way as described for an arbitrary node later, with the outstanding ones what have been received by the node. If the transaction has not been aborted (at the issuing node) then it is distributed in the network (timestamp which is the identifier, the read or written data elements).

The events of the system are: the termination of a timing, and the reception of a transaction.

Node actions taken at events:

A/ receiving the description of a transaction

1/ If the node does not contain any outstanding transaction conflicting with the received one then the received will be outstanding and a $\tau-(T_h-T_m)$ long timing is started. T_h is the local time at receiving the transaction. This timing will be referred to as primary timing.

2/ If the node contains any outstanding transaction conflicting with the received one then

i/ if $T_m < T_w$ (T_w is the issue time of the outstanding transaction)
the transaction timestamped by T_w is aborted and the transaction of T_m will be treated as described in point 1.;

ii/ $T_w + \tau > T_m > T_w$ the transaction of T_m is aborted

iii/ $T_m > T_w + \tau$ the transaction of T_m is treated as described in point 1..

B/ a primary timing is terminated
The transaction indicated in the timing is executed and an auxilary timing with interval τ is initiated.

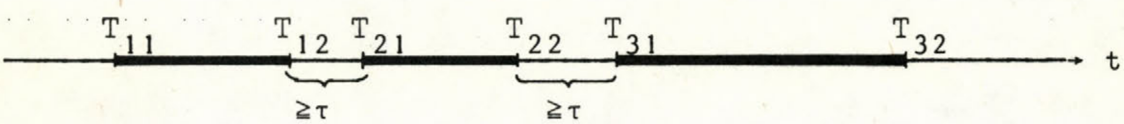
C/ an auxilary timing is terminated
The transaction indicated in the timing from this time onward is not outstanding.

At first sight, auxilary timing seems to be unnecessary. Though not unnecessary, it is not the only solution of the next implementation problem: each node at every time instant T_m must have the ability of checking every confliction it has got enough information for. Two conflicting transactions can meet at a node in such a way that the one issued earlier has been executed when the later arrives. Therefore the transactions, having been executed, must be reserved for a time interval τ.

5. MODIFICATIONS TO LEAVE OUT OR TO WEAKEN THE CONDITIONS

A. Eliminating the condition of moment-like transaction

If the reads and writes of the transactions form a finite notzero interval then the previous time axis representation shows the following picture. For the sake of simplicity the moment of execution is not displayed.



The T_{i1} is the time of the first data access of the transaction i while T_{i2} is the moment when the transaction is finished, that is, from this time the transaction can be distributed.

Theorem 1.

If T_{m2} is regarded as issue time then using T_{m2} instead of T_m the actions to be taken at events remain unchanged supposing that C3 is holding at the issue node.

C3: a transaction m is distributed only if the node did not receive any conflicting transaction during T_{m1}, T_{m2}.

Proof

For every pair of conflicting transactions in the network

1.1 $$T_{i2}-T_{j1}<\tau \rightarrow T_{i2}-T_{j2}<\tau$$

is true. This is because if $T_{i2}-T_{j2}>\tau$ then the node, where j is initiated, received the transaction i until T_{j2}. In this case C3 ensures that transaction j is not distributed. It follows from 1.1 that

1.2 $$T_{i2}-T_{j2}\geq\tau \rightarrow T_{i2}-T_{j1}\geq\tau$$

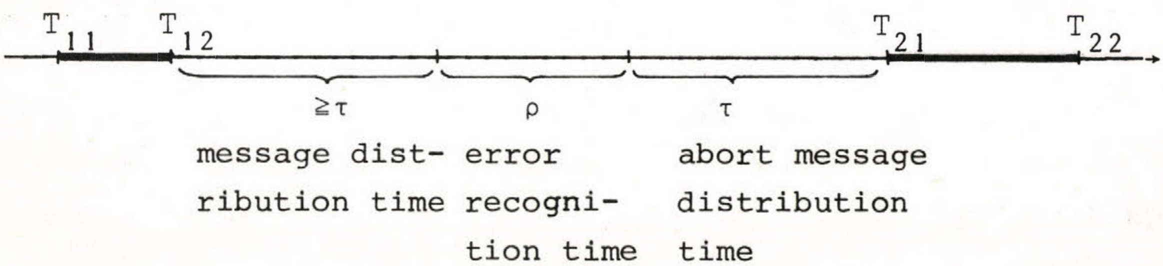
The condition part of this implication is true if the algorithm is used with the substitution $T_m=T_{m2}$.
q.e.d.

B. Eliminating the condition of reliable network

Instead of reliablity each node is expected to notice in ρ time the failure of its message.
Let's complete the algorithm with the following supplement:

S1: if a node regonises that its message (containing the descritption of a transaction) is undelivered then distributes an abort message. A node, having received an abort messages, does not execute the referred transaction.

To ensure a possible abort message to arrive at each node in time, instead of τ, $\tau+\rho+\tau$ should be used everywhere in the algorithm. With this modification the next execution log is obtained:



In spite of introducing abort messages, there may be some nodes (those whose separation from the network happens after receiving the transaction but before its corresponding abort message is obtained) which execute the transaction, causing inconsistency. Since -in case of network failure- the inconsistency is unavoidable, the algorithm must contain methods to recover the database according to the last proper state (the state resulted by the last properly executed transaction).

To solve the problem discussed above let's add the following supplements to the algorithm:

S2: a node interrupts its work until the system is recovered if it receives or generates an abort message.

S3: for every node, a log is maintained containing the identifiers of locally executed transactions.

Suplement S2 ensures every node to suspend its work in case of network failure sooner or later. Only a newly arised network failure can prevent the delivery of an abort message to a node which has got the transaction to be aborted. This means for a node that to the execution of each aborted transaction belongs a network failure. This sequence of failures approaches the node until, in the worst case, the node discoveres the failure itself. The maximum number of transactions a node can execute during the interval of network failure and its recognition can be calculated for each node from the number of nodes and from the topology of the network. Let K denote the largest value of the above counted maximums. Because each failure which prevents the delivery of an abort message separates at least one node from the network, a topology independent upper bound for K is equeal to n-1, where n is the number of nodes in the network.

Supplement S3 is used during database recovery. In an error free network every log contains the same identifiers. Logs diverge, when -because of failure- some nodes start an independent life (they execute transactions that were to be aborted). There is at least one node which detected the first failure of the network. The log and the database of this node are correct and this log will be a common slice of all logs. After system recovery, this log must be searched and on the basis of the database of the same node the whole database can be recovered. The sufficiant length of the logs to be kept at the nodes for the above search is K, while logs are circular lists of transactions in execution order.

To use the algorithm in an unreliable network, the $\tau = 2\tau + \rho$ substitution can't be left out. The further discussion above describe only a possible extension of the algorithm in the field of unreliable networks. Certainly, there are other solutions of this problem and these considerations can be used to solve different problems.

C. Eliminating the condition of fully synchronized clocks

Instead of the hypotetical $\forall_{i,j}\; C_i(t)=C_j(t)$ heretofore the realizable $\forall_{i,j}\; |C_i(t)-C_j(t)|<\varepsilon$ condition will be used.

Because of asynchronous clocks the nodes aren't able to determine the exact time a transaction has spent in the network. This time determination has a twofold role: first to ensure the simultanious execution of a transaction at different nodes secondly, to be the base of concurrency checking. To modify the algorithm, first the clock conditions should be exactly defined.

C4: $\forall_{i,j} |C_i(t)-C_j(t)|<\varepsilon$

C5: $\sum_{i=1}^{n} C_i(t)/n=t+\Delta$

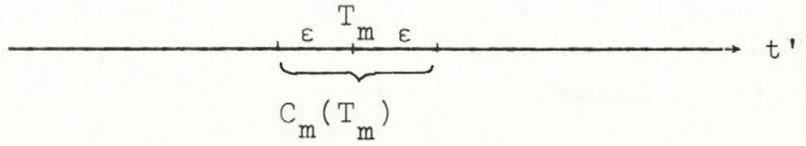
Where Δ is an additional factor to the real physical time and it is constant in the interval of our investigation $(2-3\tau)$.

Although, only time differences have role in the algorithm, for the sake of exactness $t'=t+\Delta$ instead of t will be used.

Theorem 2.

The difference between the real physical issue time (T_m) and the timestamp $C_m(T_m)$ of a transaction is at most ε.

$|C_m(T_m)-T_m|<\varepsilon$



Proof

According to condition C5, the real physical time always lies between the minimum and the maximum of local times. By C4 the differences between local times at a given moment are less than ε. Consequently, the local time at any node has a smaller difference from the physical time than ε.

<u>Theorem 3</u>

The differences between the local execution times of a transaction are not more than ε.

<u>Proof</u>

Let the transaction arrive with timestamp $C_m(T_m)$ at node <u>i</u> at physical time T_i and at node <u>j</u> at physical time T_j. The timing started at node <u>i</u> terminates at:

$$\tau-[C_i(T_i)-C_m(T_m)]+T_i$$

The timing started at node <u>j</u> terminates at:

$$\tau-[C_j(T_j)-C_m(T_m)]+T_j$$

The absolute value of their differences is:

$$|\{\tau-[C_i(T_i)-C_m(T_m)]+T_i\}-\{\tau-[C_j(T_j)-C_m(T_m)]+T_j\}|=$$

$$=|C_i(T_i)-C_j(T_j)+T_j-T_i|=|C_i(T_i)-C_j(T_i+\Delta t)+T_i+\Delta t-T_i|=$$

$$=|C_i(T_i)-C_j(T_i)-C_j(\Delta t)+\not{T}_i+\Delta t-\not{T}_i|<\varepsilon$$

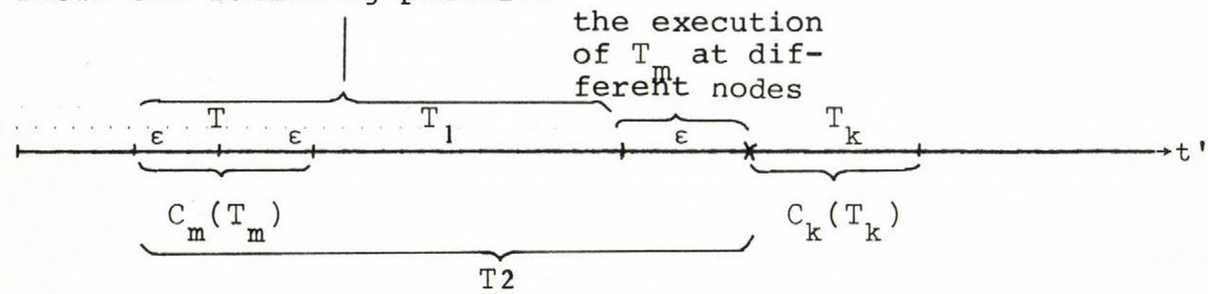
because $|C_i(T_i)-C_j(T_i)|<\varepsilon$ and

for the time interval Δt the local clocks may be regarded as fully synchronized, therefore

$C_j(\Delta t)=\Delta t$

q.e.d.

Using the theorems, the time axis representation of a log shows the following picture:



The time represented by T1 must be long enough to ensure, for every node, the reception of the transaction of the transaction of T_m and the reception of all conflicting transactions timestamped earlier than T_m. If value $\tau+\varepsilon$ is assigned to T1, then the time between the physical issue time (T_m) and the first local execution is at least τ.

In the interval succeeding T1, the database may be inconsistent. Because of this possible inconsistency, the execution of transactions conflicting with the tranaction of T_m is not permitted in this interval. Consequently, the value of T2 should be at least $\tau+2\varepsilon$.

Summing up the modifications to be done in case of real clocks:

a/ The timing should be changed from τ to $\tau+\varepsilon$.

b/ In timestamp comparison $\tau+2\varepsilon$ should be used instead of τ.

D. Weakening the condition of fully duplicated database

Instead of total duplication, the following constraint will be used: the issuing node has to contain the whole read--set of the transaction.

The execution of a transaction can be completed within a node, if the node satisfies the weakened condition with respect to that transaction. That is, a transaction can update its whole write-set on the basis of its read-set. The distribution, similarly to fully duplicated databases, serves only for introducing updates into the local databases. The nodes execute only the part of a transaction write-set that refers to its own database. Unfortunately it is impossible to reduce the number of messages since, to have the same decision results on conflictions, every node must be informed about all messages.

6. CONCLUSIONS

The suggested algorithm has some disadvantages. These are the restricted usage area and the timing that may cause implementation difficulties. On the other hand the advantages of the algorithm are that the execution of a transaction needs minimal number of messages and the amount of auxiliary information (waiting lists or the reservation of timestamps in the database etc.) is rather small.

We have some concluding remarks related to timing. The minimal time, calculated from the paramteres of the network, does not have to be used for the value of τ. With increasing τ, the number of transactions executed in time unit decreases; but on the one hand the timing can be implemented more easily using digital methods and on the other hand the short failures do not result the exceeding of τ that stops the system and needs recovery.

Other demands of the algorithm are usually provided by modern systems. In computers the preciousness of real-time clocks is in the order of 10^{-7}, 10^{-8}. That is, it is possible to hold the clock conditions for a long time without any synchronization even for an ε that is one order less than τ. Moreover, the clocks can be synchronized by messages. To determine the preciousness of synchronization the results of L. Lamport [3] can be used.

ACKNOWLEDGEMENTS

The author would like to thank L. Máté his continuous help during editing of this paper.

R E F E R E N C E S

1. P.A. Bernstein, J.B.Rothnie, N.Gooedman, C.A.Papadimitriou; The Concurrency Control Mechanism of SDD-1: A System for Distributed Databases (the Fully Redundant Case) IEEE Transactions on Software Engineering may 1978 154-167

2. P.A. Bernstein, D.W.: Shipman, J.B. Rothnie; Concurrency Control in a System for Distributed Databases (SDD-1) ACM Transactions on Database Systems march 1980 18-51

3. L.Lamport; Time, Clocks and the Ordering of Events in a Distributed System CACM july 1978 558-565

4. G.Ricart, A.K. Agrawala; An Optimal Algorithm for Mutual Exclusion in Computer Networks CACM january 1981 9-17

5. E. Chang; n-Philosophers: an Exercise in Distributed Control Computer Networks sept 1980 71-76

6. D.I. Rosenkrantz, R.E. Stearns, P.M. Lewis II; System Level Concurrency Control for Distributed Database Systems ACM transactions on Database Systems july 1978 178-198

7. J.B. Rothnie, P.A. Bernstein, et.al.; Introduction to a System for Distributed Databases (SDD-1) ACM Transactions on Database Systems march 1980 1-17

8. R.H. Thomas; A Majority Consensus Approach to Concurrency Control for Multiple Copy Databases ACM Transactions on Database Systems june 1979 180-209

Összefoglalás

EGY ÚJ IDŐZITÉSEN ALAPULÓ ALGORITMUS AZ ELOSZTOTT ADATBÁZISOK KONKURRENS FELÚJÍTÁSAINAK VEZÉRLÉSÉRE

A dolgozat röviden ismerteti a duplikált elosztott adatbázisokon alkalmazott konkurrencia vezérlő algoritmusokat, majd megad egy új eljárást, amely a felújítások szinkronizálására üzenetek helyett időzízést alkalmaz. A módszer optimális az egy felújításhoz tartozó üzenetek számára nézve.

Об одном новом алгоритме в распределенных обработках данных

В настоящей работе занимаемся распределенной обработкой данных. Даем новый алгоритм для синхронизации конкретных процессов. Наш алгоритм оптимально работает относительно мощности пакетов, которые необходимы для перепосылки информации.

MTA Számítástechnikai és Automatizálási Kutató Intézete, Közlemények 27/1982

THROUGHPUT OPTIMIZATION OF MULTISTAGE QUEUEING SYSTEMS WITH FINITE INTERMEDIATE STORAGE *)

ADAM WOLISZ +)

1. Introduction .

Multistage Queueing Systems /MQS/, that is such systems where every demand has to be served consecutively, in a predefined order, by several servers receive recently a lot of attention, being an important tool for modelling several kinds of industrial systems. For example such structure have production lines [6] and computer communication systems [12] .

If service times at different stages are not constant and equal then unavoidably some queues form between consecutive servers. In reality this queues are not allowed to exceed some fixed values because of storage facility constraints. If, upon completion of a service, no place is available in a consecutive buffer for depositing the demand involved, this very demand may perhaps be lost, and abandon the system never to return again.

The paper has been presented at VI th Conference on Operating Systems, hold at Visegrad, Hungary, in February 1980.

Polish Academy of Sciences, Department of Complex Automation Systems
44-100 Gliwice, Zwycięstwa 21, Poland.

More frequently however, no losses are permitted and server which completed the "fatal" service is used as an additional storage place being of course unable to process other demands /the blocking phenomenon/.

In fact different possible types of blocking may occure, like repeating the service of demand which couldn't have been placed in the consecutive buffer, or even forcing this demand to return to the very beginning of the system and have all the so far achieved servic repeated / cf [11] where also some equivalence rules between differe types of blocking where discussed and [31] /.

Numerous papers were concerned with the analysis of MQS.It has to be stressed that exact analytical tools generally fail when the number of consecutive servers exceeds three, and even for smaller systems only some special cases / or some special system features/ are fully investigated.Thus a great affort is being done to obtain approximate solutions either by means of specially developed methods like diffusion approximation /cf [24] /, numerical methods /eg [19] or simulation.

Other, equally important area of research is optimization of MQS operational features, like throughput, servers utilization, queue length e.t.c.For these studies usually the following way was chosen by numerous researchers: First the special cases of two and three stage systems were investigated /preferably analytically/ and afterwards, basing on conclusions obtained there some hypothesis of more general applicability were formulated.These in turn were subjec to verification using- most frequently - simulation as the tool.

The purpose of this paper is to present state of the art in the area of throughput optimization in MQS with the classical type of blocking mentioned above.In author's opinion there is a need for such survey as in a number of papers several optimization rules have been developed /usually each for some special case/, being sometimes non-consistent or even contradictory.

In consecutive sections, after dealing with some general properties including the formal statement of the optimization problem and with the special case of systems having constant service times, outlines for optimal choice of each of the parameters influencing the throughput of MQS with single servers at every stage will be considered in turn. This will be followed by considerations concerning the use of multiservers at some stages, and some remarks about optimization goals other than throughput maximization. The whole paper is completed by a set of final conclusions.

The list of references compiled in this paper, although not aimed as a complete bibligraphy includes, in author's opinion, the vast majority of papers concerned with optimization problems in unpaced MQS /i.e. such where no external synchronization in operation of different stages exists/. Papers dealing with paced systems were mentioned only if the results presented there were in strong connection with the investigated system, while papers covering the problems of system analysis exclusively, have been intentionally omitted.

As production lines are one of the common technical systems being modelled as MQS, it is worth mentioning, that a wide range of both analysis and optimization problems connected with designing of production lines was reviewed by Buxey, Slack and Wild [6] establishing also their connection with topics discussed here.

It is hoped that the unified approach presented here will be of some help in directing the future research, simultaneously providing practicians with a set of directly applicable optimization rules.

2. Concepts and Definitions .

Further in this paper MQS of the type presented in Fig. 1 will be considered.

Identical demands originating from a source W with intensity λ_0 are to be served consecutively on "M" stages /each of them consisting of several, not necesserily similar service facilities/ in a strict order.A queue S_i with N_i places is allowed to build up in front of the i-th service stage.The service time of demands on server A_i^j are independent,identically distributed nonnegative random variables b_i^j, with arbitrary distribution functions $B_i^j(x)$.

Let $E\,(b_i^j)$ denote the mean service time and μ_i^j its reciprocal /service intensity/, $\mu_i^j = \left[E\,(b_i^j)\right]^{-1}$ .Random variables b_i^j and b_l^k are statistically independent if $j \neq k$ or $i \neq l$.

It is assumed, that only one demand can be served by a server at any time.

Each stage is preceded by a buffer. Intermediate buffers S_2 , ... , S_M are of finite size $/N_i < \infty$, $i = 2, 3, \dots M/$, causing the blocking phenomenon to occure.

Each server A_i^j , $j = 1, 2, \dots n_i$; $i = 1, 2, \dots , M-1$ is always in one of three possible stages:

- busy if it is serving a demand;
- blocked, when it has completed a service but cannot pass on the demand to the next stage, because the consecutive buffer S_{i+1} is full;
- idle when it is neither busy nor blocked.

We shall assume that the server A_i^j may be idle if and only if there are no demands waiting for service in the queue S_i.

Let us now introduce some classification of MQS.

A MQS will be called <u>queueing line</u> if every stage consists of one server, exclusively /$n_i = 1$; $i = 1,2, \ldots , M$/ . Then $B_i^1(x)$ will be denoted briefly $B_i(x)$.

If all servers installed at any stage are identical / $B_i^j(x) = B_i^*(x)$; $j = 1, 2, \ldots , n_i$; $i = 1, 2, \ldots M$ / then the MQS is called <u>homogeneous</u>, otherwise it will be referred to as <u>non-homogeneous</u> .

If the joint service intensity of all servers installed at stage "i" , i= 1, 2, ... , M is constant,

$$\sum_{j=1}^{n_i} \mu_i^j = D , \qquad i = 1,2, \ldots M; \tag{1}$$

then the system is called <u>balanced</u>, otherwise it is <u>unbalanced</u> .

In order to preserve a measure of system unbalancing, we shall further assume that the following holds:

$$\sum_{i=1}^{M} \left[\left(\sum_{j=1}^{n_i} \mu_i^j \right) \right]^{-1} = M \tag{2}$$

Thus for the balanced case

$$\sum_{j=1}^{n_i} \mu_i^j = 1 . \tag{3}$$

Naturally for homogeneous, balanced systems;

$$\begin{aligned} \mu_i^j &= \mu_i^* & j &= 1,2, \ldots , n_i \\ \mu_i^* \cdot n_i &= 1 & i &= 1,2, \ldots , M \end{aligned} \tag{4}$$

Comparing the service time distribution effect on the system throughput we shall frequently utilize the variability coefficient, defined for a service time distribution B (x) as

$$C^2 = \frac{\int_0^\infty x^2 dB(x)}{\left[\int_0^\infty x dB(x) \right]^2} - 1, \tag{5}$$

and being a suitable measure for the variability of service time. Naturally $\zeta = 0$ for the constant service time and $\zeta = 1$ for exponentially distributed service time.

In some queueing lines the transfer of demands from all servers to consecutive buffers takes place simultaneously, being externally synchronized, no matter if all service processes where completed or not. This occurs for example in automated transfer / moving belt/ production lines. Such queueing lines will be called paced .

Thus in the paced systems a predetermined time quantum, /called cycle/ is imposed for every service. In the case of constant service times the cycle should be equal to the longest service time. If service times are variable then the line should be designed so as to minimize the probability of one or more stations exceeding the cycle.

Methods for designing paced queueing lines were surveyed in [6], while the case of variable service times is treated for example in [43] .

Further in this paper we shall be interested only in the cases when no external synchronization / no pacing effect/ in the flow of demands through the system is introduced, called unpaced systems.

An important feature of any queueing system is its throughput /frequently called production rate/ defined as the mean number of demands leaving the system in a time unit /completly served/.

We shall be specially interested in the maximal throughput / capacity/ which a given system may achieve.

The MQS is called open if the buffer S_1 preceding the first service stage is unlimited, $/N_1 = \infty /$, and the input stream of demand is a renewal stream.

On the other hand if the input stream is such, that the queue S_1 is never empty, then the system is called saturated. Certainly for saturated systems both the maximal queue size N_1 and the detailed characteristics of the input stream are of no importance.

Two MQS : an open one and a saturated one are called corresponding if they are identical, up to the specification of the input stream.

Let us consider some MQS. If we define another system in which the order of service is reversed, that means every demand passes through the system beginning with stage M and ending at stage 1, buffer sizes being exactly preserved, then such two systems are called dual systems .

Finally we shall introduce the concept of saturated system accumulation , being equal to the maximal number of demands which are allowed in the system simultaneously.
System accumulation V is given by the following formula

$$L = \sum_{i=1}^{M} n_i + Z , \qquad (6)$$

where

$$Z = \sum_{i=2}^{M} N_i$$

is the total number of storage places /total buffer size/ available in the system.

It is evident that dual systems have equal accumulation.

3. General considerations .

In this section we shall present some theorems and remarks concerning some large classes of MQS.

Lavenberg [28] proved / as a special case of more general dependences discussed in his paper/ an important connection between the throughputs of corresponding open and saturated multistage syste If only all service stations in a MQS have Coxian service time distr butions*) and all intermediate buffers are finite, then throughput of the saturated system T_s is equal to the maximal throughput /capacity/ V of a corresponding open system.

More precisely, the throughput T of the open system can be defined as follows:

$$T = \begin{cases} \lambda_0 & \text{if} \quad \lambda_0 < T_s \\ T_s & \text{if} \quad \lambda_0 > T_s \end{cases}$$

where λ_0 denotes /cf Fig.1/ the intensity of demands arrival in the open system. Thus $V = T_s$.

Additionally in the case when $\lambda > T_s$, the stationary distribution o the number of demands waiting in the queue S_1 doesn't exist, contr ry to the case when $\lambda < T_s$.

Thus in order to find the capacity of multistage queueing syste it is necessary to investigate the proper saturated case.

---

*) Coxian distribution means any distribution having a rational Laplace transform of the distribution function.In fact it is possible to approximate any distribution function fairly well with a Coxian distribution, thus the constraints are not restrictive.The case of constant service times, however also possible to approach as a limiting case of Erlangian distribution, will be further in this paper traated seperately.

Let us notice that without loss of generality it is enough to consider MQS without intermediate buffers. ([2], [31]).
It is quite evident, that any buffer of size N can be replaced by N servers, each having a null service time / $B(x) = \mathbb{1}(x)$, $\mathbb{1}(x)$ being the Heaviside function/ .This property simplifies many proofs.

It is also worth stressing, that the capacity of a MQS is almost independent of the service disciplines applied.In fact, as long as no server A_i^j , $j = 1, 2, \dots , n_i$; $i = 1, 2, \dots M$ may remain idle if the queue S_i is not empty, and the service is of nonpremptive type, all queueing disciplines / possibly not identical at different stages/ yield equal system capacity.

Another important result of quite general applicability is the so called reversibility property for saturated MQS with finite intermediate buffers.
Yamazaki and Sakasegawa [54] demonstrated that the capacity of a queueing line with general service time distribution is invariant for reversal ordering of the servers.That means, dual systems have equal capacity.

The reversibility property is important for the throughput optimization considerations as it makes us expect optimization rules calling for queueing line structures being in some sense "symmetric". This feature will become more meaningfull in further sections.

Recently independent proofs of the reversibility property were given in [13] and [32] .
Kawashima [22] generalized this property, stating that also both the distributions of service completion times for every customer and the number of service completions in the time interval /0 , t / are invariant for reordering the servers reversely.

Yamazaki, Sakasegawa and Kawashima [55] proved, that the reversibility property holds also for the case when, at some stages, there are installed homogeneous multiservers, having however constant service times /notice that this is not a queueing line any more/.
Wolisz [52] verified, that this property holds also for two-stage homogeneous systems with arbitrary numbers of exponential servers at each stage.

The occurance of blocking phenomenon causes a decrease of system throughput in comparison with systems without such effect.
In fact Muth [31] pointed out, that the capacity V_L of any queueing line with finite intermediate buffers has two bounds:

- The upper bound V_L^+

$$V_L^+ = \min_{i=1,2,\dots M} \mu_i \quad , \qquad (7)$$

being equal to the service intensity of the slowest server.
The throughput of a queueing line would tend to this value if all queues were allowed to build up without any restrictions.
This is a direct result from Sachs [44] theorems concerning the ergodicity of tandem queueing systems.

- The lower bound V_L^-

$$V_L^- = \frac{1}{E[\max(b_1, b_2, \dots b_M)]} \qquad (8)$$

It is easy to see that $V_L^- = V_L^+ = V_L$ in the case of constant service times.

In his paper Muth compared also the values of the difference $V_L^+ - V_L^-$ for various queueing lines.

The capacity of a MQS may be, generally speaking, influenced by the number of stages as well as following parameters describing individual stages:

-service intensity,

-service time distribution,

-buffer size allocation to the stages,

-servers reliability,

-number of servers installed at the stage,

-homogenity of servers installed at one stage.

Notice the common assumption, that a server may break-down only when service is in progress. Thus assuming that the preempted by some breakdown service is resumed after repairing of the proper server we can under some simple additional assumptions, treat the breakdown process together with the service process, describing them jointly with a modified service duration distribution function. Thus if we shall further assume, that some service time distributions are identical that will mean identicity of both the real service and break-down processes.

A MQS is defined by specifying the parameters of all stages which are - generally speaking - entirely different for individual stages. Let us point out that two main types of optimization problems are usually formulated:

a/ the improvement problem

Given an M stage system, increase its capacity through modifying the parameters of some stages.

Usually it is demanded to point out which possible modification / in the context of real process being modelled/ would be most profitable in terms of throughput increase.

b/ the rearrangement problem

Given an M stage system arrange all the facilities so as to maximize its throughput. This problem can be solved through one of the following actions:

- different dividing of processing among stages, thus influencing the mean service times of the stages involved / like for example in the case of two-stand rolling mills/,
- changing the sequence of stages / this is permitted in some processes like for example equipment maintenance, testing or tuning/,
- different allocation of buffers to individual stages with regard to the fixed total buffer size Z.

It was demonstrated / for example using approximate calculations in [19] and simulation in [41]/that MQS capacity decreases generally with the increase of number of stages. Further an attempt is done to present outlines for optimization in both of the above precised contexts with respect to parameters of individual stages.

4. MQS with constant service times .

We shall start our considerations on throughput optimization in MQS with the special type of systems, having constant service times.

Such systems have been considered in papers [9] , [16] , [27] , [31] , [49] .

As the Lavenberg theorem mentioned in the previous section does not directly apply to this case, the open systems have to be considered.

It has been proved, that for any open system of homogeneous type with i-th stage consisting of n_i parallel channels, each having the same constant service time $b_i^* = c_i$, neither the sujourn time of demands nor the stream of demands leaving the system depend on the sequence of stages and the capacity of intermediate queues. This result remains valid also for arbitrary input streams, and in the case of intermediate queues allowed to build up to infinity [49] .

Friedman [16] has introduced the concept of dominance, assuming that stage k dominates stage p if no demand ever waits at stage p if it is preceded by stage k .
He proved also that such property holds if and only if $c_p \leqslant K c_k$, where $K = [n_p/n_k]$ /greatest integer notation/.
For example when one stage dominates all other, then the only waiting in the system occurs before this very stage, which can be easily demonstrated through proper rearrangement.
Using this concept Friedman suggested, while Suzuki and Kawashima [49] developed further, a method for reducing a multi-stage system to an equivalent system with smaller number of stages, sometimes even to one stage succeded by a $\bullet/D/\infty$ system. As the stability condition of the equivalent G/D/n system can be easily established, it is possible in such special cases, to compare using the presented above methodology, the changes of the original system's capacity for different parameters of individual stages, eventually chosing the best one. This method is however of strongly restricted use.

Generally valid results may be found on this basis only for queueing lines. Muth [31] pointed out, that the capacity of such line T is given by (7), (8)

$$T = [\max(c_1, c_2, \cdots c_M)]^{-1} \qquad (9)$$

Evidently the maximal capacity is achieved, independently from the buffer size, for the balanced case.

The problem of unreliable servers leads in fact to consideration of systems having variable service times, and as such will be discussed in further sections.

5. Unbalancing of queueing lines .

Let us assume a M stage queueing line with service time distributions $B_i(x)$, i = 1, 2, ... ,M.

According to (2) we shall assume that

$$\sum_{i=1}^{M} \frac{1}{\mu_i} = M \qquad (10)$$

where $\mu_i = \left[\int_0^\infty x \, d B_i(x) \right]^{-1}$

a/ The case of identical service time distributions.

We shall temporarily constrain ourselves to the case of identical /up to the mean value/ service time distributions.Thus the only parameter which will be changed are the service time intensities μ_i , however with respect to (10) .Also the size of intermediate buffers is assumed to be fixed and equal for all stages: $N_i = N$, i = 2, 3,...,M.

For the case of two-stage systems, the reversibility property leads to a conclusion that the balanced case is optimal.

Fig. 2 presents system capacity versus its unbalancing in the case of exponential service time distributions / data taken from [18] /. Notice that losses due to unbalancing increase rapidly with the increase of intermediate buffer size N.Thus proper system balancing becomes more crucial for bigger values of N.

Hillier and Boling [18] investigated also the case of M = 3, 4 ; again for exponential service time distributions.They proved that the balanced case is not optimal any longer, and suggested the existence of so called "bowl phenomenon", asking for higher service intensity being assigned to middle stations.

The optimal solution is "symmetric" / $E(b_1) = E(b_3)$ for M=3 ; $E(b_1) = E(b_4)$, $E(b_2) = E(b_3)$ for M=4 / as it should have been expected due to the reversibility property.Proper unbalancing not necesserily very precise, leads to some gain, while inproper unbalancing leads to significant losses. Some remarkable data are given in Table 1.

It is worth noticing that applying $E(b_2) = 1.15$ for the case M=3, N=0; leads to the capacity equal to 98.7% of the balanced case.

As previously, the system sensitivity to inproper balancing increases with the increase of N.

Patterson [35] analysed three stage systems by means of numerical methods obtaining similar results. He suggested however, that for $M > 3$ the optimal arrangement should be such, where the quick stations separate the slow ones. He suggested also, that the gain obtained from such a procedure should decrease with the decrease of service time variability.

A simulational study of queueing lines with M = 3, 4, 12; $N \geq 0$ was presented by El-Rayah [41] who compared the above given strategies with the balanced case. In fact he tested also a third strategy of assigning to the consecutive stations low- medium- high service times, respectively. Such strategy was suggested by Davis [14], for a system with losses /without blocking/, and it was improbable that it will be adequate for the considered case. This strategy is however also suggested sometimes as reasonable.

Experiments with exponential service time fully confirmed the "bowl phenomenon" hypothesis. Furthermore, also for normal *) and lognormal service time distributions the bowl-type arrangement of mean service times was found to be the best one, leading always to improvement over the balanced case.

As for the case M=12 one could suggest various possibilities of defining the bowl - type arrangement, some of them have been tested.

The arrangement with two middle stages having the smallest service time, which increased stepwise with equal quantum up to the longest values /symmetrically/ on the ends, was found better than those where "bowl" was formed from groups of 3-4 servers having equal service intensity in every group. This arrangement applied for the exponential service time distributions leads to the solution given in Table 2 .

The bowl-type unbalancing is always efficient. It was demonstrate that both the possible gain from proper unbalancing, and the imbalance itself, increase with : greater variability in operation times, smaller interstage queueing capacity and larger number of stages in the line. The gain in capacity, possible to achieve due to unbalancing is never high. The possible gain for M=12 /which perhaps could be a little bit improved chosing non-identical quanta while unbalancing the system/ is only marginally higher then that for M=4.

The unbalancing method suggested by Patterson is very unreliable, and leads frequently to system capacity lower then the balanced case.

The low-medium-high arrangement was definitely found to fail generally, and was significantly worse /as a rule/ than the balanced case.

The study of El-Rayah, based on solid statistical methods, shows however how one should be cautious in assessing the results of simulation. Mean values of capacity for dual systems were usually different but the difference was assessed to be statistically insignificant - a correct result in view of the reversibility property. Curiously enough for the three stage system /p.66 of [41] / the author states "It was also verified that a low-medium-high arrangement is superior to a high-medium-low arrangement in terms of expected output rate" which is obviously wrong!.

---

*) Naturally here and furtheron the trunctated normal distribution is considered, (p 15).

b/ The case of different service time distributions.

Rao [39] presented a methodology for establishing the capacity of two-stage lines with exponential service time distribution at one of the stages, and general service time distribution at the other stage, calling for solution of a system of linear equations which order was dependent on the buffer size N.

Calculations for Erlang and normal distribution with variability coefficient $C < 1$ lead to the conclusion that system capacity is optimized if the server having greater variability is assigned slightly higher speed. Both the gain and optimal imbalance increase with increase of the difference in variability coefficients and decrease significantly for larger values of N.

Similar systems have been considered by Wolisz [52] where closed form expressions for system capacity have been given. The earlier results for $C < 1$ have been confirmed, but surprisingly quite different observations appeared for $C > 1$, investigated with second - order hyperexponential distribution. The optimal capacity was in this case obtained for slightly quicker exponential server, which however this time was the one having smaller variability coefficient. Also the increase of N led initially to the increase of gain /and unbalancing/, and only further increase of N reduced this effects.
Sample results are plotted in Fig 3.

Typical optimal unbalancing in the two-stage line lies in the range $E(b_1) = 0.92 - 0.96$ leading to a gain of 0.2 - 0.3 % over the balanced case.

Rao [40] investigated also analytically the case of M=3 without intermediate buffers, assuming all possible combinations of exponential and deterministic service time distributions.

Table 3 contains the results of optimal unbalancing /notice the existence of dual systems/. Rao introduced the term "variability imbalance" calling for assigning shorter service times to the more variable stations.The "variability imbalance" effect may either coincide /eg pattern f/ or contradict /cf pattern c/ with the "bowl phenomenon".

It was suggested that the strength of the variability imbalance effect depends on the difference in variabilities.Rao verified that if a server with uniform service time distribution and variability coefficient equal to 0.5 is located between two exponential servers then the balanced case becomes optimal.

Concluding we can establish the existence of both the "bowl phenomenon" and the "variability imbalance",effect of the later being clear for $C < 1$,while the case of $C > 1$ needs further research. Joint effect of the two phenomena given above can be significant /cf the 6.79% gain in patterns d,e,f of Table 3/.

6. <u>The effect of service time distribution on system capacity</u> .

The irregularity of service time is caused by two main reasons:

- the inavoidable stochastic differences among demands as well as stochastic disturbances in the service process.Those lead usually to small changes of service time - lying in the range of variabilit coefficients less than unity / or more frequently less than 0.5, as for example a value of $C = 0.27$ was found typical for the production lines by Slack [47] .Values of C approaching one are reported in some data transmission applications/.
- break-down of the server.Such situations occure rarely, but it takes usually a mean repair period several times longer than mean service time to resume server's operation.Thus the resulting joint service time has in this case a high variability.

The effects of those two reasons for service irregularity are usually studied seperately : either perfectly reliable servers are assumed or the service times are assumed to be constant, while in random periods break-downs of random duration are assumed.

In queueing systems for the sake of simplicity, there is a strong trend to characterize the random variables involved, only with two first stochastic moments.Let us first assess what error does such attitude introduce in the case of MQS.

a/ The effect of ignoring service time distribution higher moments.

Fig 4, based on data from [38] compares system capacity for M=2,N=0 versus C for different /Erlang and normal/ service time distributions at both stages, suggesting that the differences are of quantitative type only, and increase with the increase of C . Rao [39] demonstrated, that this effect becomes even more visible when quite different types of distributions are compared, like Erlang and uniform distributions with identical values of C .He provided also examples that if, in a two-stage system one of the stages has some fixed distribution, then with the increase of its variability coefficient the influence of higher moments at the other stage will be greater.This influence becomes stronger in the case of small intermediate buffer /Table 4/.

Anderson and Moodie [1] used in a simulation study of multi-stage balanced lines aiming at optimization of buffer size from the point of view of some complex criterion /cf section 9/ both the exponential and normal service time distributions, finding the results qualitatively identical and quantitatively similar, however different.

Such conclusions were also presented by El Rayah [41] who used both normal and lognormal distributions in his simulation studies of unbalanced queueing lines with M=3,4.

Thus we conclude, that for $C < 1$ no example of qualitative difference in optimization rules due to the higher moments are known, while conditions when the quantitative differences can be expected to be small where listed above. In practical cases, for simulational studies where the choice of distribution functions is unrestricted, some afford is made to preserve the shape of this function, using for example positively skewed Weibull distributions, eg [10] , [47] .

b/ The influence of service time variability on system capacity.

From Fig. 2,3,4 and Tables 3,4 it is evident, that system capacity decreases with the increase of service time variability coefficient at any stage. This loss becomes however significantly smaller when larger buffers are applied. Similar conclusion was achieved by Barten [24] for a simulation study of six-stage systems with normal service time distributions, who stressed specially the beneficial role of buffers in canceling the bad influence of service time variability. Data supporting this property can be found also in a simulation study with Weibull type distributions [10] .

On the other hand it has been observed that a similar effect is visible in queueing lines with constant service times and unreliable servers. Studies of such systems where reported [7],[33] ,[34] for paced queueing lines. Buzacott [9] demonstrated however their direct applicability for the unpaced case as well. He investigated analytically a balanced two-stage system with $N \geqslant 0$ and exponential service times, in which stochastic breakdowns of random duration occured.

It was pointed out that the decrease of system capacity in comparison with the fully reliable - constant service time case, can be very precisely approximated by a sum of losses due to either service time variability itself or nonperfect reliability.

Such superposition was expected to hold also for larger systems.

/ Arrangement of balanced queueing lines due to service time variability.

Some data on a balaneed three-stage queueing line with different rdering of servers having unequal variability coefficients are given s a result of a simulation study by Smith and Brumbaugh [46] . hey suggested that allocation of a station with highest variability n the middle of the line led to the worst results,as everage calcu-ated over different buffer allocation patterns.The data presented n Table 1 of [46]/in terms of means only/ violate however signifi-antly the reversibility property, thus it is not possible to draw sing them any more detailed conclusions.

Systems with M=4,10 and different values of N have been simulated y Carnall and Wild [10] .Service times where assumed to be either onstant or variable, described by the positively skewed Weibull type istribution.In every experiment it was essumed that variable stations re identical.

It was found that for M=4, and two variable stations significantly igher capacity has a system with variable stations located at the nds of the line, in comparison with the case of their location at the iddle. The gain from such strategy increases with the increase of ariability coefficient and decreases with the increase of buffer sizeN. or example in the case $C = 0.5$, M=1 the gain approaches 4% .

Experiments with M=10 lead also to a conclusion that locating ariable stations at the ends is justified, resulting in a 1.33% gain ver a random sequencing of stages, and 3% gain over allocation of ariable stages in the middle of the line.

Thus the authors suggested the existence of something like the bowl phenomenon" concerning the service time variability for balanced ines.

An extensive simulation study with normally distributed service times and no intermediate storage was reported by El-Rayah [42] . For M=3 the bad effect of allocating highest variability to the middle station was confirmed. Also for M=4 the above given suggestions were verified to be true. Further the author demonstrated that this queueing line with equal mean service times and "unbalanced" variability coefficients /the sum of them over all servers being constant/ may yield higher capacity then a line with identical servers. It was demonstrated for this case, and also for M=12 that something like the "bowl phenomenon" exists also for the service time variability. In the investigated range of $C < 0.3$ it was found that increasing of some servers variability coefficient, and decreasing its mean service time yield highly similar results. For example if 4 out of 12 servers had $C = 0.15$, while other had $C = 0.3$ then locating the servwrs with smaller variability in the middle of the line, instead of locating them / in one group/ either at the beginning or at the end of the system /both of these cases being equivalent/ led to an improvement in capacity of over 2.5%.

7. The effect of intermediate buffers on system capacity .

In previous sections it was several times mentioned, that the intermediate buffers may increase or decrease the above discussed effects. Now we shall discuss directly the influence of intermediate buffers on system capacity.

In section 3 it was mentioned that intermediate buffers eliminate / to some extent/ the blocking and idleness of individual stages. Thus it is clear that both the preceding and consecutive buffer size influence the operation of any stage.

'urthermore as it yields from (7) that for the unlimited intermediate ›uffers always the balanced system yields the maximal throughput, we :onclude /cf [7] ,[31] / that buffers cannot reduce that portion of ›roduction line inefficiency which is caused by unequal mean service :imes at different stations.

Buzacott [8] points out, that if due to long term imbalance ›etween stages a buffer is permanently full / or permanently empty/ .t is serving no usefull purpose.Thus the magnitude of queue length 'ariations may serve as a measure of the buffer effectivness. 'hese remarks are consistent with results cited in section 5, where 'or large enough buffer sizes the balanced case was demonstrated to ›e optimal.

Certainly as it is visible from the previous section, buffers may ›ignificantly decrease the bad results of service time irregularities.

Using the upper bound V_L^+ given by (7) , one can suggest, that :he ratio V_L / V_L^+ is some measure of buffer efficiency.Let us notice :hat in all the figures and tables presented so far $V_L^+ = 1$.

Fig. 2 gives a good example of buffer size N influence on system :apacity.This influence can be presented in a simple, analytical form. Iunt [20] found that for a two-stage balanced system with exponential ›ervice times and $E(b_i) = 1;\ i=1,2$ system capacity can be expressed ›y a simple formula

$$V = \frac{N+2}{N+3} . \qquad (11)$$

'hus adding an additional waiting place to a buffer of size N leads :o a relative gain of E,

$$E = \frac{1}{N^2+6N+8} . \qquad (12)$$

'alues of both V and E for different N are printed in Table 5.

It is evident that increasing the buffer size by one leads always to ›ome gain being however significant for small N and only marginal for

Thus Hunt concluded that using buffers of size larger than 5 doesn't, generally, pay.

Similar conclusions can be drown for unbalanced two-stage queueing lines with nonexponential servers from Fig 3, and for three-stage exponential lines from Table 1.

Barten [4] simulated systems with normal service time distributions, M=4,6,10 and different values of variability coefficient, demonstrating that the above given remarks remain generally true. This was also confirmed by Slack and Wild [48] for M=5,10,15.

Evidently, however, in order to achieve some predetermined system capacity, the buffer sizes applied should be larger, if service time variability increases /cf Table 4/.

The situation changes significantly when servers break-downs are included into consideration.

In this case, as pointed out by Buzacott [7], the minimal size of buffer should be equal to the mean number of services completed during the mean repair time, while 2-3 times larger buffers seem to be the reasonable choice. The typical buffer size would be rather 30-50 this time.

The problem of allocating the proper joint buffer size Z to a queueing line and dividing it inbetween different stages obtained a lot of attention, eg. [15], [17], [23], [25], [26], [29] . Unfortunately enough the majority of research was devoted to paced queueing lines and thus the obtained results are not of direct use in our case. Let us present an example of the differences.

For paced lines it is stressed / [15], [29] / that the unreliable stages /in other words : stages with high variability/ located at the end of the line ask for more significant increase of buffer size in order to achieve proper system capacity, than identical stages positioned in front of the line. This is obviously in disagreement with the reversibility property. Also the suggestion / [15] / that the division of total buffer size Z among different locations in the line doesn't depend on the value of Z is not applicable for the unpaced case.

An extensive study of buffer allocation problems has been presented by El-Rayah [42]. The case of balanced queueing lines with M=3,4 has been simulated, assuming identical variability coefficients of all servers. The objective of experiment was to verify, should the intermediate queues have identical maximal sizes, or should they be unequal, preserving during the experiment the condition

$$\sum_{i=2}^{M} N_i = Z = const.$$

It was verified, that for $Z \leqslant (M-1)4$ indeed equal buffers yield the maximal system capacity, as what could have been anticipated - every attempt to decrease any of the buffers significantly handicappes the stages involved.

For larger values of Z it occured that the equal assignment leads to almost optimal results; sometimes only assigning larger buffers to middle stages yields some - almost negligible - gain.

Thus it occurs that increasing the buffer size at some stations leads to similar, but considerably smaller, effect as increasing the servers speed.

This effect being small enough does hardly lead to "unbalancing" queueing lines with identical servers in the sense of buffer capacities, it may however be quite valuable when different servers are involved.

Smith and Brumbaugh [46] concluded, that service time variability should be considered when allocating buffer sizes. They stated that relatively greater buffers should be allocated around more variable stages, while possible benefits from proper allocation are smaller than losses incurred if the arrangement is improper. Departures from equal buffer allocation were found to have grater impact when the total buffer size Z was small.

8. The effect of introducing parallel servers .

In sections 5-7 only the case of queueing lines was discussed. Let us now consider the effect of introducing multiservers at some stages.

Wild and Slack [51] in a simulation study compared the operation of two queueing lines with the case of a MQS having two servers at each stage and the same buffer size per server. It was found that the second system is always more effective, the gain being high in the case of large number of stages, low buffer size, and high service time variability, and comparatively lower otherwise.

A systematic analytical study of homogeneous two-stage queueing systems with exponential multiservers at both stages was published by Wolisz [53] .

a/ Two-stage systems : the balanced case.

For the balanced case /as defined by equality (3) / the influence of buffer size N and number of servers $n_1 = n_2 = K$ on system capacity was investigated. As presented in Fig. 5 the influence of intermediate buffer size is similar as in the queueing lines. It is worth stressing, that system capacity increases with the increase of the parallel servers number K, as the additional servers act also as additional buffers. Thus in the case when the buffer capacity is strongly limited, applying of many slow servers at every stage pays better then using a few quick ones.

Also another experiment was reported there. The total system accumulation L was assumed to be constant, it could however be differently devided between servers and buffer with respect to an equality

$$2K + N = L .$$

Results for $L \leq 10$ are plotted in Fig.6 showing that it is evidently better to have as large buffer as possible.Furthermore a system with larger accumulation can have a lower capacity than another system having smaller accumulation, but devided so as to favor buffers size on expence of the number of servers / cases K=4,N=2 and K=2,N=4 compared, may serve as example/.

Thus the following rule should be applied:

If possible a queueing line with large buffers should be used.

If there is no possibility of intrøducing buffers / or their size is severly limited/ then the use of slow multiservers instead of quick single-servers results in capacity increase.

Fig 7 demonstrates that systems having the equal number of servers at both stages /buffer size being constant/ are always most efficient.

A comparison of a MQS and several queueing lines as presented in Fig 8 was done resulting in an experiment similar to that reported in [51] .The results are presented in Table 6.

The conclusions of[51] have been fully supported.Thus the system from Fig 8b was always better

b/ Two-stage systems - the unbalanced case

Let us denote for the sake of simplicity

$$\mu_I = n_1 \cdot \mu_1^* , \quad \mu_{II} = n_2 \cdot \mu_2^*$$

According to (2) following equation is to be respected

$$\frac{1}{\mu_I} + \frac{1}{\mu_{II}} = 2$$

System capacity for $n_1 = n_2 = K$, $N = 0$ is plotted in fig 9, showing that in this case the balanced system is always optimal.Losses due to unbalancing increase with the increase of K.

If however the number of servers at both stages is unequal, then proper unbalancing i.e. assigning higher service intensity to the stage having larger number of servers leads to increase of system capacity.

Examples advocating this statement are plotted in Fig 10, where a constant difference in numbers of servers $n_1 - n_2 = 2$ was assumed.

Both the gain over balanced case and optimal unbalancing decrease with increase of either buffer size N or system accumulation L, and increase with increase of the difference $n_1 - n_2$.

However it was demonstrated that although for $n_1 \neq n_2$ an optimally unbalanced system has higher capacity than a balanced one, it is worse than a balanced case where the identical total number of servers $n_1 + n_2$ would be equally devided between the stages. The loss is higher for small $n_1 + n_2$ and small N.

Thus to optimize the capacity of such systems one should try to apply equal numbers of servers at both stages and balance the system, only if this is impossible, the loss resulting from the unequal numbers of servers may be minimized by proper unbalancing.

9. Remarks on other optimization criteria .

Throughout this paper we were concerned with a unique optimization goal : system capacity maximization. In this section we shall mention briefly other characteristics which are also frequently considered as important design factors.

The most commonly used characteristics are:

- The expected number of demands in the system
- The idle time of individual stages
- The mean in- system time /sujourn time/ of demands.

It should be mentioned that those characteristics are of significant interest both for open and saturated systems, as well as for systems with infinite intermediate queues.

It has been proved / for example [42] ,[46] / that optimal system parameters chosen for different of this characteristics do not coincide. Also the sensitivity of this factors on parameter changes is quite different.

For example for e queueing line with $M=3, Z=8$ changing the buffer allocation pattern from $N_2=3$, $N_3=5$ to $N_2=5$, $N_3=3$ increases the expected number of units in system from 6.3 up to 7.6 not influencing the system capacity at all - systems being dual [42] .

The buffer allocation from small to large along the line was advocated, as decreasing significantly the expected number of units in system, with only marginal loss of capacity for larger M as well.

Optimal sequencing of open two-stage queueing lines minimizing the delay was studied analytically by Tembe and Wolff [50] for infinite, and by Kawashima [22] for finite intermediate queues.

Simulation studies of queueing lines with infinite intermediate queues together with optimization outlines were reported for $M=2$ in [30] and [45] while lines with $M=4$ and $M=20$ were studied in [21][30] respectively.

Complex optimization criteria are often used for determining the proper buffer size [5], [1], [23] .

It is suggested that buffer size should be chosen with regard to following factors:

- costs of servers idle time
- costs of decreasing systems throughput due to limited storage
- costs of holding the inventory of demands in system
- cost of storage space

Closer discussion of these problems exceeds however the scope of this paper.

10. Conclusions .

In view of the statements presented above, some general conclusions can be made.

Let us introduce informally the concept of individual stage efficiency described as following: if the efficiency of any stage increases, while other stages remain unchanged, then the capacity of the whole system will be increased. We shall also assume that the efficiency of identical stages is equal.

The reviewed in this paper results demonstrated, that the efficiency of individual stages may be increased by:

- decreasing service time / i.e. increasing service intensity/,
- reducing service variability ,
- improving servers reliability,
- increasing the size of buffer belonging to this stage ,
- replacing a single server with a multiserver with identical service intensity.

Generally queueing line achieve the maximal throughput, when the stages efficiency is unequal, most efficient stages being located in the middle of the line, and less efficient being gradually moved in the direction of its ends. This can be, essentially, obtained by changing any of the above listed parameters, however with quite different sensitivity. Some illustrative data were given in previous chapters.

It should be strongly stressed, that the value of results review here lies by far more in their qualitative than quantitative part. In practice neither obtaining strictly equal nor optimally unequal stage parameters is possible with high precision.

The most important thing is to know, in which direction should the unevoidable discrepences from the ideal case be permitted, in order t cause serious losses.

Notice that in all the optimization outlines, as a rule the possible gain from observing the optimization pattern was significantly smaller than the loss caused by a wrong decision , although sometimes the gain itself was also worth approaching .
In this sense the exact knowledge of service time distribution is not essential, as long as their variability coefficient does not exceed unity. All the results remain qualitatively similar in this case, and quantitative discrepences are, usually, small.

The case of $C > 1$ received very few attention so far. This occurs if the breakdown process is treated jointly with the service Also in the case if at some stages one of several non-identical servers is used alternatively for a given demand / because of its special, individual features/ such situation occurs, leading for example to hyperexponential service time distribution.
Such situations are not uncommon while modelling production systems. As some irregularities have been noticed /cf. section 5/ for $C > 1$ this area needs further research.

Similarly further research is needed in order to verify to what extent the results of introducing parallel servers discussed in section 8 may be generalized for the system with larger number of stages.

It is essential while using optimization rules suggested in some papers to make sure, that the models are quite identical in the referred report and in the application considered.
Misunderstandings caused by some rules suggested for paced systems in application to the unpaced case where mentioned earlier.
Similarly the Davis [14] conjecture about unbalancing pattern is frequently cited without noticing that it was orginally formulated for a system with losses /eg [10] / !

Seemingly small differences in the investigated models lead sometimes to quite different optimization rules.

As it was illustrated by figures and tables the gain in capacity obtained through system optimization is sometimes small.
Thus there is an essential difficulty in optimization studies for systems where no exact analytical results are available / the error introduced by approximate methods may deform the conclusions obtained/

Using simulation, a great afford must be done to verify the statistical significance of the obtained results, erroneous statesments being by far not uncommon.

In this paper figures and tables were constructed mainly for the models investigated by use of analytical tools, to avoid quoting some mean values, being meaningless without citing also the backing them statistical reasoning.

Finally it has to be stressed that factors other than system capacity generally react differently to the optimization rules presented here. However the information about effects of individual stages parameter changes on system capacity remains important also if other optimization goals are chosen, making it possible to decide on a reasonable policy for the occuring tredeoffs.

References

[1] Anderson,D.R.,Moodie,C.L., "Optimal Buffer Storage Capacity in Production Line Systems ", International Journal of Production Research 7 /1969/ pp 233-240

[2] Avi-Itzhak,B.,Yadin,M., "A Sequence of Two-Servers with No Intermediate Queue" Management Science 11 /1965/ pp 553-563

[3] Avi-Itzhak,B., "A Sequence of Service Stations with Arbitrary Input and Regular Service Times " Management Science 11 /1965/ pp 565-571

[4] Barten,K., "A queueing Simulator for Determining Optimum Inventory Levels in a Sequential Process" The Journal of Industrial Engineering 13 /1962/ pp 245-252

[5] Basu R.N., "The Interstage Buffer Storage Capacity of Non-Powered Assembly Lines, a Simple Mathematical Approach", International Journal of Production Research, 15 /1977/ pp 365-382

[6] Buxey,G.M.,Slack,N.D.,Wild,R. "Production Flow Line System Design-A Review " AIIE Transactions 5 /1973/ pp 37-48

[7] Buzacott,J.A., "Automatic Transfer Lines with Buffer Stocks " International Journal of Production Research 5 /1967/ pp 183-200

[8] Buzacott,J.A., "The Role of Inventory Banks in Flow-Line Production Systems" International Journal of Production Research 9 /1971/ pp 425-436

[9] Buzacott,J.A., "The Effect of Station Breakdowns and Random Processing Times on the Capacity of Flow-Lines with In-Process Storage" AIIE Transactions 4 /1972/ pp 308-312

[10] Carnall,C.A.,Wild,R., "The Location of Variable Stations and the Performance of Production Flow Lines " International Journal of Production Research 14 /1976/ pp 703-710

[11] Caseau, P., Pujolle, G., "Throughput Capacity of a Sequence of Queues with Blocking Due to Finite Waiting Room" Typescript/1978/ to appear in IEEE Transactions on Software Engineering

[12] Chang, W., "Sequential Server Queues for Computer Communication Systems Analysis" IBM Journal of Research and Developement /1975/ pp 476-485

[13] Dattatreye, E.S., "Tandem Queueing Systems with Blocking" Ph.D. Dissertation, IEOR Department, University of California, Berkeley, January 31, 1978

[14] Davis, L.E., "Pacing Effects on Manual Assembly Lines", International Journal of Production Research 4 /1966/ pp 171-184

[15] Freeman, M.C., "The Effect of Breakdowns and Interstage Storage on Production Line Capacity" The Journal of Industrial Engineering 15 /1964/ pp 194-200

[16] Friedman, H.D., "Reduction Methods for Tandem Queueing Systems" Operations Research 13 /1965/ pp 121-131

[17] Hatcher, J.M., "The Effect of Internal Storage on the Production Rate of a Series of Stages Having Exponential Service Times" AIIE Transactions 1 /1969/ pp 150-156

[18] Hillier, F.S., Boling, R.W., "The Effect of Some Design Factors on the Efficiency of Production Lines with Variable Operation Times" The Journal of Industrial Engineering 17 /1966/ pp 651-658

[19] Hillier, F.S., Boling, R.W., "Finite Queues in Series with Exponential or Erlang Service Times - A Numerical Approach" Operations Research 15 /1967/ pp 286-303

[20] Hunt, G.C., "Sequential Arrays of Waiting Lines" Operations Research 4 /1956/ pp 674-683

[21] Kala, R., Hitchings, G.G., "The Effects of Performance Time Variances on a Balanced Four-Station Manual Assembly Line" International Journal of Production Research 11 /1973/ pp 341-353

[22] Kawashima,T., "Reverse Ordering of Services in Tandem Queues" Memoirs of The Defense Academy Japan 15 /1975/ pp 151-159

[23] Knott,A.D., "The Inefficiency of a Series of Work Stations - A Simple Formula" International Journal of Production Research 8 /1978/ pp 109-119

[24] Kobayashi, H., "Application of the Diffusion Approximation to Queueing Networks" Journal of ACM 21 /1974/ pp 316-328

[25] Koenigsberg,E., "Production Lines and Internal Storage", Management Science 5 /1956/ pp 410-433

[26] Kraemer,S.A.,Love,R.F., "A Model for Optimizing the Buffer Inventory Storage Size in a Sequential Production System" AIIE Transactions 11 /1970/ pp 64-

[27] Labetoulle,J.,Pujolle,G., "A Study of Queueing Networks with Deterministic Service and Application to Computer Networks" Acta Informatica 7 /1976/ pp 183-195

[28] Lavenberg,S.S "Stability and Maximum Departure Rate of a Certain Open Queueing Networks Having Finite Capacity Constraints", RAIRO- Informatique 12/1978/ pp 353-370

[29] Masso,J.,Smith,M.L., "Interstage Storages for Three Stage Lines Subject to Stochastic Failures" AIIE Transactions 6 /1974/ pp 354-358

[30] McGee,G.R.,Webster,D.B., "An Investigation of a Two-Stage Production Line with Normally Distributed Interarrival and Service Time Distributions" International Journal of Production Research 14 /1976/ pp 251-261

[31] Muth,E.J., "The Production Rate of a Series of Work Stations with Variable Service Times" International Journal of Production Research 11 /1973/ pp 155-169

[32] Muth,E.J., "The Reversibility Property of Production Lines" Management Science 25 /1979/ pp 152-158

[33] Okamura, K., Yamashina, H., "Analysis of the Effect of Buffer Storage Capacity in Traffic Line Systems" AIIE Transactions 9 /1977/ pp 127-135

[34] Okamura, K., Yamashina, H., "Justification for Installing Buffer Stocks in Balanced and Unbalanced Flow-Line Production Systems" Typescript /1978/, to appear in AIIE Transactions

[35] Patterson, R.L., "Markov Prossses Occuring in the Theory of Traffic Flow Through an N-Stage Stochastic Service System" The Journal of Industrial Engineering 15 /1964/ pp 188-193

[36] Payne, S., Slack, N., Wild, R., "A Note on the Operating Characteristics of Balanced and Unbalanced Production Flow Lines" International Journal of Production Research 10 /1972/ pp 93-98

[37] Pujolle, G., "The Influence of Protocols on the Stability Conditions in Packet-Switching Networks" Typescript /1978/, to appear in IEEE Transactions on Communication

[38] Rao, N.P., "On The Mean Production Rate of a Two-Stage Production System of The Tandem Type" International Journal of Production Research 13 /1975/ pp 207-217

[39] Rao, N.P., "Two- Stage Production Systems with Intermediate Storage AIIE Transactions 7 /1975/ pp 414-421

[40] Rao N.P., "A Generalization of the "Bowl Phenomenon" in Series Production Systems" International Journal of Production Research 14 /1976/ pp 437-443

[41] El-Rayah, T.E., "The Efficiency of Balanced and Unbalanced Production Lines" International Journal of Production Research 17 /1979/ pp 61-75

[42] El-Rayah, T.E., "The Effect of Inequality of Interstage Buffer Capacities and Operation Time Variability on the Efficiency of Production Line Systems" International Journal of Production Research 17 /1979/ pp 77-89

[43] Reeve,N.R.,Thomas,W.H., "Balancing Stochastic Assembly Lines" AIIE Transactions 5 /1973/ pp 223-229

[44] Sachs,J., "Ergodicity of Queues in Series" Annales of Mathematical Statistics 31 /1960/ pp 579-588

[45] Shimshak, D.G., "Optimal Sequencing of Servers in the Two-Station Series Queueing System" Journal of the Operational Research Society 29 /1978/ pp 779-788

[46] Smith,L.D.,Brumbaugh,P., "Allocating Inter-Station Inventory Capacity in Unpaced Production Lines with Heteroscedastic Processing Times" International Journal of Production Research 15 /1977/ pp 163-172

[47] Slack,N.D.C., "The weibull Distribution and Its Use in Describing Service Time Distributions" Prod.Mgmt.Res.Group Monograph 72/1, University of Bradford /1972/

[48] Slack,N.D.C., Wild,R., "Production Flow Line and "Collective" Working: A Comparison" International Journal of Production Research 13 /1975/ pp 411-418

[49] Suzuki,T.,Kawashima,T., "Reduction Methods for Tandem Queueing Systems" Journal of Operational Research Society of Japan 17 /1974/ pp 133-144

[50] Tembe,S.V.,Wolff,R.W., "The Optimal Order of Service in Tandem Queues", Operations Research 22 /1974/ pp 824-837

[51] Wild,R.,Slack,N.D., "The Operating Characteristics of "Single" and "Double" Non-mechanical Flow Systems" International Journal of Production Research 10 /1972/ pp 139-146

[52] Wolisz,A., "More About Two-Stage Production Systems with Finite Intermediate Storage" Report ZSAK-BO /79, Polish Academy of Sciences, Dept. of Complex Automation Systems /1979/, to appear in the International Journal of Production Research

[53] Wolisz,A., "Throughput Optimization of Two-Stage Queueing Systems with Finite Intermediate Buffer" Podstawy Sterowania 10/1980/

[54] Yamazaki, G., Sakasegawa, H., "Properties of Duality in Tandem Queueing System " Ann. Inst. Stat. Math. 27 /1975/ pp 201-212

[55] Yamazaki,G.,Sakasagawa,H.,Kawashima,T., Production Rate Estimated with Flow-Shop Reversibility" Bulletin of the JSME 21 /1978/ pp 161-171

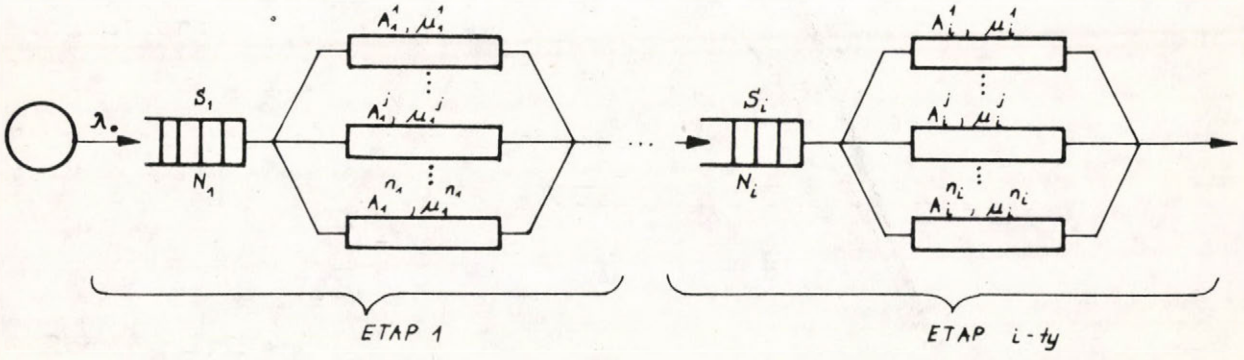


Fig. 1. *A multistage queueing system*

λ_0 - *the intensity of demands arrivals*

n_i - *the number of servers installed at i-th stage*

A_i - *j-th server belonging to i-th stage*

S_i - *buffer for demands waiting to be served at i-th stage*

N_i - *the size of buffer*

μ_i - *service intensity of server* A_i^j

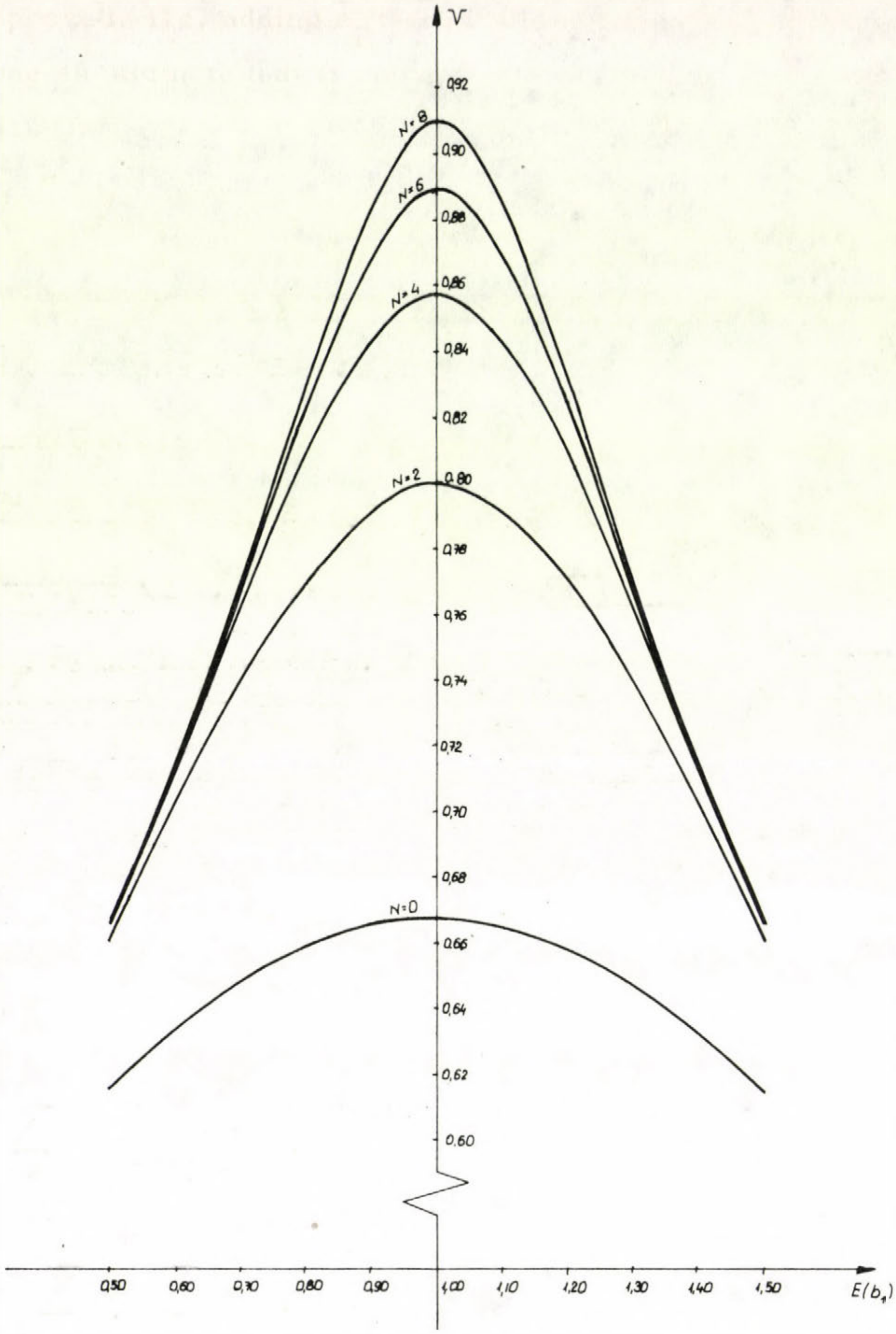


Fig. 2. Capccity of a two-stage unbalanced queueing line with exponential servers. Data taken from [18].

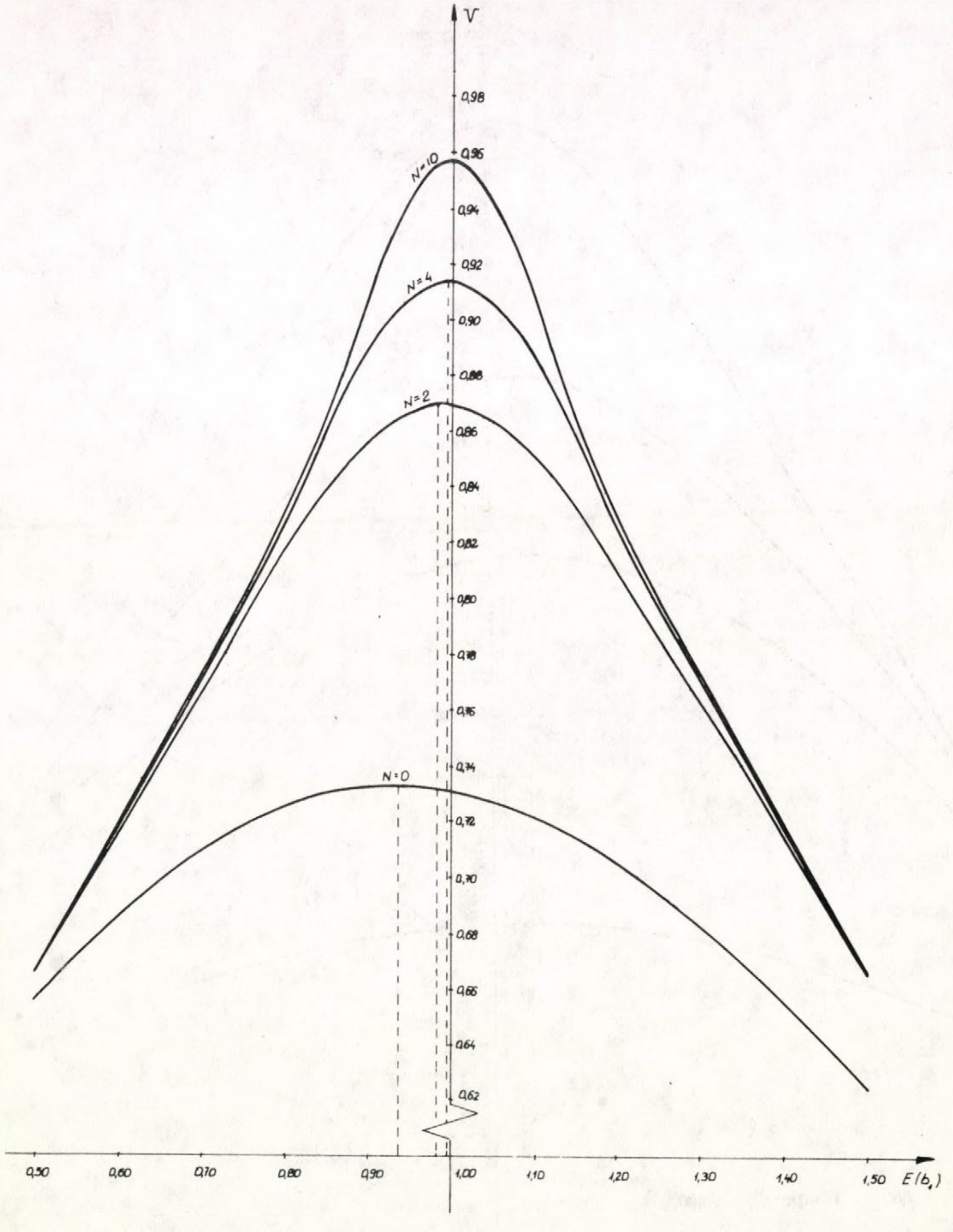


Fig. 3. Capacity of a two-stage, unbalanced queueing line. Exponential service time was assumed at the first stage, while either regular (case a) or hyperexponential (case b) distribution with $C = \sqrt{3}$ is applied at the second stage. Data taken from [52].

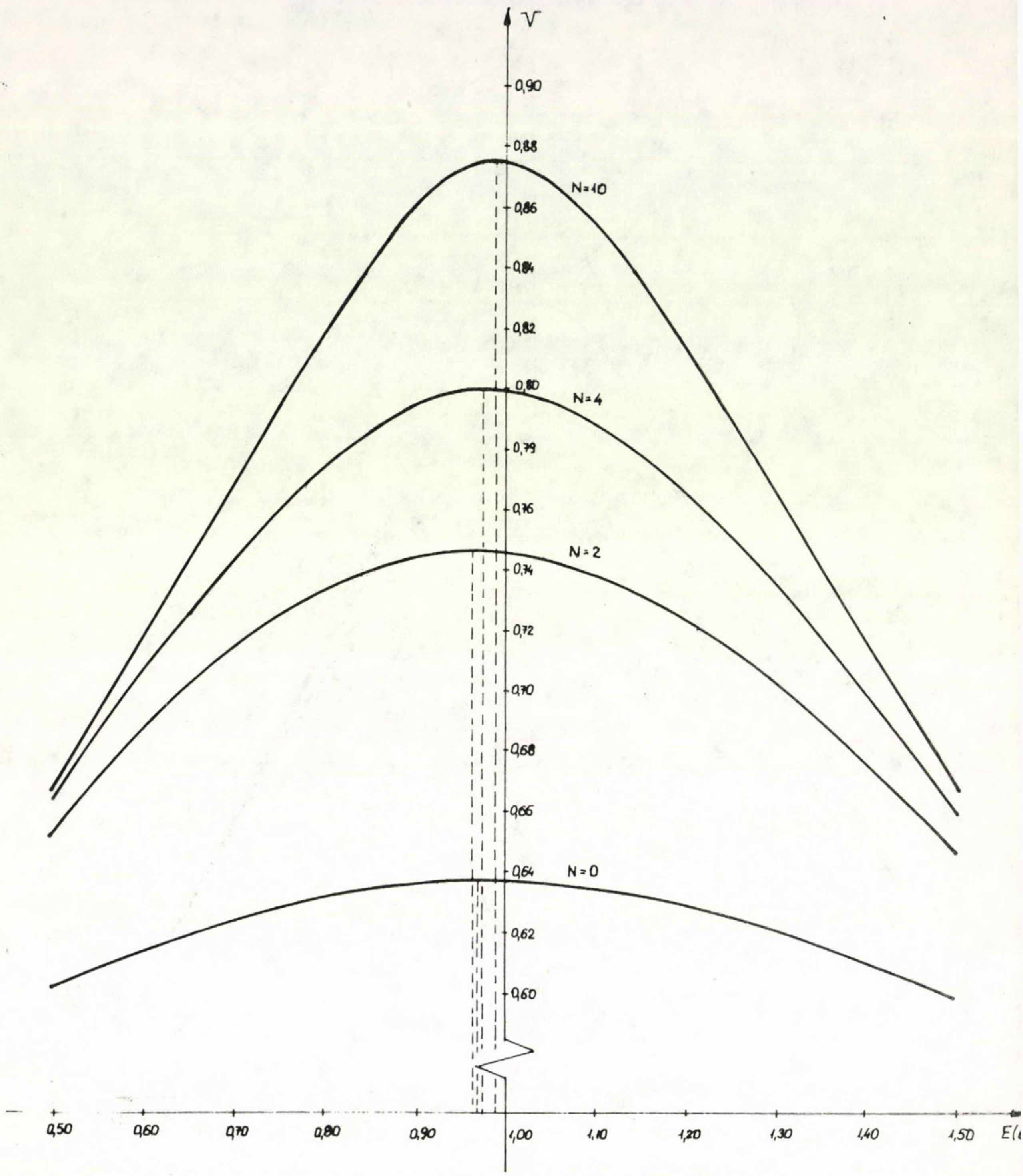

V
0,90
0,88
0,86
0,84
0,82
0,80
0,78
0,76
0,74
0,72
0,70
0,68
0,66
0,64
0,62
0,60
N=10
N=4
N=2
N=0
0,50
0,60
0,70
0,80
0,90
1,00
1,10
1,20
1,30
1,40
1,50

Fig. 3.b

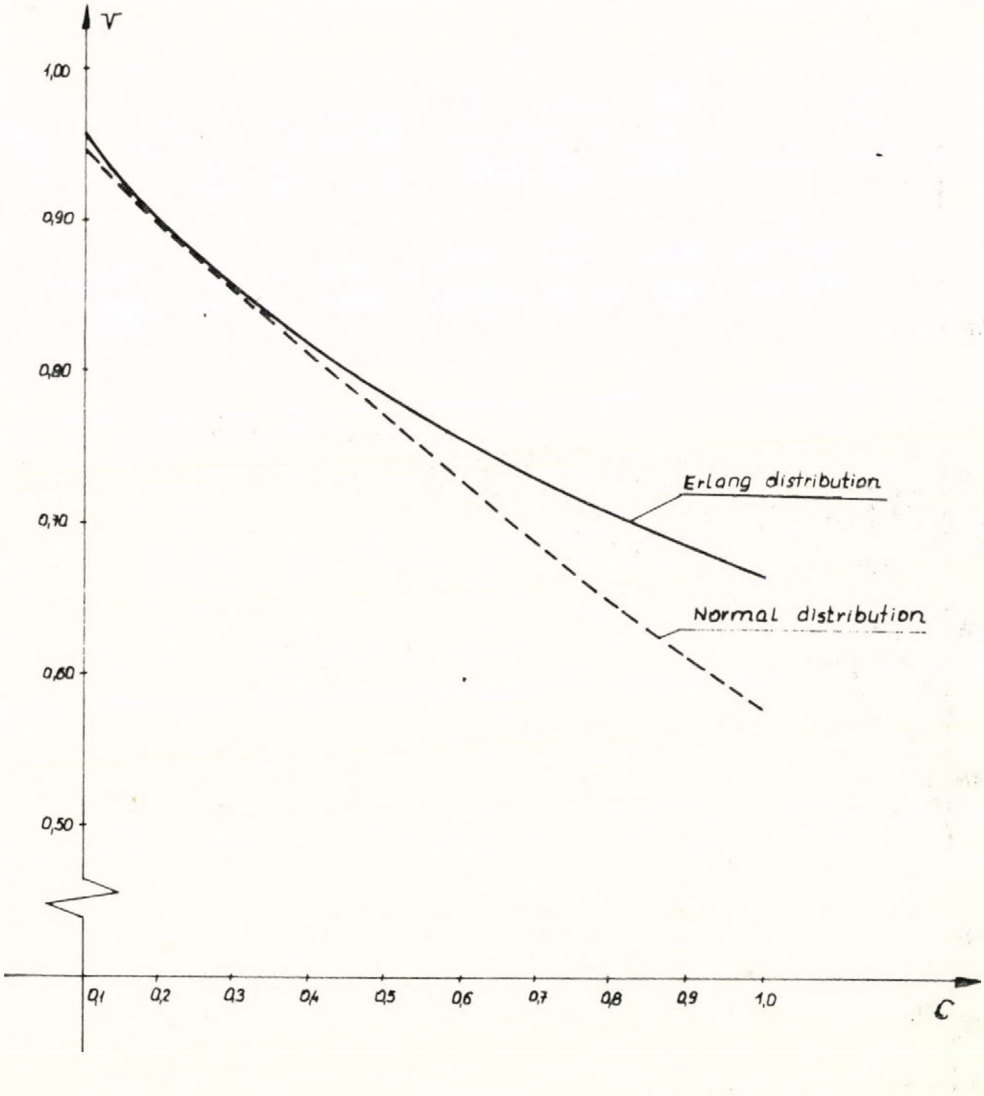


Fig. 4. Capacity of two-stage balanced queueing line versus the variability coefficient C in the case of identical service time distribution at both stages. N = 0. Data taken from [38].

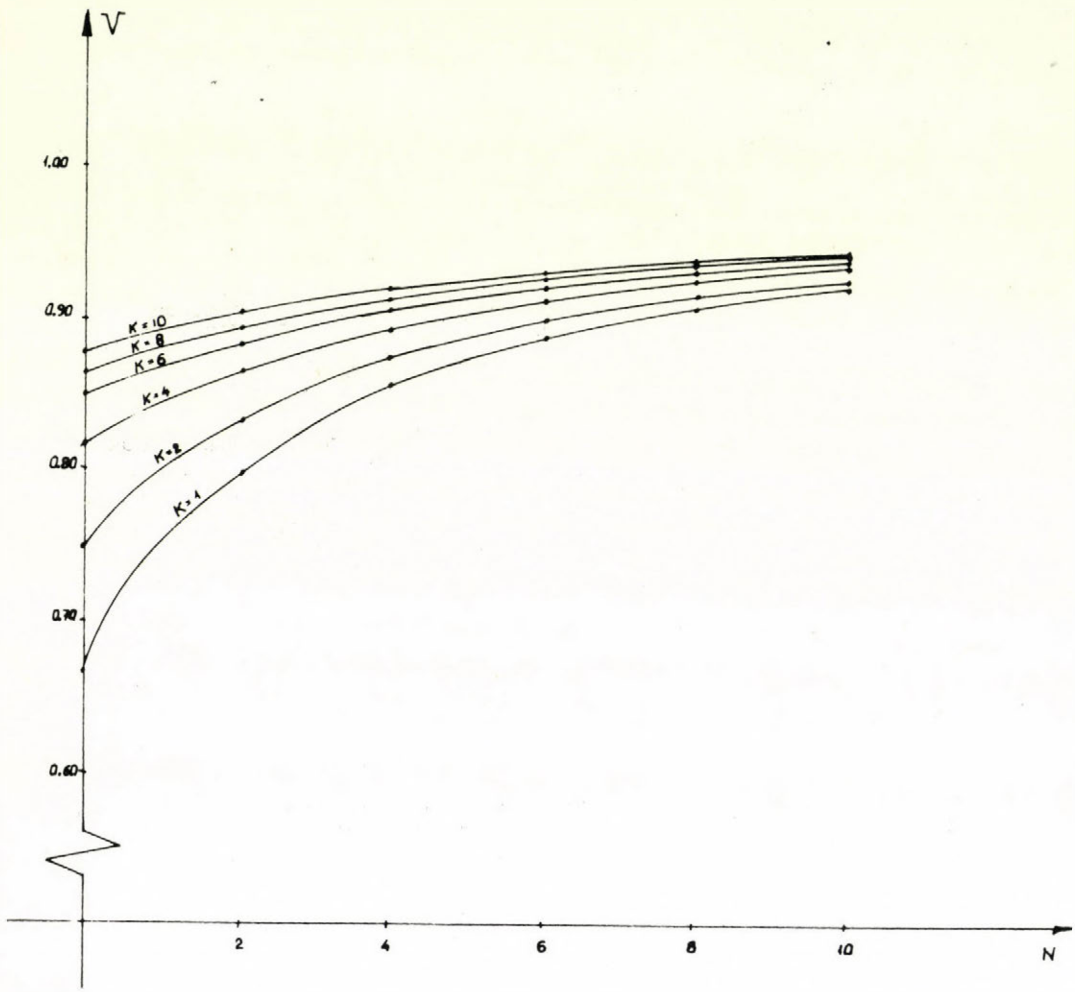


Fig. 5. The influence of buffer size N and number of stations K on system capacity. Data taken from [53].

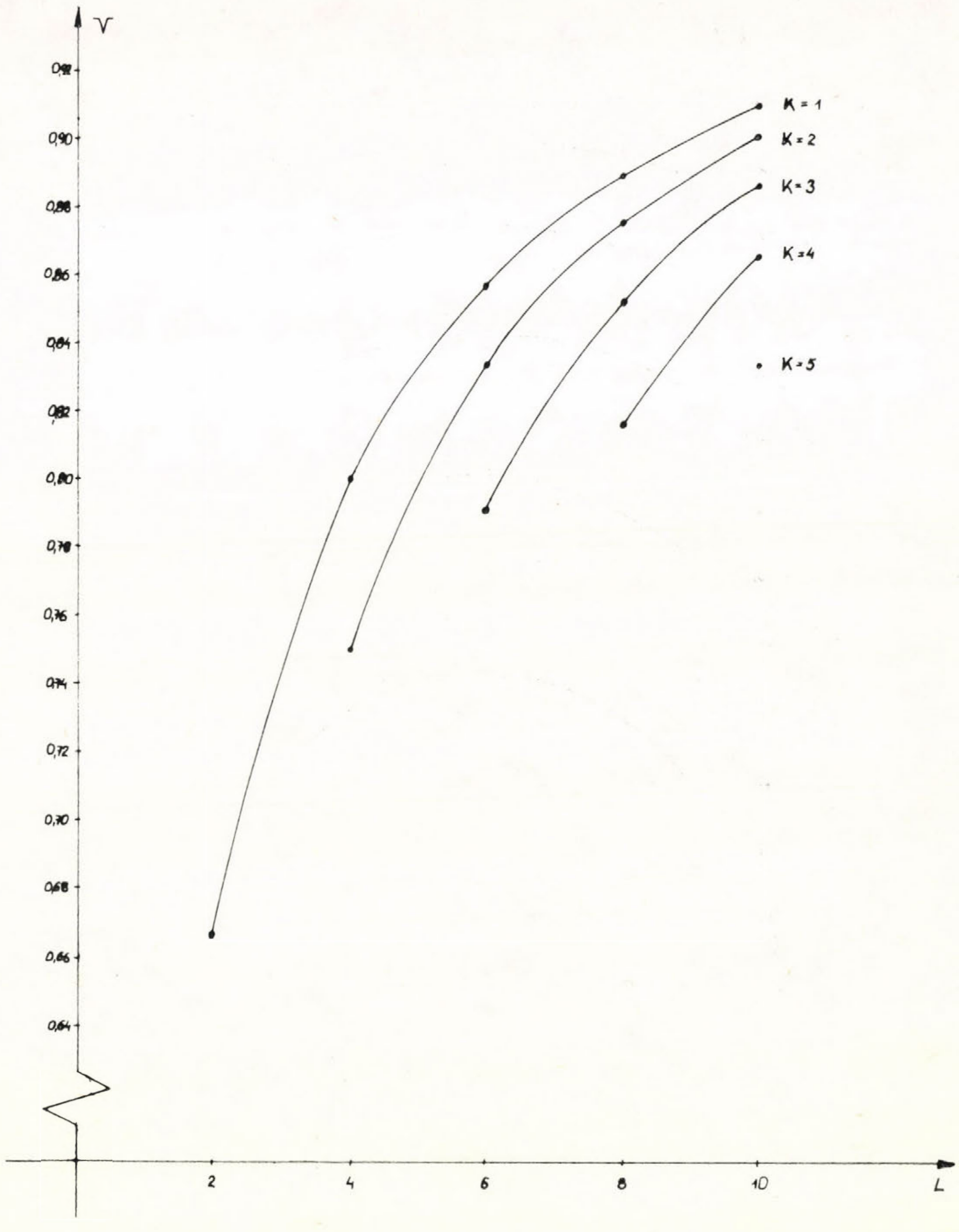


Fig. 6. Comparison of system capacity in the case of constant system accumulation L and different buffer size N as well as number of servers K, L = 2K + N. Data taken from [53].

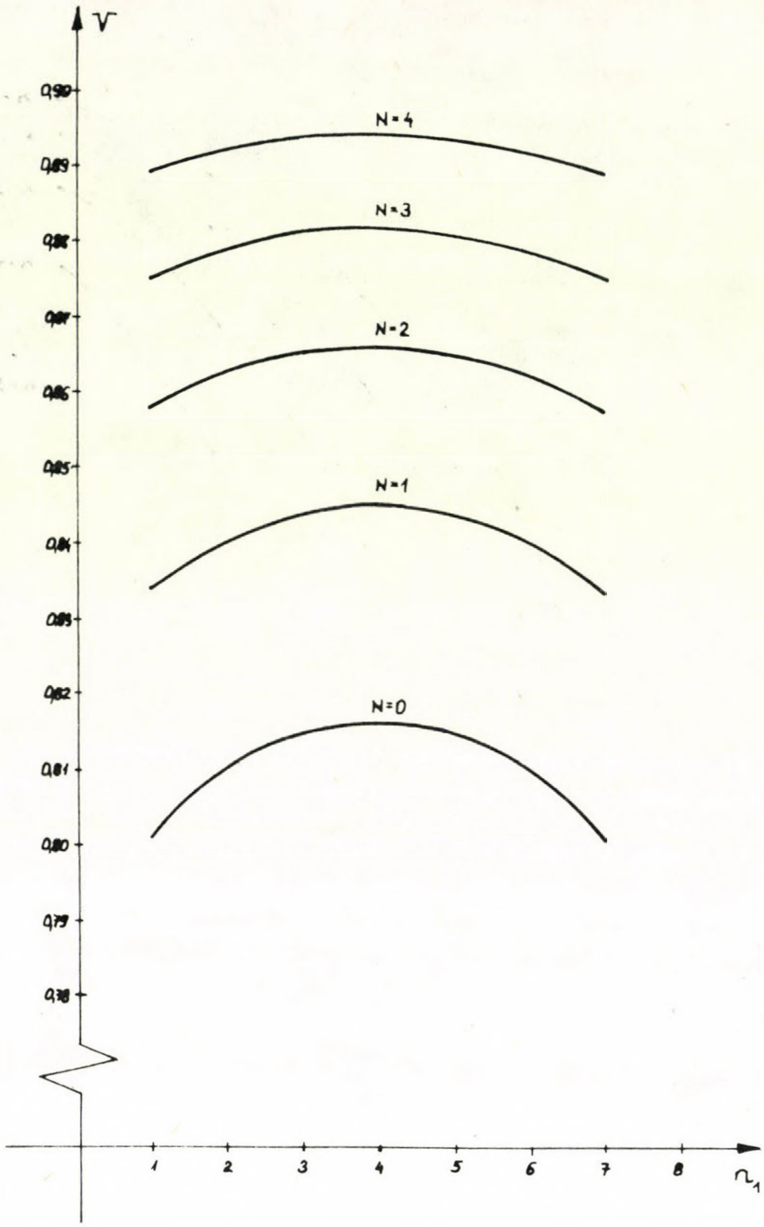
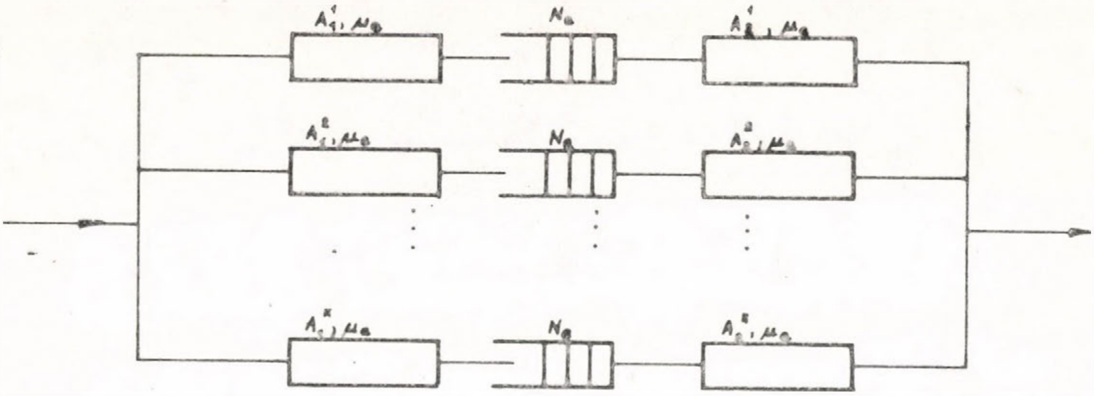


Fig. 7. Comparison of system capacity in the case of constant buffer size N and different number of servers at individual stages, $n_1 \neq n_2$, $n_1 + n_2 = const.$ *Data taken from* [53].

a)



b)

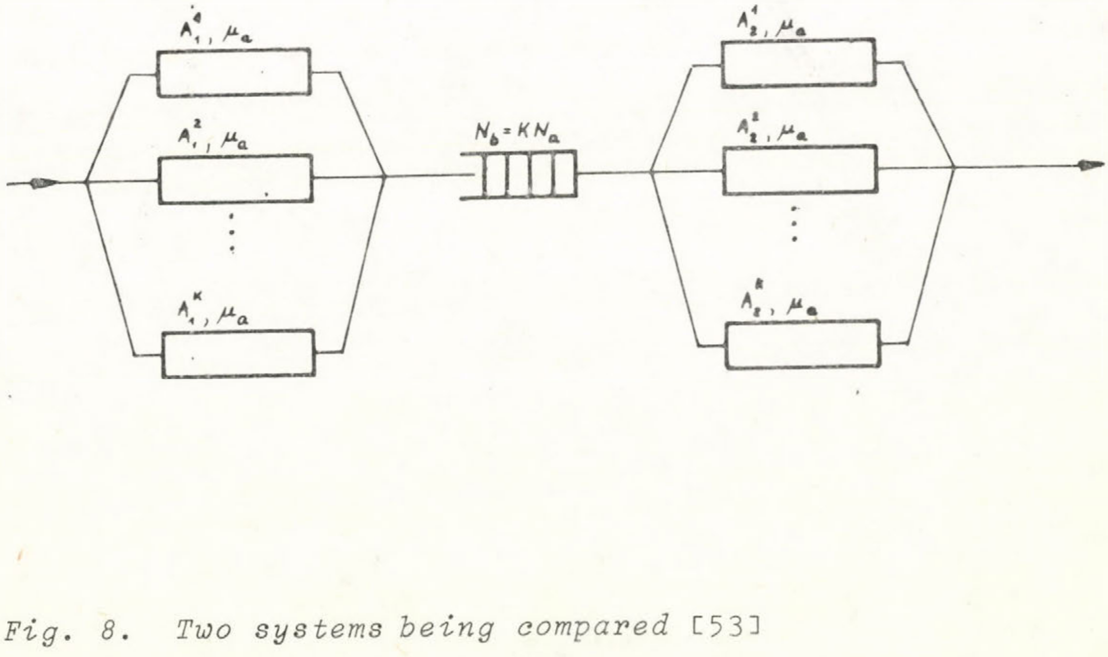


Fig. 8. Two systems being compared [53]

a) K queueing lines, each with intermediate buffer of size N_a

b) homogeneous two-stage queueing system with buffer size $N_b = k\ N_a$

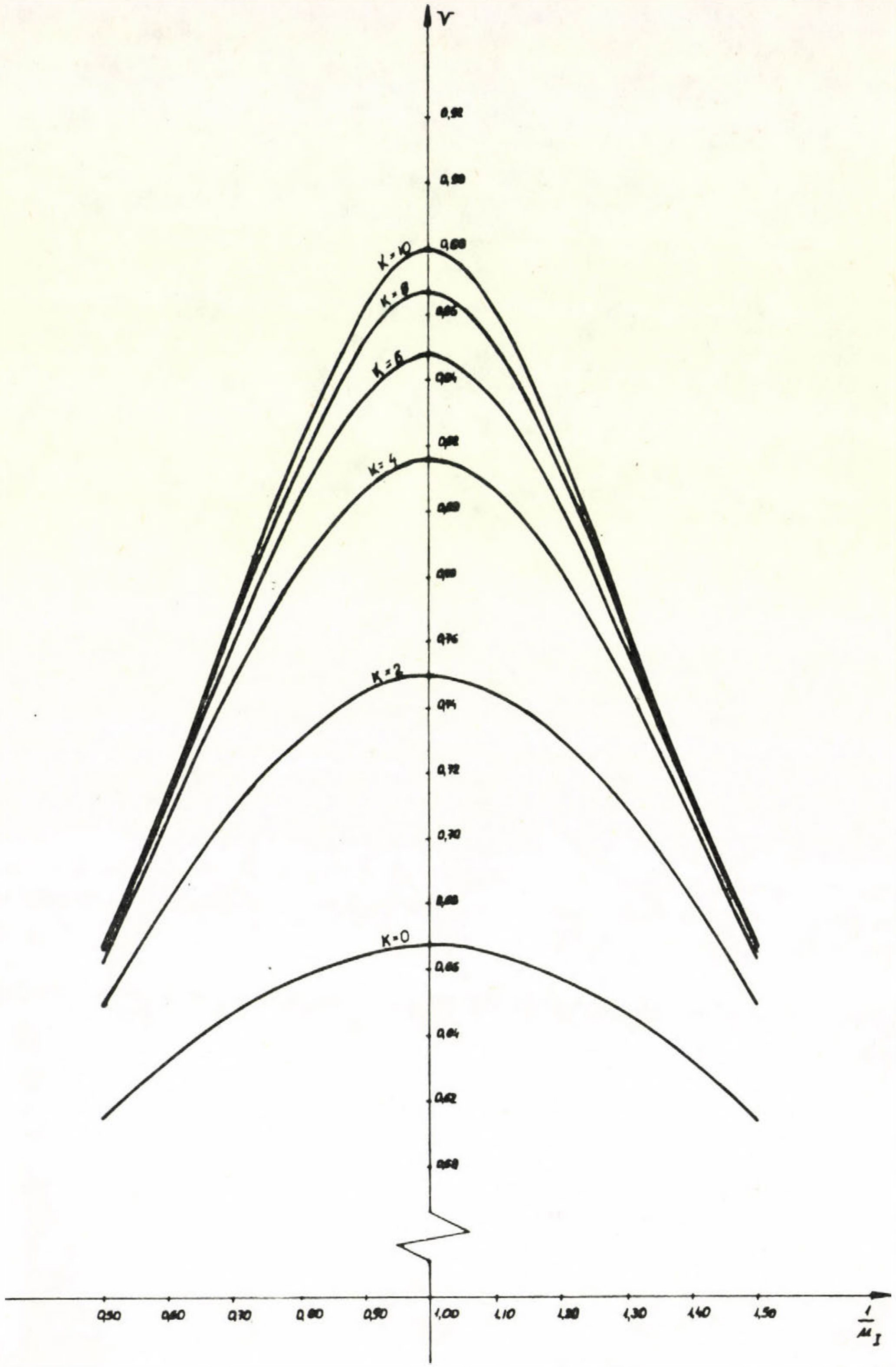


Fig. 9. The capacity of a two-stage unbalanced homogeneous system [53] having equal number of servers at both stages. N = 0.

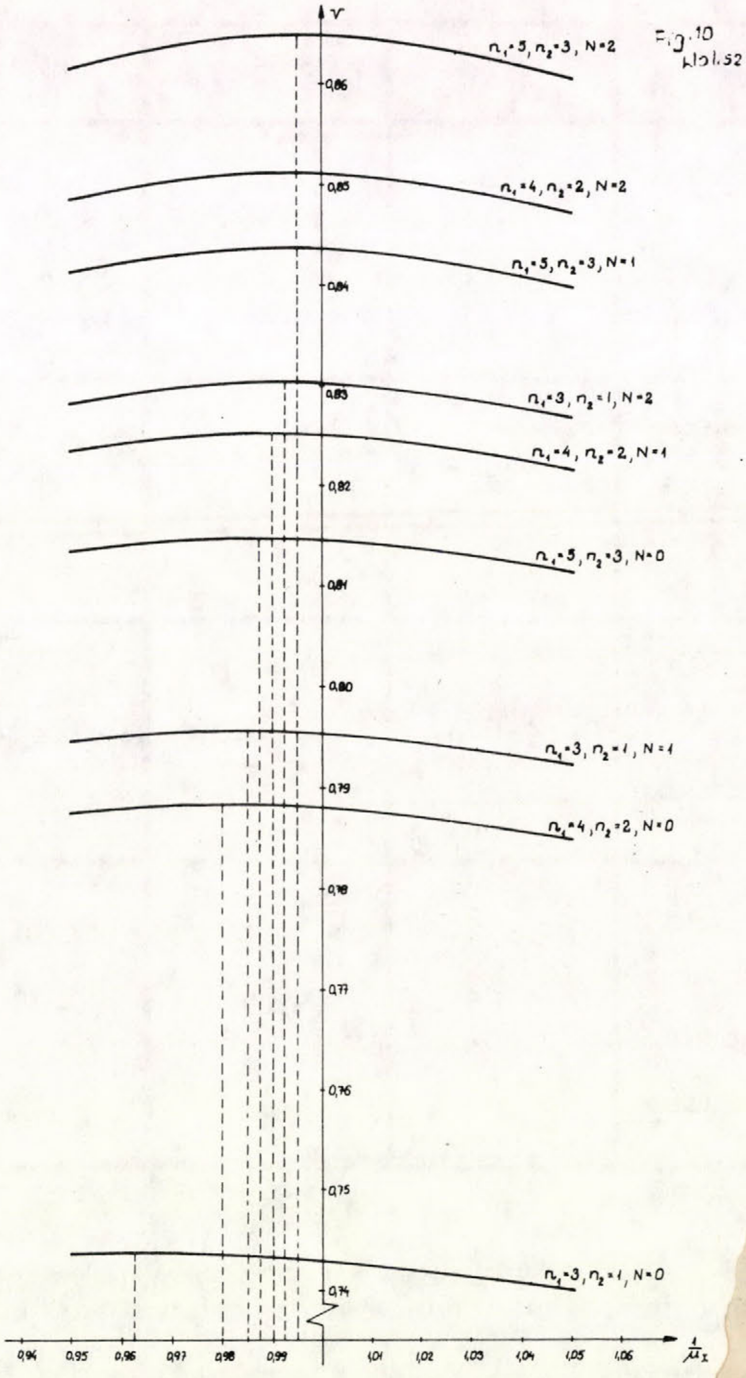


Fig. 10. The capacity of a two-stage unbalanced queueing system [53] with different number of servers at individual stages.

	M=3			M=4
	N=0	N=2	N=4	N=0
Optimal unbalancing $E(b_2)$	0.82 /0.83/	0.92	0.94	0.86
Throughput increase over the balanced case [%]	100.5 /100.54/	100.4	100.3	100.9
Range of unbalancing preserving maximal gain -0.1%	0.74-0.92	0.86-0.96	0.92-0.98	0.82-0.92
Unbalancing /in proper direction preserving the throughput of the balanced case	0.66	0.82	0.88	0.72

Table 1. The capacity of unbalanced three and four- stage queueing lines. Based upon data from [18] , data in parenthesis taken from [41]

Optimal unbalancing	$E(b_1) = 1.1$ $E(b_2) = 1.06$ $E(b_3) = 1.02$ $E(b_4) = 0.98$ $E(b_5) = 0.94$ $E(b_6) = 0.9$
Unbalancing /in proper/ direction preserving the throughput of the balanced case	$E(b_1) = 1.2$ $E(b_2) = 1.12$ $E(b_3) = 1.04$ $E(b_4) = 0.96$ $E(b_5) = 0.88$ $E(b_6) = 0.80$
Throughput increase over the balanced case [%]	1 %

Table 2. Some data characterizing the capacity of an unbalanced queueing line with M=12, N=0, according to [41] . An equality $E(b_i) = E(b_{12-i})$ holds.

The pattern considered	Capacity of balanced case	Gain from unbalancing	Optimal unbalancing $E(b_1)$	$E(b_2)$	$E(b_3)$
a/ E E D	0.6160	102.80	1.00	0.73	1.27
b/ D E E	0.6160	102.80	1.27	0.73	1.00
c/ E D E	0.6167	100.35	0.945	1.110	0.945
d/ E D D	0.7311	106.79	0.62	1.19	1.19
e/ D D E	0.7311	106.79	1.19	1.19	0.62
f/ D E D	0.7311	106.79	1.19	0.62	1.19
g/ D D D	1	No		-----	
h/ E E E	0.5641	100.5	1.09	0.82	1.09

Table 3. Optimal unbalancing of three-stage systems with different service time distributions /based on [40] /. The last pattern utilizes data from [18] E stands for exponential and D for deterministic service time distribution.

C_2	Service time distribution at the second stage					
	Uniform			Erlang		
	C_1=0, N=0	C_1=1, N=0	C_1=1, N=1	C_1=0, N=0	C_1=1, N=0	C_1=1, N=1
0.00	1.0	0.73106	0.82366	1.0	0.73106	0.82366
0.10	0.95850	0.73008	0.82263	0.91670	0.73008	0.82263
0.20	0.92030	0.72712	0.81954	0.92633	0.72721	0.81960
0.30	0.88503	0.72220	0.81443	0.89372	0.72262	0.81471
0.40	0.85237	0.71530	0.80730	0.86530	0.71635	0.80773
0.50	0.82203	0.70640	0.79824	0.83655	0.70942	0.80029

Zable 4. Capacity of a two-stage, balanced queueing line with either regular /C_1 =0/ or exponential /C_1 =1/ service time distributions at the first stage and two different service time distributions with variability coefficien C_2 at the second stege. Data taken from [39] .

N	0	1	2	3	4	5	6	7	8	9	10
V	0.66	0.75	0.8	0.833	0.857	0.876	0.889	0.9	0.909	0.917	0.924
E	12.5	6.66	4.17	2.86	2.08	1.59	1.25	1.01	0.834	0.7	0.595

Table 5. Values of V and E versus N for the two-stage system with exponential servers.

		$N_a = 0$	$N_a = 1$	$N_a = 2$
K queueing lines		0.6667	0.7500	0.8000
Multistage Queueing Systems with multiservers	K = 2	0.7500	0.8333	0.8750
	K = 3	0.7900	0.8714	0.9072
	K = 4	0.8161	0.8940	0.9256

Table 6. Comparison of the capacity achieved by a MQS and corresponding set of queueing lines. Data taken from [53] .

Összefoglalás

Véges tárolókapacitású többlépcsős sorbanállási modellek átbocsátóképességének optimalizálása

A többlépcsős sorbanállási modellek mostanában nagy figyelemnek örvendenek, mert jól használhatók számos ipari rendszer modellezésére. Ha a különböző lépcsők kiszolgálási ideje eltérő, akkor közöttük sorok jöhetnek létre és a sorok hossza a gyakorlatban korlátozott. A sorhossz túllépése az igény elvesztését eredményezi. Ezt a jelenséget "blokkolvasással" kerülik el.

Sok cikk foglalkozott már a többlépcsős rendszerekkel, de még nem állnak rendelkezésre egzakt analitikus eszközök arra az esetre, ha a kiszolgálók száma háromnál több, és kisebb rendszerekben is csak egyes speciális eseteket vizsgáltak részletesen. A cikk célja az átbocsátóképesség-optimalizálás jelenlegi helyzetének bemutatása. A feladat formális felállítása után a cikk felvázolja az átbocsátóképességet befolyásoló egyes paraméterek optimális kiválasztását.

Р Е З Ю М Е

Об оптимизации пропускной способности многофазных систем массового обслуживания с конечной очередью в отдельных фазах

Многофазные системы массового обслуживания имеют большое значение, так как их хорошо можно использовать для моделирования различных промышленных систем. Несмотря на то, что внимание ряда математиков обратилось на проблемы систем, в настоящее время нет точных аналитических средств для исследования систем, в которых число обслуживающих приборов больше трех. Для систем, в которых это число не больше трех, только частные случаи рассматриваются подробно.

Целью данной работы является показ достижений в области оптимизации пропускной способности. После постановки задачи в статье подробно изучаются основные параметры, от которых зависит пропускная способность систем.