Computational Mathematics and Mathematical Physics, Vol. 38, No. 5, 1998, pp. 838–847.
Translated from Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki, Vol. 38, No. 5, 1998, pp. 870–880.
Original Russian Text Copyright © 1998 by Vorontsov.
English Translation Copyright © 1998 by МАИК Наука/Interperiodica Publishing (Russia).

The Task-Oriented Optimization of Bases in Recognition Problems

K. V. Vorontsov
Computing Center, Russian Academy of Sciences, ul. Vavilova 40, Moscow, 117967 Russia
Received January 24, 1997

Abstract—This work is performed within the framework of the algebraic approach to the recognition problem. To construct a basic set of algorithmic operators, an iterative process is proposed such that the problem of joint optimization of the recurrent operator and a correction operation is solved at each step of this process. In this case, the basis appears to be task-oriented; i.e., it is adjusted to the given precedental information. The case of monotone correction operations is considered in detail, and the convergence of the method is proved for it. It is shown that the choice of the basis operator is reduced to the standard problem of finding a subsystem of inequalities of maximal weight. An efficient numerical procedure for constructing a monotone correction operation is described.

1. INTRODUCTION

Let the sets $\mathfrak{J}_i$ and $\mathfrak{J}_f$, called, respectively, the spaces of admissible initial and final information, be given. Computable mappings from $\mathfrak{J}_i$ into $\mathfrak{J}_f$ are called information-transforming algorithms, or, simply, algorithms. The set of all algorithms is denoted by $\mathfrak{M}^*$.

The problem of the synthesis of an algorithm consists in constructing an algorithm that satisfies a set of constraints and is defined by the predicate $Z: \mathfrak{M}^* \longrightarrow \{0, 1\}$. We will denote by the symbol Z both the predicate and the problem itself. Any algorithm such that $Z(A) = 1$ is called a solution to the problem Z, or an algorithm correct for the problem Z. A problem is called solvable if there exists a correct algorithm for it.

The mapping F from $(\mathfrak{M}^*)^p$ into $\mathfrak{M}^*$ is called a correction operation. We denote by $\mathfrak{F}^*$,

$$\mathfrak{F}^* = \bigcup_{p=0}^{\infty} \{F \mid F: (\mathfrak{M}^*)^p \longrightarrow \mathfrak{M}^*\},$$

the set of all correction operations.

If $\mathfrak{F} \subseteq \mathfrak{F}^*$ is a family of correction operations, then the set

$$\mathfrak{F}(\mathfrak{M}) = \{F(A_1, \dots, A_p) \mid F \in \mathfrak{F}, (A_1, \dots, A_p) \in \mathfrak{M}^p\}$$

is called the $\mathfrak{F}$-extension of the set of algorithms $\mathfrak{M} \subseteq \mathfrak{M}^*$.

Definition 1. A finite set of algorithms $\{A_1, \dots, A_p\}$ is called a *basis* of the problem Z with respect to the family of correction operations $\mathfrak{F}$ if there exists an algorithm correct for Z in $\mathfrak{F}(A_1, \dots, A_p)$.

Below, we will consider only recognition problems, rather than all problems of the algorithm synthesis.

Let $\mathfrak{M}^u$ be a subset of $\mathfrak{M}^*$, $\{x_k\}_{k=1}^q$ be a sequence of different elements of the set $\mathfrak{J}_i$, and $\{y_k\}_{k=1}^q$ be a sequence of elements of $\mathfrak{J}_f$. The recognition problem is defined by the predicate

$$Z(A) = \bigwedge_{k=1}^{q} (A(x_k) = y_k) \wedge (A \in \mathfrak{M}^u). \tag{1.1}$$

The sequence of pairs $\{(x_k, y_k)\}_{k=1}^q$ is called precedental, or local, information. The subset $\mathfrak{M}^u$ is called the universal information. Following [1, 2], we will assume that the universal information is taken into account at the stage of constructing the families $\mathfrak{M}$ and $\mathfrak{F}$, so that $\mathfrak{M} \subseteq \mathfrak{M}^u$ and $\mathfrak{F}(\mathfrak{M}) \subseteq \mathfrak{M}^u$. In this paper, these families are assumed to be fixed. Therefore, the main attention is paid to the local information.

There exist several approaches to solving this problem.

1. *The optimization approach.* First, we choose a model of algorithms $\mathfrak{M} \subset \mathfrak{M}^*$ on the basis of certain *a priori* considerations. Then, by means of optimization in $\mathfrak{M}$, we seek an algorithm A that satisfies the predicate Z. In practice, it may appear that either the model chosen does not contain an algorithm that is correct for Z or the method of optimization applied does not find this algorithm. Then, we have to content ourselves with an algorithm that is approximate in a certain sense or we have to change the model completely.

2. *The algebraic approach.* Along with the model $\mathfrak{M}$, we choose a family of correction operations $\mathfrak{F}$ such that $\mathfrak{M}$ contains the basis of the given problem with respect to $\mathfrak{F}$. Then, the solution is reduced to the construction of the basis $\{A_1, \ldots, A_p\}$ and to the choice of the correction operation $F \in \mathfrak{F}$ for which the algorithm $F(A_1, \ldots, A_p)$ is correct.

In fundamental papers on the algebraic approach [3, 4], it is shown that this construction is possible for a broad class of problems, called regular problems, under the condition that $\mathfrak{M}$ and $\mathfrak{F}$ are complete. The methods used in the above papers in the constructive proofs of the existence theorems are not designed for direct application in practice. For applied problems, the synthesis of solutions by the methods of the algebraic approach is reduced to the solution of a sequence of optimization problems. The goal of the present paper is an introductory study of these problems.

2. OPTIMIZATION PROBLEMS OF CONSTRUCTING A BASIS

The mapping $Q: \mathfrak{M}^* \longrightarrow [0, \infty)$ that depends on $\{(x_k, y_k)\}_{k=1}^{q}$ and $\mathfrak{M}^u$ as parameters is the quality functional of the recognition problem Z. The quality functional is chosen on the basis of *a priori* considerations, including the ease of solving optimization problems. As usual, in this case, the condition $Q(A) = 0 \Leftrightarrow Z(A) = 1$ is fulfilled.

To obtain a solution of the problem Z in the form $A = F(A_1, \ldots, A_p)$, we pose the following optimization problem: find minimal p and $A_1, \ldots, A_p \in \mathfrak{M}$, $F \in \mathfrak{F}$, such that $Q(A) \le \varepsilon$ for a given $\varepsilon \ge 0$. For $\varepsilon = 0$, we seek a correct algorithm, while, for $\varepsilon > 0$, we seek an approximate solution.

In practice, the joint optimization with respect to p elements of the set $\mathfrak{M}$ and an element of the set $\mathfrak{F}$ can face considerable technical difficulties. Therefore, it is proposed to use one of the iterative processes that, in the general case, do not guarantee the minimality of p.

1. At the pth step, $p = 1, 2, \ldots$, a submodel $\mathfrak{M}_p \subseteq \mathfrak{M}$ is fixed, and, by the minimization of the functional $Q(A)$ over the set $\mathfrak{M}_p$, an algorithm $A_p \in \mathfrak{M}_p$ is found. Then, a correction operation F_p and the algorithm $A^{(p)} = F_p(A_1, \ldots, A_p)$ for which the initial quality functional is estimated are constructed. The iterative process terminates as soon as $Q(A^{(p)}) \le \varepsilon$ for a given $\varepsilon \ge 0$. This process is frequently applied in combination with the correction operations based on the voting principle or on the selection of the competence domains.

2. In this paper, we consider another iterative process, for which the algorithms A_p, beginning with $p = 2$, are adjusted not only to the initial problem, but to diminishing the defect of the preceding algorithms:

$$A_1 = \underset{A \in \mathfrak{M}}{\arg\min}\, Q(A), \tag{2.1}$$

$$(A_p, F_p) = \underset{(A, F) \in \mathfrak{M} \times \mathfrak{F}}{\arg\min}\, Q(F(A_1, \ldots, A_{p-1}, A)), \quad p = 2, 3, \ldots. \tag{2.2}$$

The algorithm $A^{(p)} = E_p(A_1, \ldots, A_p)$ is a solution at the pth step. The iterative process terminates as soon as $Q(A^{(p)}) \le \varepsilon$ for a given $\varepsilon \ge 0$.

When analyzing this process, the following questions connected with its convergence are of particular interest:

(a) Under what conditions is the inequality $Q(A^{(p+1)}) < Q(A^{(p)})$ guaranteed?

(b) How many algorithms are required to obtain a correct $A^{(p)}$?

The answers to these questions and the practical implementation of the given process depend on the choice of particular sets $\mathfrak{I}_i$, $\mathfrak{I}_f$, $\mathfrak{M}$, and $\mathfrak{F}$. As the first step of this specific definition, we assume that algorithms from $\mathfrak{M}$ and the correction operations from $\mathfrak{F}$ have the structure that admits the application of the algebraic approach.

Following [1, 3], along with the sets $\mathfrak{I}_i$ and $\mathfrak{I}_f$ defined by the initial problem statement, we choose another set $\mathfrak{I}_e$ called the space of admissible estimates. We take the model $\mathfrak{M}$ as a family of superpositions

$$\mathfrak{M} = \mathfrak{M}^1 \circ \mathfrak{M}^0 = \{C \circ B | C \in \mathfrak{M}^1, B \in \mathfrak{M}^0\},$$

where $\mathfrak{M}^0$ is a given set of mappings from $\mathfrak{I}_i$ into $\mathfrak{I}_e$ that are called algorithmic operators and $\mathfrak{M}^1$ is a given set of mappings from $\mathfrak{I}_e$ into $\mathfrak{I}_f$ that are called decision rules. We define a family of correction operations $\mathfrak{F}$ by the family of mappings $\mathfrak{f}$:

$$\mathfrak{f} \subseteq \bigcup_{p=0}^{\infty} \{f | f: \mathfrak{I}_e^p \longrightarrow \mathfrak{I}_e\}.$$

To each mapping f from $\mathfrak{f}$, we assign the correction operation F_f: $(\mathfrak{M}^0)^p \longrightarrow \mathfrak{M}^0$ over the algorithmic operators by setting

$$F_f(B_1, \ldots, B_p)(x) = f(B_1(x), \ldots, B_p(x)) \quad \forall B_1, \ldots, B_p \in \mathfrak{M}^0, \quad x \in \mathfrak{I}_i.$$

To each pair of mappings (f, C) from $\mathfrak{f} \times \mathfrak{M}^1$, we assign the correction operation F_{fC} over algorithms by setting

$$F_{fC}(C_1 \circ B_1, \ldots, C_p \circ B_p) = C \circ F_f(B_1, \ldots, B_p)$$

for all $B_1, \ldots, B_p \in \mathfrak{M}^0$ and $C_1, \ldots, C_p \in \mathfrak{M}^1$.

Thus, the family $\mathfrak{f}$ and the set $\mathfrak{M}^1$ induce a family of correction operations $\mathfrak{F} = \{F_{fC} | f \in \mathfrak{f}, C \in \mathfrak{M}^1\}$.

The choice of an "intermediate" space $\mathfrak{I}_e$, and, as a consequence, the representation of algorithms as superpositions is a classical method of the algebraic approach. In essence, this method allows us to construct a solution in the space $\mathfrak{I}_e$, which, in contrast to $\mathfrak{I}_i$ and $\mathfrak{I}_f$, is chosen for the sake of convenience.

Further specification of the problem statement is connected with imposing additional constraints on the sets $\mathfrak{I}_i$, $\mathfrak{I}_f$, $\mathfrak{I}_e$, $\mathfrak{M}^0$, $\mathfrak{M}^1$, and $\mathfrak{f}$, as well as with the choice of the quality functional Q.

3. MONOTONE CORRECTION OPERATIONS

Consider the case where $\mathfrak{I}_f$ and $\mathfrak{I}_e$ are linearly ordered sets, $\mathfrak{f}$ is a family of all monotone mappings from $\mathfrak{I}_e^p$ into $\mathfrak{I}_e$, and $\mathfrak{M}^1$ is a nonempty family of all monotone surjective mappings from $\mathfrak{I}_e$ into $\mathfrak{I}_f$.

These constraints formally realize the following heuristic principle. If algorithms $A_1, \ldots, A_p$ are adjusted to the extrapolation of the same function, then a simultaneous increase in their output values does not lead to a decrease in the output of the algorithm $F(A_1, \ldots, A_p)$.

Introduce on $\mathfrak{I}_e^p$ an order relation by setting $(u_1, \ldots, u_p) \le (v_1, \ldots, v_p)$ if $u_i \le v_i \ \forall i = 1, 2, \ldots, p$. If vectors $\mathbf{u}$ and $\mathbf{v}$ are not comparable, we will denote it by $\mathbf{u} \| \mathbf{v}$. If U and V are arbitrary ordered sets, then the mapping $g: U \longrightarrow V$ is called monotone if, for any $u_1, u_2 \in \mathfrak{I}_e^p$, $u_1 \le u_2$ implies $g(u_1) \le g(u_2)$. If mappings $f \in \mathfrak{f}$ and $C \in \mathfrak{M}^1$ are monotone, then the correction operations F_f and F_{fC} are also called monotone.

Below, we assume that $g \ge 2$. We denote by $\mathbb{N}$ the set of indices $\{1, 2, \ldots, q\}$.

Consider the recognition problem defined by predicate (1.1).

Definition 2. A pair of indices $(j, k) \in \mathbb{N}^2$ is called a *defective pair* of the algorithm $A = C \circ B$ if $y_i < y_k$ and $B(x_j) \ge B(x_k)$. We denote by $\mathbb{D}(A)$ the set of all defective pairs of the algorithm A.

Note that an arbitrary defective pair of the algorithm $C \circ B$ is also a defective pair for all algorithms of the form $C' \circ B$, $C' \in \mathfrak{M}^1$. Therefore, the set $\mathbb{D}(A)$ does not depend on the choice of the decision rule.

Introduce the functional $Q(A) = |\mathbb{D}(A)|$, and consider its certain properties.

If $Z(A) = 1$, then the algorithm cannot have defective pairs; consequently, $Q(A) = 0$. In general, the opposite is not true. However, if $Q(C \circ B) = 0$, then it is always possible to choose a decision rule $C' \in \mathfrak{M}^1$ such that $Z(C' \circ B) = 1$. Indeed, since the algorithm $C \circ B$ has no defective pairs, $B(x_j) \le B(x_k)$ implies $y_j \le y_k$ for any pair $(j, k) \in \mathbb{N}^2$. Therefore, there exists a monotone mapping $C' \in \mathfrak{M}^1$ such that $C'(B(x_k)) = y_k \ \forall k \in \mathbb{N}$. Consequently, $Z(C' \circ B) = 1$.

Thus, for an appropriate choice of the decision rule, the conditions $Q(A) = 0$ and $Z(A) = 1$ are equivalent. This means that the functional introduced can be considered as a quality functional. Below, we will assume that it is this functional that is minimized while solving problem (2.2).

Introduce a set of q vectors $\mathbf{a}_k = (B_1(x_k), \ldots, B_p(x_k)) \in \mathfrak{J}_e^p$, $k = 1, 2, \ldots, q$. Then, the condition of correctness of the algorithm $F_{fC}(A_1, \ldots, A_p)$ is rewritten as

$$C(f(\mathbf{a}_k)) = y_k, \quad k \in \mathbb{N}. \tag{3.1}$$

A set of algorithms $A_1, \ldots, A_p$ is called *admissible* if, for any pair $(j, k) \in \mathbb{N}^2$, $y_j \neq y_k$ implies $\mathbf{a}_j \neq \mathbf{a}_k$. The admissibility is a sufficient condition for the existence of a mapping (not necessarily monotone) $C \circ f$ for which (3.1) is valid.

Let $\{u_k\}_{k=1}^q$ be an arbitrary sequence of elements of a certain ordered set. A sequence of pairs $\{(\mathbf{a}_k, u_k)\}_{k=1}^q$ is called monotone if, for all $(j, k) \in \mathbb{N}^2$, $\mathbf{a}_j \leq \mathbf{a}_k$ implies $u_j \leq u_k$.

Lemma 1. *If $A_1, \ldots, A_p$ is an admissible set of algorithms, then monotone p-ary mapping $C \circ f$ that satisfies* (3.1) *exists if and only if $\{(\mathbf{a}_k, y_k)\}_{k=1}^q$ is a monotone sequence.*

Proof. The necessity is obvious. Let us prove the sufficiency. Let the sequence $\{(\mathbf{a}_k, y_k)\}_{k=1}^q$ be monotone. Let us take an arbitrary mapping $C \in \mathfrak{M}^1$. Since it is surjective, for any $k \in \mathbb{N}$, there exists $u_k \in \mathfrak{J}_e$ such that $C(u_k) = y_k$. Let us choose elements u_k such that $y_j = y_k$ implies $u_j = u_k$ for all $(j, k) \in \mathbb{N}^2$.

The sequence $\{(\mathbf{a}_k, u_k)\}_{k=1}^q$ is monotone. Indeed, for all $(j, k) \in \mathbb{N}^2$, $\mathbf{a}_j \leq \mathbf{a}_k$ implies $y_j \leq y_k$. If $y_j = y_k$, then $u_j = u_k$ by the construction. If $y_j < y_k$, then $u_j < u_k$ due to the monotonicity of the mapping C. Therefore, $\mathbf{a}_j \leq \mathbf{a}_k$ implies $u_j \leq u_k$.

Let us define, for any $\mathbf{a} \in \mathfrak{J}_f^p$, the set $U(\mathbf{a}) = \{u_k | k \in \mathbb{N}, \mathbf{a}_k \leq \mathbf{a}\}$ and the mapping

$$f(\mathbf{a}) = \begin{cases} \max U(\mathbf{a}), & \text{if } U(\mathbf{a}) \neq \varnothing, \\ \min\{u_1, \ldots, u_q\}, & \text{if } U(\mathbf{a}) = \varnothing. \end{cases}$$

The mapping $C \circ f$ is monotone and satisfies condition (3.1) due to the admissibility of the set of algorithms $A_1, \ldots, A_p$ and the monotonicity of the sequence $\{(\mathbf{a}_k, u_k)\}_{k=1}^q$. The lemma is proved.

The set $\mathbb{D}(A_1, \ldots, A_p) = \mathbb{D}(A_1) \cap \ldots \cap (A_p)$ is called an *inherent defect* of the set of algorithms $A_1, \ldots, A_p$. The introduction of this term is justified by the following lemma.

Lemma 2. *For any p-ary monotone correction operation F, we have*

$$\mathbb{D}(F(A_1, \ldots, A_p)) \supseteq \mathbb{D}(A_1, \ldots, A_p). \tag{3.2}$$

Proof. Let $(j, k) \in \mathbb{D}(A_i)$ for all $i = 1, 2, \ldots, p$. Then, $y_j < y_k$ and $\mathbf{a}_j \geq \mathbf{a}_k$. Consequently, $C(f(\mathbf{a}_j)) \geq C(f(\mathbf{a}_k))$ for all monotone C and f, and, hence, for any monotone correction operation F, the pair (j, k) is defective for the algorithm $F(A_1, \ldots, A_p)$. The lemma is proved.

To solve problem (2.2), it is necessary to find of what pairs the set $\mathbb{D}(F(A_1, \ldots, A_p))$ consists and how its cardinality (the quality functional) depends on F and A_p. Taking into account the lemma proved, we reduce this question to the following: under what conditions inclusion (3.2) turns into an equality and what pairs generate the difference of sets $\mathbb{D}(F(A_1, \ldots, A_p)) \backslash \mathbb{D}(A_1, \ldots, A_p)$ when the equality does not hold.

The following theorem concerns the simplest but important case when relation (3.2) turns into an equality.

Theorem 1. *If $\mathbb{D}(A_1, \ldots, A_p) = \varnothing$, then there exists a monotone correction operation F such that $\mathbb{D}(F(A_1, \ldots, A_p)) = \varnothing$.*

Proof. The hypothesis of the theorem is equivalent to the fact that, for any pair $(j, k) \in \mathbb{N}^2$, $\mathbf{a}_k \leq \mathbf{a}_j$ implies $y_k \leq y_j$. Consequently, the sequence $\{(\mathbf{a}_k, y_k)\}_{k=1}^q$ is monotone, and, by Lemma 1, there exists a mapping $C \circ f$ that satisfies (3.1). We set $F = F_{fC}$ and assume that there exists a pair of numbers $(j, k) \in \mathbb{D}(F(A_1, \ldots, A_p))$. Then, $y_j < y_k$ and $C(f(\mathbf{a}_j)) \geq C(f(\mathbf{a}_k))$, which contradicts (3.1). Consequently, the set $\mathbb{D}(F(A_1, \ldots, A_p))$ is empty. The theorem is proved.

When $\mathbb{D}(A_1, \ldots, A_p) \neq \varnothing$, relation (3.2) can be both an equality and a strict inclusion. Below, we show that it is possible to find of what elements the difference $\mathbb{D}(F(A_1, \ldots, A_p)) \backslash \mathbb{D}(A_1, \ldots, A_p)$ consists without constructing the correction operation F.

Definition 3. The triple of indices $(j, s, k) \in \mathbb{N}^3$ is called a *defective triple* for the set of algorithms $A_1, \ldots, A_p$ if the following holds:

(a) the pair (j, k) is defective for all A_i, $i = 1, 2, \ldots, p$;

(b) the vector $\mathbf{a}_s$ is not comparable with $\mathbf{a}_j$ and $\mathbf{a}_k$;

(c) the chain of inequalities $y_j \leq y_s \leq y_k$ holds.

A defective triple (j, s, k) is said to be *strongly defective* if $y_j < y_s < y_k$. A pair (j, k) is called the *base* of the defective triple (j, s, k), and pairs (j, s) and (s, k) are called the edges of this triple. It is obvious that the base of any defective triple belongs to the inherent defect.

Example 1. Let $p = 2$, $q = 3$, $\mathbf{a}_1 = (3, 2)$, $\mathbf{a}_2 = (1, 3)$, $\mathbf{a}_3 = (2, 1)$, and $y_k = k$ for $k = 1, 2, 3$. Then, the triple $(1, 2, 3)$ is strongly defective. If we replace y_2 by 1 or 3, it becomes weakly defective.

Introduce on $\mathbb{N}$ a binary relation $\prec$ by setting $j \prec k$ if and only if either $a_j \leq a_k$ or $a_j \| a_k$ and $y_j \leq y_k$.

The relation $\prec$ is an order relation, since the cycle $k \prec j$, $j \prec s$, $s \prec k$ is generated on any defective triple (j, s, k).

The following lemma states that there are no other sequences of indices that prevent the relation $\prec$ from being a preorder.

Lemma 3. *If there are no defective triples in* $\mathbb{N}^3$, *then the relation* $\prec$ *is a linear preorder on* $\mathbb{N}$.

Proof. The absence of incomparable elements and the reflexivity of the relation $\prec$ are obvious. Let us show that, in the absence of defective triples, it is transitive, i.e., for any j, k, s from $\mathbb{N}$, $j \prec s$ provided that $j \prec k$ and $k \prec s$. Each of the relations $j \prec k$ and $k \prec s$ is valid in one of two cases; therefore, four cases are possible altogether.

Case 1. $\mathbf{a}_j \leq \mathbf{a}_k$ and $\mathbf{a}_k \leq \mathbf{a}_s$. Then, $\mathbf{a}_j \leq \mathbf{a}_s$, and we obtain the required $j \prec s$.

Case 2. $\mathbf{a}_j \leq \mathbf{a}_k$ and $\mathbf{a}_k \| \mathbf{a}_s$, $y_k \leq y_s$.

Consider the possible relations between $\mathbf{a}_j$ and $\mathbf{a}_s$. If $\mathbf{a}_j \leq \mathbf{a}_s$, then we obtain the required $j \prec s$. The case $\mathbf{a}_s \leq \mathbf{a}_j$ is impossible, since, otherwise, we have $\mathbf{a}_s \leq \mathbf{a}_k$, which contradicts the fact that these vectors are incomparable. Now, let $\mathbf{a}_j \| \mathbf{a}_s$. Consider the possible relations between y_j and y_k. If $y_j \leq y_k$, then $y_j \leq y_s$, and we obtain the required $j \prec s$. If $y_j > y_k$, then the assumption $y_s \leq y_j$ leads to the defective triple (k, s, j): therefore, $j \prec s$.

Case 3. $\mathbf{a}_j \| \mathbf{a}_k$, $y_j < y_k$, and $\mathbf{a}_k \leq \mathbf{a}_s$.

Consider the possible relations between $\mathbf{a}_j$ and $\mathbf{a}_s$. If $\mathbf{a}_j \leq \mathbf{a}_s$, then we obtain the required $j \prec s$. The case $\mathbf{a}_s \leq \mathbf{a}_j$ is impossible, since, otherwise, we have $\mathbf{a}_k \leq \mathbf{a}_j$, which contradicts the fact that they are incomparable. Now, let $\mathbf{a}_j \| \mathbf{a}_s$. Consider the possible relations between y_k and y_s. If $y_k \leq y_s$, then $y_j \leq y_s$, and we obtain the required $j \prec s$. If $y_k > y_s$, then the assumption $y_s \leq y_j$ leads to the defective triple (s, j, k); therefore, $y_j < y_s$, and, consequently, $j \prec s$.

Case 4. $\mathbf{a}_j \| \mathbf{a}_k$, $y_j \leq y_k$ and $\mathbf{a}_k \| \mathbf{a}_s$, $y_k \leq y_s$.

Consider the possible relations between $\mathbf{a}_j$ and $\mathbf{a}_s$. If $\mathbf{a}_j \leq \mathbf{a}_s$, then we obtain the required $j \prec s$. The case $\mathbf{a}_s \leq \mathbf{a}_j$ is impossible, since, otherwise, the triple (j, k, s) would be defective. If $\mathbf{a}_j \| \mathbf{a}_s$, then, by $y_j \leq y_s$, we have the required $j \prec s$. The lemma is proved.

Introduce on $\mathbb{N}$ another binary relation θ by setting $j\theta k$ if and only if $j \prec k$ and $k \prec j$. It is easy to verify that, if $\prec$ is a preorder, then θ is an equivalence relation on $\mathbb{N}$. Thus, for any j and k from the same class of equivalence, the conditions $\mathbf{a}_j \| \mathbf{a}_k$ and $y_j = y_k$ hold.

The following theorem states that inclusion (3.2) can be transformed into an equality if and only if there are no defective triples.

Theorem 2. *Let* $A_1, \ldots, A_p$ *be an admissible set of algorithms. Then, the following assertions hold:*

(a) *If there exists at least one strongly defective triple in* $\mathbb{N}^3$, *then the following strict inclusion holds for any monotone correction operation* F:

$$\mathbb{D}(F(A_1, \ldots, A_p)) \supset \mathbb{D}(A_1, \ldots, A_p). \tag{3.3}$$

(b) *If there are no defective triples in* $\mathbb{N}^3$, *then there exists a monotone correction operation F such that*

$$\mathbb{D}(F(A_1, \ldots, A_p)) = \mathbb{D}(A_1, \ldots, A_p). \tag{3.4}$$

Proof. Consider an arbitrary strongly defective triple (j, s, k) and an arbitrary p-ary monotone correction operation F_{fC}. Let f_j, f_s, and f_k be the values of the mapping f at the points $\mathbf{a}_j$, $\mathbf{a}_s$, and $\mathbf{a}_k$, respectively. Since $\mathbf{a}_k \le \mathbf{a}_j$, it follows from the monotonicity that $f_k \le f_j$. If we suppose that none of the pairs (j, s) and (s, k) is defective, then we obtain $f_j < f_k$, which contradicts the monotonicity. Therefore, at least one of these two pairs is defective. Since $\mathbf{a}_j \| \mathbf{a}_s$ and $\mathbf{a}_s \| \mathbf{a}_k$, none of them can belong to the intersection $\mathbb{D}(A_1) \cap \ldots \cap \mathbb{D}(A_p)$, and the strict conclusion (3.3) is valid.

Assuming that there are no defective triples in $\mathbb{N}^3$, we construct a monotone correction operation F that satisfies (3.4).

We can arrange the set $\mathbb{N}$ with respect to the preorder relation $\prec$. This means that there exists a rearrangement σ of elements of the set $\mathbb{N}$ such that $s < t$ implies $\sigma(s) \prec \sigma(t)$. Let us form a sequence $\{\tilde{y}_k\}_{k=1}^q$ by setting

$$\tilde{y}_{\sigma(t)} = \max(y_{\sigma(1)}, \ldots, y_{\sigma(t)}), \quad t \in \mathbb{N}. \tag{3.5}$$

Let us show that the sequence of pairs $\{(\mathbf{a}_k, \tilde{y}_k)\}_{k=1}^q$ is monotone. Consider an arbitrary pair $(j, k) \in \mathbb{N}^2$, $j \ne k$. It is obvious that $(j, k) = (\sigma(s), \sigma(t))$ for certain s and t. By the definition of the relation $\prec$, $\mathbf{a}_j \le \mathbf{a}_k$ implies $\sigma(s) \prec \sigma(t)$. The following two cases are possible: either $s < t$, and (3.5) directly implies the required $\tilde{y}_j \le \tilde{y}_k$, or $t < s$ and $\sigma(t) \prec \sigma(s)$, then elements $\sigma(t)$ and $\sigma(s)$ belong to the same class of equivalence with respect to the relation θ. Consequently, $y_{\sigma(t)} = y_{\sigma(s)}$, or, analogously, $\tilde{y}_j = \tilde{y}_k$. The monotonicity of the sequence of pairs $\{(\mathbf{a}_k, \tilde{y}_k)\}_{k=1}^q$ is proved.

By Lemma 1, there exists a monotone mapping of the form $C \circ f$ such that $C(f(a_k)) = \tilde{y}_k$ for all $k \in \mathbb{N}$. We set $F = F_{fC}$.

Let us take an arbitrary element $(j, k) = (\sigma(s), \sigma(t))$ of the set $\mathbb{D}(F(A_1, \ldots, A_p))$ and show that it also belongs to the set $\mathbb{D}(A_1, \ldots, A_p)$. By Definition 2, $y_j < y_k$ and $f(\mathbf{a}_j) \ge f(\mathbf{a}_k)$ hold. The first inequality implies that j and k cannot belong to the same class of equivalence with respect θ. The second inequality leads to $\tilde{y}_j \ge \tilde{y}_k$.

Assume that the condition $\mathbf{a}_k \le \mathbf{a}_j$ does not hold. Then, either $\mathbf{a}_j \le \mathbf{a}_k$ or $\mathbf{a}_j \| \mathbf{a}_k$. Taking $y_j < y_k$ into account, we conclude that $j \prec k$ in both cases. The opposite relation $k \prec j$ cannot be true, since j and k are not equivalent. This implies that $s < t$ and $\tilde{y}_j \le \tilde{y}_k$, and, consequently, $\tilde{y}_{\sigma(s)} = \tilde{y}_{\sigma(t)}$. Formula (3.5) implies that $y_{\sigma(s)} \ge y_{\sigma(t)}$, but this contradicts the condition $y_j < y_k$.

Thus, $\mathbf{a}_k \le \mathbf{a}_j$, and, consequently, the pair (j, k) is defective for all algorithms $A_1, \ldots, A_p$; i.e., it belongs to $\mathbb{D}(A_1, \ldots, A_p)$. The theorem is proved.

Thus, the set $\mathbb{D}(F(A_1, \ldots, A_p))$ consists of defective pairs of the following three kinds: the elements of an inherent defect, the edges of defective triples, and all remaining pairs. It follows from the theorem proved that we can choose a correction operation so that the absence of pairs of the second kind automatically leads to the absence of pairs of the third kind and the absence of pairs of the first kind leads to the absence of any defective pairs. On this basis, we propose the following heuristic principle for the minimization of the quality functional $Q(F(A_1, \ldots, A_p))$: consecutively eliminate the defective pairs of the first kind while trying to eliminate simultaneously the maximum number of pairs of the second kind and completely ignore the pairs of the third kind.

In accordance with this principle, we choose the algorithm $A_p = C_p \circ B_p$ so that we eliminate as many pairs from the set $\mathbb{D}(A_1, \ldots, A_{p-1})$ as possible. First of all, we will eliminate the pairs that lie at the base of the maximum number of defective triples. To eliminate the pair (j, k), it is sufficient to require that the algorithmic operator B_p satisfy the condition $B_p(x_j) < B_p(x_k)$.

In the following section, using these considerations, we introduce a weight function on the set of defective pairs and formulate an optimization problem to find the algorithm A_p.

4. PROBLEM OF THE JOINT OPTIMIZATION OF AN ALGORITHM AND A CORRECTION OPERATION

The results obtained allow us to reduce problem (2.2) to the successive determination of the algorithm $A_p \in \mathfrak{M}$ and the correction operation $F_p \in \mathfrak{F}$. The first subproblem consists in finding the algorithm A_p such that the cardinality of the set $\mathbb{D}(F(A_1, \ldots, A_p))$ is minimal for the best choice of F. The second subproblem consists in constructing a correction operation F_p for the known A_p.

First, we consider the first subproblem.

For brevity, we denote by $\mathbb{D}(A_1, \ldots, A_{p-1})$, $\mathbb{T}$, and $\mathbb{T}_0$ the set Δ, the set of all defective triples of the set of algorithms $A_1, \ldots, A_{p-1}$, and the set of all strongly defective triples, respectively.

Let $w(j, k)$ be an estimate of the number of defective pairs that are automatically eliminated from $\mathbb{D}(A_1, \ldots, A_p)$ when eliminating the pair $(j, k) \in \Delta$ and choosing the best correction operation. For example, we can set

$$w(j, k) = |\{s: (j, s, k) \in \mathbb{T}\}| + 1$$

for all $(j, k) \in \Delta$. It is possible to assign different weights to strongly defective and weakly defective triples, taking into account the fact that only the first ones generate no less than two defective pairs:

$$w(j, k) = W_0|\{s: (j, s, k) \in \mathbb{T}_0\}| + W_1|\{s: (j, s, k) \in \mathbb{T}\backslash\mathbb{T}_0\}| + 1,$$

where W_0 and W_1 are *a priori* constants, for example, $W_0 = 5/2$ and $W_1 = 3/2$.

The function introduced generates a weight function on the subsets of the set Δ in a natural way:

$$w(\Delta') = \sum_{(j, k) \in \Delta'} w(j, k), \quad \Delta' \subseteq \Delta.$$

Let $B \in \mathfrak{M}^0$ be an arbitrary algorithmic operator. We denote by $\Delta(B)$ the set of all pairs $(j, k) \in \Delta$ such that $B(x_j) < B(x_k)$. We pose the problem of finding the algorithm A_p, $p \geq 2$, in the following way.

Problem 1. Let a quality functional $Q'(A)$, $A \in \mathfrak{M}$, and a nonnegative number δ be given. It is required to find an algorithm $A_p = C_p \circ B_p$ such that its weight $w(\Delta(B_p))$ is maximal and the condition $Q'(A_p) \leq \delta$ holds.

For the given method for choosing the algorithm A_p, it is easy to prove that process (2.1), (2.2) converges in a finite number of steps.

Theorem 3. *Let $p_0 = |D(A_1)| + 1$ and, for any pair $(j, k) \in \mathbb{D}(A_1)$, there exists an algorithmic operator B in the model $\mathfrak{M}^0$ such that $B(x_j) < B(x_k)$. Then, for any p, $p = 2, 3, \ldots, p_0$, it is possible to choose a number δ such that an algorithm A^* satisfying the equation $Q(A^*) = 0$ will be found in no more than p_0 steps.*

Proof. By the hypothesis of the theorem, for any p, $p = 2, 3, \ldots, p_0$, there exists an algorithm $A'_p = C'_p \circ B'_p$ in the model $\mathfrak{M}^0$ such that $\Delta(B'_p) \neq \varnothing$. Let us set $\delta = Q'(A'_p)$. Then, there exists an algorithm $A_p = C_p \circ B_p$, for which the weight $w(\Delta(B_p))$ is maximal and the condition $Q'(A_p) \leq \delta$ holds. By virtue of the chain of inequalities $w(\Delta(B_p)) \geq w(\Delta(B'_p)) > 0$, the set $\Delta(B_p)$ cannot be empty.

Any pair (j, k) from the set $\Delta(B_p)$ belongs to the set $\mathbb{D}(A_1, \ldots, A_{p-1})$ and does not belong to $\mathbb{D}(A_1, \ldots, A_p)$. This implies that, at each step of the iteration process, beginning with $p = 2$, the cardinality of the set $\mathbb{D}(A_1, \ldots, A_{p-1})$ decreases by at least one. For a certain $p \leq |\mathbb{D}(A_1)| + 1$, the set $\mathbb{D}(A_1, \ldots, A_p)$ will be empty. By Theorem 1, this implies that $Q(A^{(p)}) = 0$. The theorem is proved.

If we use the number of defective pairs of the algorithm as the functional Q', then the problem can be formulated in a slightly different manner.

Introduce the set $\mathbb{J} = \{(j, k) \in \mathbb{N}^2 | y_j < y_k\}$. The minimization of the functional $Q(C_p \circ B_p)$ is reduced to finding an algorithmic operator B_p such that the maximum number of constraints of the form

$$B_p(x_j) < B_p(x_k), \quad (j, k) \in \mathbb{J} \tag{4.1}$$

hold.

In our case, this system is divided into two parts. On the subset of constraints Δ, $\Delta \subseteq \mathbb{J}$, we must find a consistent subsystem with the maximum weight, and, on the subset $\mathbb{J}\backslash\Delta$, a maximal (i.e., consisting of the maximum number of constraints) consistent subsystem.

Let us introduce the following weight function on the set $\mathbb{J}$:

$$w_\lambda(j,k) = \begin{cases} \lambda + w(j,k), & \text{if} \quad (j,k) \in \Delta, \\ \lambda, & \text{if} \quad (j,k) \in \mathbb{J}\setminus\Delta. \end{cases}$$

Similar to the function $w(j, k)$, it induces the weight function $w_\lambda(\mathbb{J}')$ on the set of all subsets of $\mathbb{J}$. We denote by $\mathbb{J}(B)$ the set of all pairs $(j, k) \in \mathbb{J}$ such that $B(x_j) < B(x_k)$. The set $\mathbb{J}(B)$ defines a subsystem of the system of constraints (4.1) that is consistent for a given algorithmic operator B. Let us formulate the problem of finding the algorithm A_p, $p \geq 2$, as follows.

Problem 1'. Let λ be a given nonnegative number. Find an algorithm $A_p = C_p \circ B_p$ such that its weight $w_\lambda(\mathbb{J}(B_p))$ is maximal.

Thus, in the case $Q' \equiv Q$, the problem is reduced to the well-known problem of finding a consistent subsystem with the maximal weight.

Note that parameters δ and λ have a similar sense in the statements presented. By means of these parameters, the relation between the adjustment to the precedental source information and the correction of the preceding algorithms is regulated. The maximum decrease in δ or the indefinite increase in λ lead to the solution of completely independent problems of finding the algorithms $A_1, \ldots, A_p$. As δ indefinitely increases, or when $\lambda = 0$, the opposite situation is observed—only the algorithm A_1 is adjusted to the precedental information, whereas all the subsequent algorithms are aimed exclusively at the compensation of the errors made by A_1. The choice of optimal values of the parameters δ and λ is a separate problem. In practice, they are either given *a priori* or are chosen on the basis of a number of test solutions.

5. CONSTRUCTION OF A CORRECTION OPERATION

Consider the problem of constructing a monotone correction operation F_p for a known algorithm A_p.

Problem 2. Find a correction operation F_p that minimizes the quality functional $Q(F) = |\mathbb{D}(F(A_1, \ldots, A_p))|$, $\mathbb{D}(F(A_1, \ldots, A_p)) = \{(j, k) \in \mathbb{N}^2 | f(\mathbf{a}_j) \geq f(\mathbf{a}_k) \text{ and } y_j < y_k\}$, where $F \equiv F_{fC}, f \in \mathfrak{f}$, and $C \in \mathfrak{M}^1$.

The solution of this problem is divided into two stages. First, we choose f_k, $k \in \mathbb{N}$, in order to minimize the quality functional under the condition of the monotonicity of the sequence of pairs $\{(\mathbf{a}_k, f_k)\}_{k=1}^q$. Then, we construct a monotone mapping f that satisfies the condition $f(\mathbf{a}_k) = f_k$ for all $k \in \mathbb{N}$ and, possibly, has certain additional properties; for instance, it may be continuous or smooth. The value of the quality functional becomes known after the first stage. Therefore, it is sufficient to construct the monotone approximating function only once at the last step of iterative process (2.2).

Generally, the method proposed for generating the sequence $\{f_k\}_{k=1}^q$ consists in the following. First, we eliminate from the set $\mathbb{N}$ the least possible number of indices so that the sequence of the remaining pairs $(\mathbf{a}_k, f_k)$ should be monotone. The remaining $\tilde{q}$ indices are arranged with respect to the linear preorder $\prec$ to generate a sequence $i_1 \prec \ldots \prec i_{\tilde{q}}$. Then, we consecutively insert into this sequence the indices that were eliminated previously; the place of insertion is found by minimizing the number of defective pairs. As a result, we obtain a sequence $\{i_k\}_{k=1}^q$ that defines an optimal order on the set $\mathbb{N}$, according to which the sequence $\{f_k\}_{k=1}^q$ is generated. Below, we describe the steps of this method in detail.

Step 1. We will consecutively eliminate from $\mathbb{N}$ the indices that belong to the maximum number of pairs from $\mathbb{D}(A_1, \ldots, A_p)$. Let r run through the values $q, (q-1), \ldots, (\tilde{q}+1)$. We set $\mathbb{N}_q = \mathbb{N}$,

$$\mathbb{D}_r(k) = \{j \in \mathbb{N}_r | \mathbb{D}(A_1, \ldots, A_p) \cap \{(j,k),(k,j)\} \neq \varnothing\}, \quad k \in \mathbb{N}_r,$$

$$k_r = \arg\max_{k \in \mathbb{N}_r} |\mathrm{D}_r(k)|, \quad \mathbb{N}_{r-1} = \mathbb{N}_r \setminus \{k_r\}.$$

We find the number $\tilde{q}$ from the condition $\mathbb{D}_{\tilde{q}}(k) = \varnothing$ for all $k \in N_{\tilde{q}}$. By this condition, the sequence $\{(\mathbf{a}_k, y_k)\}_{k \in \mathbb{N}_{\tilde{q}}}$ is monotone and does not have any defective pairs or triples on the set $\mathbb{N}_{\tilde{q}}$. Consequently, the relation $\prec$ is a preorder relation on $\mathbb{N}_{\tilde{q}}$. Arranging the set $\mathbb{N}_{\tilde{q}}$ with respect to this relation, we obtain the sequence of indices $i_1 \prec \ldots \prec i_{\tilde{q}}$.

Let $\mathbb{I}_r = \{i_1, \ldots, i_r\}$ be a set of pairwise different elements of the set $\mathbb{N}$. A pair $(i_s, i_t) \in \mathbb{I}^2$, $t < s$, is called an *order violation* and a *strong order violation* if $y_{i_s} < y_{i_t}$ and $\mathbf{a}_{i_s} < \mathbf{a}_{i_t}$, respectively. We denote by $\mathbb{D}(\mathbb{I})$ the set of order violations in the sequence $\mathbb{I}$. It is obvious that $\mathbb{D}(\mathbb{I}) \subset \mathbb{N}^2$.

Step 2. Thus, it is necessary to add the previously eliminated indices $k_{\tilde{q}+1}, \ldots, k_q$ into the sequence $\mathbb{I}_{\tilde{q}}$ so that the number of order violations is minimal and there are no strong order violations at all.

We will add these indices consecutively. Let r run through the values from $\tilde{q} + 1$ to q, and, for every r, let the index k_r be inserted into the sequence $\mathbb{I}_{r-1}$ before the element with the ordinal number t_r, $1 \le t_r \le r$. If $t_r = r$, then the element k_r is added to the end of the sequence. As a result, we obtain the sequence $\mathbb{I}_r$.

Define penalty functions $\xi_r(t)$, $\eta_r(t)$, and $\varphi_r(t)$:

$$\xi_r(t) = \sum_{s=1}^{t-1} [\chi(y_{k_r} < y_{i_s}) + M\chi(\mathbf{a}_{k_r} \le \mathbf{a}_{i_s})], \quad \eta_r(t) = \sum_{s=t}^{r-1} [\chi(y_{i_s} < y_{k_r}) + M\chi(\mathbf{a}_{i_s} \le \mathbf{a}_{k_r})],$$

$$\varphi_r(t) = \xi_r(t) + \eta_r(t),$$

where $t = 1, 2, \ldots, r$; the sum of the zero number of terms is assumed to be equal to zero; M is a given nonnegative number; and χ is the characteristic function of the predicate, which is equal to unity if the predicate is true and is equal to zero if the predicate is false. We set t_r equal to the number that minimizes the function $\varphi_r(t)$.

The functions introduced have the following meaning. Suppose that the element k_r is inserted into the sequence $\mathbb{I}_{r-1}$ before i_t. Then the functions $\xi_r(t)$ and $\eta_r(t)$ determine the penalty for all order violations generated by this element pairwise with elements $i_1, \ldots, i_{t-1}$ and $i_t, \ldots, i_{r-1}$, respectively. The values of the penalty for every order violation and strong order violation are equal to 1 and M, respectively. The function $\varphi_r(t)$ determines the total penalty for order violations in $\mathbb{I}_r$ that are associated with the addition of the element k_r.

Theorem 4. *Let $A_1, \ldots, A_p$ be an admissible set of algorithms and $M > q$. Then, the number of order violations in the sequence $\mathbb{I}_q$ is calculated by the formula*

$$|\mathbb{D}(\mathbb{I}_q)| = \sum_{r=\tilde{q}+1}^{q} \varphi_r(t_r),$$

and there are no strong order violations in this sequence.

Proof. We prove the theorem by induction on r. The sequence $\mathbb{I}_{\tilde{q}}$ does not contain strong order violations. Otherwise, for a certain pair $(i_s, i_t) \in \mathbb{I}_{\tilde{q}}^2$, we would have $i_t < i_s$ and $\mathbf{a}_{i_s} \le \mathbf{a}_{i_t}$; hence, $\mathbf{a}_{i_s} = \mathbf{a}_{i_t}$, which contradicts the admissibility of the set of algorithms $A_1, \ldots, A_p$.

The sequence $\mathbb{I}_{\tilde{q}}$ does not contain order violations. Otherwise, for a certain pair $(i_s, i_t) \in \mathbb{I}_{\tilde{q}}^2$, we would have $i_t < i_s$ and $y_{i_s} < y_{i_t}$; hence, $\mathbf{a}_{i_t} \le \mathbf{a}_{i_s}$, which is impossible due to the monotonicity of the sequence $\{(a_k, y_k)\}_{k \in \mathbb{N}_{\tilde{q}}}$. Thus, $\mathbb{D}(\mathbb{I}_{\tilde{q}}) = \varnothing$.

Let the assertion of the theorem be valid for the sequence $\mathbb{I}_{r-1}$. We find in $\mathbb{I}_{r-1}$ an element i_u with the maximum ordinal number u such that $\mathbf{a}_{i_u} \le \mathbf{a}_{k_r}$. We set $u = 0$ if there are no such elements at all. Similarly, we find the element i_v with the minimum ordinal number v such that $\mathbf{a}_{k_r} \le \mathbf{a}_{i_v}$; if there are no such elements, we set $v = r$. By the inductive hypothesis, there are no strong order violations in $\mathbb{I}_{r-1}$; therefore, $\mathbf{a}_{i_u} \le \mathbf{a}_{k_r} \le \mathbf{a}_{i_v}$ implies $u \le v$. If we assume that $u = v$, then we obtain the equality $\mathbf{a}_{i_u} = \mathbf{a}_{k_r}$. This equality (provided that $k_r \ne i_u$) contradicts the admissibility of the set of algorithms $A_1, \ldots, A_p$. Consequently, $u < v$.

Let us estimate the function $\varphi_r(t)$ in the following three cases:

(a) for $t \le u$, the estimates $\varphi_r(t) \ge \eta_r(t) \ge \eta_r(u) \ge M > q$ hold;

(b) for $t > v$, the estimates $\varphi_r(t) \ge \xi_r(t) \ge \xi_r(v) \ge M > q$ hold;

(c) for $u < t \leq v$, we find the upper estimate. Since $u < t$, the relation $\mathbf{a}_{i_s} \leq \mathbf{a}_{k_r}$ does not hold for all s, $t \leq s < r$. Consequently, $\eta_r(t) \leq r - t$. Similarly, by $t \leq v$, the relation $\mathbf{a}_{k_r} \leq \mathbf{a}_{i_s}$ does not hold for all s such that $1 \leq s < t$. Consequently, $\xi_r(t) < t$. Summing over all s, we obtain $\varphi_r(t) < r \leq q$.

Thus, the number t_r that minimizes the function $\varphi_r(t)$ satisfies the condition $u < t_r \leq v$. This leads to two conclusions. First, the index k_r is not involved in strong order violations in the sequence $\mathbb{I}_r$. Taking into account the inductive hypothesis, we find that there are no strong order violations in $\mathbb{I}_r$. Second, the value of $\varphi_r(t_r)$ is exactly equal to the number of order violations in which the element k_r is involved. Consequently, $|\mathbb{D}(\mathbb{I}_r)| = |\mathbb{D}(\mathbb{I}_{r-1})| + \varphi_r(t_r)$. The theorem is proved.

Step 3. By means of the sequence $\{i_k\}_{k=1}^{q}$ obtained, we construct the sequence $\{f_k\}_{k=1}^{q}$ that satisfies the conditions

$$f_{i_1} \leq \ldots \leq f_{i_q}, \tag{5.1}$$

$$\text{for all } (s, t) \in \mathbb{N}^2, \ s < t \text{ and } y_{i_s} < y_{i_t} \text{ imply } f_{i_s} < f_{i_t}. \tag{5.2}$$

The sequence of pairs $\{(\mathbf{a}_k, f_k)\}_{k=1}^{q}$ is monotone, since, by Theorem 4, there are no strong order violations in $\mathbb{I}_q$. Consequently, there exists a monotone mapping f such that $f(\mathbf{a}_k) = f_k$ for all $k \in \mathbb{N}$, and a monotone correction operation $F_p = F_{fC}$ corresponds to this mapping.

Let us show that the equality $\mathbb{D}(F_p(A_1, \ldots, A_p)) = \mathbb{D}(\mathbb{I}_q)$ holds.

Let us take an arbitrary element (i_s, i_t) of the set $\mathbb{D}(\mathbb{I}_q)$. Then, $t < s$ and $y_{i_s} < y_{i_t}$. The first inequality and (5.1) imply that $f_{i_t} \leq f_{i_s}$; hence, $(i_s, i_t) \in \mathbb{D}(F_p(A_1, \ldots, A_p))$. If (i_s, i_t) is an arbitrary element of the set $\mathbb{D}(F_p(A_1, \ldots, A_p))$, then $y_{i_s} < y_{i_t}$ and $f_{i_t} \leq f_{i_s}$. Taking (5.2) into account, we find that $t < s$, and, consequently, $(i_s, i_t) \in \mathbb{D}(\mathbb{I}_q)$.

Thus, the quality functional can be calculated by the formula

$$Q(F_p(A_1, \ldots, A_p)) = \sum_{r=\tilde{q}+1}^{q} \varphi_r(t_r).$$

Note that the presented method of generating the sequence $\{f_k\}_{k=1}^{q}$ can be slightly simplified if we eliminate the first step and immediately pass to the construction of the sequence $\mathbb{I}_q$ beginning with $\tilde{q} = 2$ and an arbitrary $\mathbb{I}_{\tilde{q}} = \{i_1, i_2\}$, where $i_1 < i_2$. However, the numerical experiments demonstrate that this modification leads to a certain deterioration of the quality of the results and does not provide any advantages compared with the method described.

ACKNOWLEDGMENTS

This work was supported by the Russian Foundation for Basic Research, project no. 96-01-00552.

I thank Yu.I. Zhuravlev for his support and K.V. Rudakov for setting up the problem and attention to this work.

REFERENCES

1. Rudakov, K.V., Universal and Local Constraints in the Problem of the Correction of Heuristic Algorithms, *Kibernetika*, 1987, no. 2, pp. 30–35.
2. Rudakov, K.V., Completeness and Universal Constraints in the Problem of the Correction of Heuristic Classification Algorithms, *Kibernetika*, 1987, no. 3, pp. 106–109.
3. Zhuravlev, Yu.I., An Algebraic Approach to Recognition and Classification Problems, *Probl. Kibern.*, 1979, no. 33, pp. 5–68.
4. Zhuravlev, Yu.I., Correct Algebras over a Set of Incorrect (Heuristic) Algorithms: I, *Kibernetika*, 1977, no. 4, pp. 14–21; Correct Algebras over a Set of Incorrect (Heuristic) Algorithms: II, *Kibernetika*, 1977, no. 6, pp. 21–27; Correct Algebras over a Set of Incorrect (Heuristic) Algorithms: III, *Kibernetika*, 1978, no. 2, pp. 35–43.