

---

## FINDING THE RELATIONSHIP BETWEEN A SEARCH ALGORITHM AND A CLASS OF FUNCTIONS ON DISCRETE SPACE BY EXHAUSTIVE SEARCH<sup>1</sup>

Victor Nedel'ko, Svetlana Nedel'ko

**Abstract:** *The task of revealing the relationship between a search algorithm and a class of functions those it solves is considered. Particularly, there was found a class of functions solvable by some adaptive search algorithm for a discrete space of low cardinality. To find an optimal algorithm exhaustive search was used. Algorithm quality criterion based on equivalence classes was also introduced.*

**Keywords:** *search algorithm, adaptive search, optimization, global extreme, no free launch theorem.*

**ACM Classification Keywords:** *G.1.6. Numerical analysis: Optimization; G.2.m. Discrete mathematics: miscellaneous.*

---

### Introduction

According to the no free launch theorem [Wolpert, 1996] a cardinality of the set of discrete functions solvable by a search algorithm is the same for all the algorithms. A function here may be thought as a permutation of integer numbers from 1 to  $n$ , and a function is called solvable by an algorithm if the algorithm will find global extreme for given number of steps. This theorem means that having no information about class of functions one can't prove one algorithm versus other ones.

At the same time the using some comprehensive search algorithm, for example genetic or adaptive one [Lbov, 1965; Lbov, 1999], practically seems to be more reasonable than the random search without any learning. But a rational argumentation of such preference is possible only if functions solvable by chosen comprehensive algorithm occur in real world tasks more often than others. This work presents the results of a numerical experiment that consists in explicit finding the classes of functions solvable by some algorithms including an adaptive search algorithm.

One can introduce on classes of functions a preferability measure that is projected on algorithms. As such a measure a functions class variety may be used. The simplest variety measure is a number of different extremum positions over all functions in the class. The informativity criterion [Lbov, 2001; Nedel'ko, 2004] also may be used as a variety measure.

On algorithms it's also possible to introduce some complexity measure, for example, as a number of different points arrangements those may be generated by the algorithm over the all functions. For deterministic algorithms this measure corresponds to intuitive measure of "nontriviality".

The numerical experiment performed reveals a correlation between an algorithm complexity and a functions class variety.

Since a metric is given on definitional domain of functions, the set of all functions may be split onto equivalence subclasses. If one defines that an algorithm solves a given function only if it solves also all the functions of the correspondent equivalence subclass, the number of solvable functions for different algorithms cease to be equal.

Thereby it is possible to introduce some criteria those allow to say that one algorithm is better than another one.

---

<sup>1</sup> The work is supported by RFBR, grant 04-01-00858-a

---

## Formal Statement of the Problem

---

The task of global extremum search will be considered in the following statement.

Let unknown numerical function  $f$  be defined on a discrete set  $X = \{x_j \mid j = \overline{1, n}\}$  and  $j \neq k \Rightarrow f(x_j) \neq f(x_k)$ . Let's arrange all the values of the function and denote a rank of value  $f(x_j)$  by  $f_j$ . Then a function  $f$  may be thought as a permutation  $f_j$  of integer numbers from 1 to  $n$ .

A search of extremum (assume that we need a minimum) consists in successive selection of  $m$  points from  $X$ , i. e. in selection of some subset of indices  $J_m = \{j_i \in I_n \mid i = \overline{1, m}\}$ , where  $I_n$  is a set of integer numbers from 1 to  $n$ . When the next point  $x_{j_i} \in X$  is chosen not the true value  $f(x_{j_i})$ , but only  $r_i \in I_i$  – a rank of this value among the all previously taken values got to be known.

So a selection of a next point in  $X$  may be represented as a function of all the ranks obtained by previous steps, i. e.  $j_i = Q_i(r_1, \dots, r_{i-1})$ . A set of the all such functions for  $m$  steps comprises a search algorithm  $Q = \{Q_i \mid i = \overline{1, m}\}$ . Note that for each  $f$  an algorithm selects a certain set  $J_m$ .

A search algorithm may be represented by a decision tree, where nodes are assigned by  $j_i$  – indices of selected points and branches correspond to possible values of ranks  $r_i$ . So the root is assigned by the index of a point selected first. Since for the first point there are no points to compare the only one branch issues from the root. The second value may be greater or smaller than the first one therefore there are two branches issuing from the second layer node. Similarly,  $i$  branches issue from each node of  $i$ -th layer.

As a search quality criterion we shall use  $K(Q, f) = \min_{i=1, m} f_{j_i}$  – the lowest rank of function values among the points selected by the algorithm.

We shall say that a function  $f$  is solvable by an algorithm  $Q$  if  $K(Q, f) = 1$ , i. e. the minimum was found.

The goal of this paper is to reveal a relationship between an algorithm and a class of functions solvable by it.

---

## Research Method

---

The basic research method in this work is a direct running over all the variants. It poses strong dimensionality restrictions.

The number of all functions is  $n!$ , so an exhaustive search over all the functions is possible until  $n \approx 14$ .

The number of nodes in the decision tree for a search algorithm is  $\sum_{i=0}^m i!$ . Since a tree need to be stored parameter  $m$  should not be greater than 11.

Under these restrictions it is possible to find the class of functions solvable by any given algorithm using exhaustive search over all the functions.

We are interesting also in finding an algorithm being optimal for given class of functions. To solve this task one need to store all the combinations of selected  $j_i$  and received  $r_i$ . The number of such variants is  $\sum_{i=0}^m \frac{n! i!}{(n-i-1)!}$ .

Therefore the ultimate dimensionality is given by  $n \approx 12$ ,  $m \approx 5$ .

Permitted for an exhaustive search task complexity is very far from complexities of real practical problems. But it allows modeling quite nontrivial function classes and search algorithms, including an adaptive search algorithm.

### No Free Launch Theorem

According to NFL theorem an algorithm quality averaged over the all functions is the same for all the algorithms of search. For the considered statement of a search problem it means that the number of functions solved by an algorithm is the same for all the algorithms. This number is  $\frac{m}{n}n!$ .

This fact means that to prove an advantage of some algorithm one need to have an information about functions to be analyzed. Another way is to introduce some characteristics those provide a ranking of classes of functions.

### The Equivalence of Functions

Until now we didn't take into account any properties of space  $X$ , although such properties are very important for an optimization algorithm.

One of the most important properties is a presence of the metric.

Assume in  $X$  the metric  $\delta(x_{j_1}, x_{j_2}) = \Delta(j_1, j_2) = \min(|j_1 - j_2|, n - |j_1 - j_2|)$  to be given. Note that the points  $x_1$  and  $x_n$  are adjacent. This metric may be illustrated via placing the points on a circle.

The metric introduced allows to define equivalence relationship on functions. Two functions will be equivalent if they may be transformed one to another via some preserving a distance mapping the space  $X$  onto itself. It's easy to see that functions  $f'(x)$  and  $f''(x)$  are equivalent if and only if the correspondent permutations  $f'_j$  and  $f''_j$  may be transformed one to another via a round permutation and a mirror reflection if needed.

We shall say that an algorithm solves a function if all the functions of correspondent equivalence class are solvable by this algorithm.

Note that while the number of functions solvable by any algorithm is the same, the number of solvable equivalence classes may be different, so it may be used as a measure of effectiveness.

### Adaptive Search Algorithm

Placing five points in a space of cardinality 10 one can only approximately model an adaptive search, but the basic ideas of this algorithm may be represented. The main idea of adaptive search is an inclination of the next selected point both to less investigated areas of  $X$  and to already known points with smallest  $f(x)$ .

But this idea may be realized by various ways, so to avoid ambiguity let's use a class of functions. It's clear that functions solved by adaptive search algorithm should satisfy the condition of that the function values in close points are usually close. Let's take a class  $F_\kappa$  of functions with restricted variation speed, i. e.  $f \in F_\kappa \Leftrightarrow |f_{j_1} - f_{j_2}| \leq \kappa \Delta(j_1, j_2)$ . Let  $\kappa = 2$  that is the lowest possible value. Note that if  $f \in F_\kappa$  then the all equivalent functions also belong to  $F_\kappa$ .

Tab. 1. The class  $F_2$

7	5	3	1	0	2	4	6	8	9
7	5	4	1	0	2	3	6	8	9
7	6	3	1	0	2	4	5	8	9
7	6	4	1	0	2	3	5	8	9
8	5	3	1	0	2	4	6	7	9
8	5	4	1	0	2	3	6	7	9
8	6	3	1	0	2	4	5	7	9
8	6	4	1	0	2	3	5	7	9

The content of  $F_2$  by  $m = 4, n = 10$  is shown in table 1 where only one function from each equivalence class is presented.

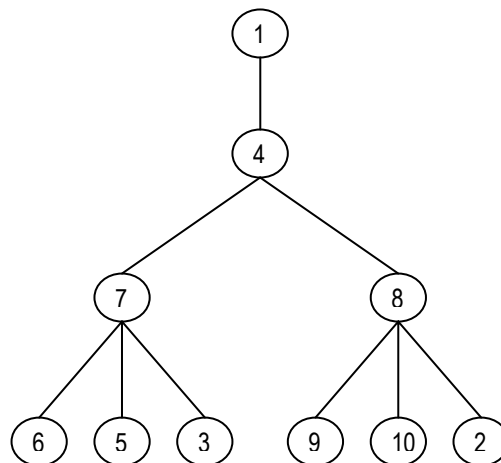


Fig. 1. The decision tree for the algorithm being optimized on the class  $F_2$  by  $m = 4, n = 10$ .

Under  $m = 4, n = 10$  the class  $F_2$  is solved by the algorithm that represented on figure 1 via a decision tree. Note that this algorithm completely reflects the ideas of adaptive planning. Indeed, the first three points are arranged equidistantly: on the first step the algorithm selects the point number 1, on the second step — the point number 4, on the third step — the point number 7 if  $f_4 < f_1$  and the point number 8 else. On the fourth step the algorithm selects a point near to previous points where smaller function values encountered, for example if  $f_7 < f_4 < f_1$  then the algorithm selects the point number 6, if  $f_4 < f_7 < f_1$  — the point number 5 etc.

Tab. 2. Some extra classes of functions solved by the algorithm being optimized on the class  $F_2$

4	5	7	2	0	6	3	1	9	8
5	1	7	3	0	6	4	2	8	9
4	6	7	2	0	5	3	1	9	8
5	6	7	2	0	4	3	1	9	8
6	2	7	1	0	5	3	4	8	9
9	2	5	3	0	4	7	1	6	8
8	1	6	3	0	5	4	2	7	9

Besides introduced in tab. 1 functions the algorithm solves 955 equivalence classes more. Some elements of these classes are in table 2.

The results obtained show that an adaptive algorithm can solve some equivalence classes including functions with rather quick changing. But in all solved classes the functions have minimum and maximum on the far distance.

## Conclusion

The research performed demonstrates that the method of a direct running over all the variants is appropriate for revealing some properties of search algorithms and a relationship between an algorithm and a class of functions solvable by it. The method allows explicit finding an algorithm being optimal for given class of functions. Note that the analytic finding an optimal algorithm is very difficult because an algorithm selecting on a regular step a point with maximal probability of extremum is not generally an optimal algorithm, so one need while planning a regular point to take account of the following steps.

Using this method we have found out the adaptive search algorithm to be optimal for the class of functions with the lowest changing speed.

The using a number of metrical equivalence classes solved by an algorithm as an effectiveness measure was also suggested.

---

## Bibliography

---

- [Wolpert, 1996] D. H. Wolpert, W. G. Macready. No Free Lunch Theorems for Search. Santa Fe Institute report, SFI-TR-95-02-010, 1996.
- [Lbov, 1965] G. S. Lbov. The choice of effective system of interdependent features. // Computational systems, N 19, Novosibirsk, IM SB RAS, 1965, pp. 21–34 (in Russian).
- [Lbov, 1999] G. S. Lbov, N. G. Startseva. Logical decision functions and issues of statistical robustness. Novosibirsk, IM SB RAS, 1999, 211 p. (in Russian).
- [Lbov, 2001] G.S. Lbov, V.M. Nedel'ko. A Maximum informativity criterion for the forecasting Several variables of different types. // Computer data analysis and modeling. Robustness and computer intensive methods. Minsk, 2001, vol 2, 43–48.
- [Nedel'ko, 2004] S. V. Nedel'ko. A transitional matrix informativity criterion and forecasting heterogeneous time series. // Artificial Intelligence, №2, Ukraine, 2004, pp.145–149. (in Russian).
- 

## Authors' Information

---

Victor Mikhailovich Nedel'ko – Institute of Mathematics SB RAS, Laboratory of Data Analysis, 660090, pr. Koptyuga, 4, Novosibirsk, Russia, e-mail: [nedelko@math.nsc.ru](mailto:nedelko@math.nsc.ru)

Svetlana Valeryevna Nedel'ko – Institute of Mathematics SB RAS, Laboratory of Data Analysis, 630090, pr. Koptyuga, 4, Novosibirsk, Russia, e-mail: [nedelko@math.nsc.ru](mailto:nedelko@math.nsc.ru)

## FORMALIZATION OF STRUCTURAL CONSTRAINTS OF RELATIONSHIPS IN MODEL „ENTITY-RELATIONSHIP”

Dmytro Buy, Lyudmila Silveystruk

**Abstract:** *The basic conceptions of the model „entity-relationship” as entities, relationships, structural constraints of the relationships (index cardinality, participation degree, and structural constraints of kind (min, max)) are considered and formalized in terms of relations theory. For the binary relations two operators (min and max) are introduced; structural constraints are determined in terms of the operators; the main theorem about compatibility of these operators' values on the source relation and inversion to it is given here.*

**Keywords:** *entity, relationship, index cardinality, participation degree, structural constraints of kind (min, max).*

**ACM Classification Keywords:** *E.4 Coding and information theory – Formal models of communication*

---

## Introduction

Data design is the creation process of logical presentation of database structure. There are different approaches to the data design, which have its own admirers. One of such models is ER-model (Entity-Relationship model, the model „entity-relationship”). This model became traditional and most popular.

This model was introduced by P. Chen in 1976. It is necessary to point out, that this model is constantly being developed and modified. More over the model „entity-relationship” has come into the set of many CASE-instruments, which have made their own contribution into its evolution. Recent studies in the ER-design are