On the Structure of the Class NP

Anatoly Plotnikov

Faculty of Computer Systems and Technologies,

Volodymyr Dahl East Ukrainian National University, UKRAINE

E-mail: a.plotnikov@list.ru

Abstract: A new class UF of problems is introduced, strictly included in the class NP, which arises in the analysis of the time verifying the intermediate results of computations. The implications of the introduction of this class are considered. First of all, we prove that P ≠ NP and establish that it needs to consider the problem "P vs UF" instead the problem "P vs NP". Also, we determine the set-theoretical of properties of one-way functions that used in cryptology.

Keywords: Class P, class NP, class UF, problem without foresight, NP-complete problem

1 Introduction

A problem is defined as a certain general question that must be answered [2]. *A mass problem* Z is defined by means of the following information:

- by a parameter list;
- by the properties of the problem solution.

A mass problem is also called simply *problem*. If the parameters of the problem have concrete values then we have *instance*.

Several major classes of problems [2, 7] are defined in complexity theory. Let us consider some of them.

A problem Z belongs to the class NP if:

- the problem can be defined by a finite number n of symbols;
- the problem solution can be represented by a finite number m of symbols, where m is a polynomial function of n: $m = f(n)$;
- the time t for verifying the obtained solution is some polynomial functions of n: $t = \varphi(n)$.

A set of problems of NP, having polynomial-time solution algorithm, forms a class P, where P $\subset$ NP. An elementary example of a problem of the class P is the problem of finding the maximum element of an array of size *n*. Clearly, for this it is necessary to consider all the elements of the array, i.e. the search time for a solution in this case equal to $O(n)$.

The problem Z_1 is *polynomially reduced* to the problem of Z_2 if each instance $X \in Z_1$ can be transformed in polynomial time in the instance $Y \in Z_2$, and a solution of Y can be converted in polynomial time to a solution of *X* [7].

In the class NP, universal or NP-complete problems are allocated. The problem Z is called NP-*complete* if any problem of NP reduced in polynomial time to the problem of Z. The set of all NP-complete problems forms the class NPC. There are hundreds of such problems in different

fields [1, 2].

Difficulties finding efficient (polynomial-time) algorithms for the NP-complete problems have led to the formulation of the problem "P vs NP", whose solution is the answer to the question: how can we effectively solve NP-complete problems? In [6] has been shown that there is a class of NP redundant.

There is also the problem of the existence of one-way function used in cryptology. This problem is also related to the solution of the previous problems "P vs NP".

The aim of this paper is to show the meaning and implications of the introduction of the new class of problems.

2 Problems without Foresight

A Turing machine is a generally accepted model of the computing process for any problem of NP [6]. The essence of this process is that the source data (parameters) of instance sequentially is converted per clock cycle (step by step). The result of these changes is to obtain a solution of the problem, if it exists, or the answer that the solution of this problem does not exist. All the results of the computational process by Turing machine to obtain the final answer, obviously, should be considered as *intermediate results*.

In order to computational process was directed to obtain the correct result, it is necessary to check each step of computing an intermediate result with its belonging to some admissible solution of the problem.

For example, if we look for a solution of the satisfiability problem then at each step of choosing different literals (values of Boolean variables) to obtain a admissible solution, the results must satisfy the condition of consistency, i.e. impossible to choose the variable x and its negation $\bar{x}$ in an intermediate solution simultaneously. And the possibility of such a choice in this problem is verified easily.

In the definition of a problem of the class \np, verification procedure of an intermediate result does not defined. Therefore, in the class NP, we select a class of problems for which verification procedure of an intermediate result is the polynomial function of the problem dimension n. The set of all such problems is denoted UF. Any problem of class UF is called a *problem without foresight*.

The proposed name connected with the study of graph problem in the papers [3, 4], in which the result of the current selection of the solution depends on the subsequent election of such elements.

Thus, from the definition of the problem without foresight follows that in the problems that belong to NP \ UF, the verification procedure of at least one intermediate result is an exponential function of the dimension of the problem. Here is another example of the optimization problem in the class NP \ UF as a decision problem.

Heavy tuple (HT)

Condition. We are given a finite set $X = \{x_1, x_2, ..., x_n\}$ of Boolean variables, a bent-function $f(x_1, x_2, ..., x_n) \in \{0,1\}$, the function $w(x_1, x_2, ..., x_n) \in Z^+$ and the boundary $E \in Z^+$ that can be represented by a finite number m of symbols, where m is a polynomial function of n: $m = \omega(n)$. The functions $f(X)$, $w(X)$ are computed in the polynomial time and the function $w(X)$ is not a constant.

Question. Is there a set of values of Boolean variables X such that $f(X)=1$ and the value of $w(X)$ is at least *E*?

Lemma 1. *The HT problem belongs to the class NP.*

Proof. Indeed, the initial data of the HT problem and the solution are finite, and the solution has linear dimension of the problem dimension. In addition, the obtained tuple (solution) may be verified in the polynomial time. Therefore, HT $\in$ NP.

Q.E.D.

Lemma 2. *The HT problem can not be solved in a polynomial time.*

Proof. In the HT problem, there is an implicit requirement to consider all tuples on which the bent-function $f(X)$ equals to 1. It follows of a fact that finding a tuple of maximum weight can only if we consider all the elements of an array of such tuples. Since one of the properties of a bent-function is that exactly on a half of the tuples X its value equal to 1, and on the other half of tuples equal to 0, the verification of the intermediate results is possible only after considering the weights of all tuples on which $f(X)=1$. Since the number of such tuples is equal to 2^{n-1}, from here it follows the validity of the above statement.

Q.E.D.

Theorem 3. *The HT problem belongs to the class NP \ UF.*

Proof. It is easy to see that in the process of solving HT problem, intermediate result (a tuple X, where *f*(X) = 1, and there is some value *w*(X)) requires at least $O(2^{n-1})$ time units to verify that the functions *w*(X) satisfies the problem condition.

Q.E.D.

Corollary 4. $NP \setminus UF \neq \varnothing$.

Corollary 5. $UF \neq NP$.

3 On the Problem "P vs NP"

The class UF allows us to consider the problem "P vs NP" and point to another proof mismatch of these classes than in [6].

Theorem 6. *If the problem Z $\in$ NP has a polynomial-time solution algorithm then Z $\in$ UF.*

Proof. Suppose that Z $\in$ NP \ UF. It follows that the intermediate results obtained in an exponential time, and the final result is obtained in a polynomial time. Contradiction, since the final result we obtain after the construction of intermediate results.

Q.E.D.

Corollary 7. $P \subset UF$.

Theorem 8. $P \neq NP$.

Proof. *The validity of the above statement follows from Corollaries 5 and 7.* Q.E.D.

4 On the Existence of One-way Functions

The concept of a one-way function is used in cryptology. Suppose that a function $y = f(x)$ is

one-way if the value y for the given x is computed "easily", and the value x for the given y is computed "difficultly". We will specify the concepts "lightness" and "difficulty" for a one-way function.

In computational complexity theory, the concept of "easy" or "effective" of computations means that the computation time t of a function $y = f(x)$ is a polynomial function of the problem dimension n: $t = \lambda(n)$. Then the concept of "difficult" of computations means that the computation time T of the function $x = f^{-1}(y)$, when it exists, is an exponential function of the problem dimension n: $T = \Lambda(n)$. Thus, we have the following result.

Theorem 9. *If* $y = f(x)$ *is a one-way function, then it belongs to the class P, and the function* $x = f^{-1}(y)$ *exists and belongs to NP $\setminus$ UF.*

Proof. *The validity of the above statement follows from the definition of a one-way function.*

Q.E.D.

From the set-theoretic point of view, we can examine the function $y = f(x)$ as the mapping $R: Dom(R) \to Ran(R)$, where $x \in Dom(R)$ and $y \in Ran(R)$. Clear, that for every element $x \in Dom(R)$ of the mapping $x \to Ran(R)$ must contain the polynomial number of elements. On the other hand, the function $x = f^{-1}(y)$ be the mapping $R^{-1}: Dom(R^{-1}) \to Ran(R^{-1})$ or, equivalently, $Ran(R) \to Dom(R)$. In order that a function $y = f(x)$ was one-way, it is necessary, that the mapping of element $y \to Dom(R)$ has the exponential number of elements.

An example of a one-way function is the bent-function $f(x_1, x_2, ..., x_n)$ of the HT problem. In this problem, the computation of "direct" function $f(X)$ and of "weight" $w(X)$ requires, by the condition, a polynomial time of n, and the inverse problem − search tuple of the maximum weight when $f(X) = 1$ − requires an exponential time of n.

Thus, we have proved the following statement.

Theorem 10. A class $Q \subset NP$ of one-way functions is not an empty, and each one-way function $y = f(x)$ has the following properties:

- an argument x can be expressed by a finite number of symbols n;
- a value of y can be written m symbols, where m is of a polynomial function of n: $m = \pi(n)$;
- the computation time t of the function y is a polynomial function of n: $t = \varphi(n)$;
- the computation time of some intermediate values of $x = f^{-1}(y)$ (or its length) is an exponential function of n.

5 Conclusion

In our view, the formal solution "P vs NP" found in the narrow sense. But only one step has made to solving the problem of solvability for a class NP in the broad sense. This result shows that the accepted definition of the class NP is redundant and can be removed from the consideration of the problems of NP \ UF. In other words, consider the polynomial solvability of the class UF, i.e. instead the problem "P vs NP" we have a problem "P vs UF". This problem can be solved if we find a polynomial-time algorithm for solving a NP-complete problem $Z \in NPC$ or it is proven that a

polynomial-time algorithm for solving NP-complete problem does not exist.

In [5] proposed the heuristic polynomial-time algorithm for solving NP-complete problem for the maximum independent set problem. This algorithm is based on the construction of a directed graph that satisfies certain conditions that allows to find a maximal independent set as a maximum antichain poset. There have not yet found a counterexample for the proposed algorithm.

From the theory of NP-completeness, it follows that any problem of the class NP can be reduced to any NP-complete problem in a polynomial time. This raises a number of questions, since it is known that any reduction allows us to restate the difficulties of solving one problem in difficulty of other problem.

Also, it follows from the paper that a cryptosystem is theoretically stable if it satisfies Theorem 10, when it not based on the difficulties of working with huge numbers only.

References

[1] Crescenzi, P., and Kann, V. (1997), "Approximation on the web: A compendium of NP optimization problems". In *RANDOM*: 111−118.

[2] Garey, M.R., and Johnson, D.S. (1979), "Computers and Intractability". W.H. Freeman and Company, San Francisco.

[3] Plotnikov, A.D. (1997), "Clarifying the class of problems solved by a non-deterministic Turing machine. *Cybernetics and Systems Analysis*, (5): 30−36.

[4] Plotnikov, A.D., (1999), "A class of problems of NP to be worth to search an efficient solving algorithm". Available at: http: //xxx.lanl.gov/ abs/cs/9903010.

[5] Plotnikov, A.D. (2012), "Heuristic algorithm for finding the maximum independent set". *Cybernetics and Systems Analysis*, 48(5): 41−48.

[6] Plotnikov, A.D. (2012), "On the relationship between classes P and NP". "Journal of Computer Science", 8(7): 1036−1040.

[7] Reingold, F.M., Nivergelt, J., and Deo, N. (1977) "Combinatorial Algorithms (Theory and Practice)". Prentice-hall Inc., New Jersey.