Technical Report 2010-003

COMPUTING A PROPERTY ON STRUCTURES

Nikolay Mirenkov

December 15, 2010

Office for Planning and Management
The University of Aizu
Tsuruga, Ikki-machi, Aizu-Wakamatsu City
Fukushima 965-8580, Japan

Technical Report 2010-003

Title:
Computing a property on structures

Author:
Nikolay Mirenkov

Key Words and Phrases:
P≠NP, underdetermined problems, graph Hamiltonicity, enhancing the graph definition

Abstract:

The ability to compute a property on structures in polynomial time depends on the structures and the property we are going to compute. We analyze such dependences and demonstrate a basis for the existence of algorithms of non-polynomial time leading to the separation of **P** and **NP**. This is based on considering **NP** problems as one type of "boundary value" problems where specification of "boundary values" is skipped. We present an example of such a problem and its "boundary values" and discuss some issues related to making theoretical models more practical.

Report Date:
12/15/2010

Written Language:
English

Any Other Identifying Information of this Report:
To submit to Communication of ACM and/or to some Internet sides

Distribution Statement:
First Issue: 20 copies

Supplementary Notes:

Active Knowledge Engineering Laboratory
The University of Aizu
Aizu-Wakamatsu
Fukushima 965-8580
Japan

Computing a property on structures

Nikolay Mirenkov
The University of Aizu, Japan
nikmir@u-aizu.ac.jp

The ability to compute a property on structures in polynomial time depends on the structures and the property we are going to compute. We analyze such dependences and demonstrate a basis for the existence of algorithms of non-polynomial time leading to the separation of **P** and **NP**. This is based on considering **NP** problems as one type of "boundary value" problems where specification of "boundary values" is skipped. We present an example of such a problem and its "boundary values" and discuss some issues related to making theoretical models more practical.

1. Preliminary consideration

Most computer scientists and mathematicians believe that $\mathbf{P} \neq \mathbf{NP}$ [1-2]. This paper is to justify this believe.

Let us consider as a structure of our interest the interval of $\{0, \frac{\pi}{2}\}$ where for any x in this interval the following equation with the unknown (but twice differentiable) function $y(x)$ is valid:

$$y''(x) + y(x) = 0. \quad (1)$$

In fact, we can consider *N* points within this interval (with $N \rightarrow \infty$). The property to be discovered is behind the question: *is there a point* x_0 *where*

$$y(0) = 10 \cdot y(x_0)\ ? \quad (2)$$

For simplicity, we can define some level of accuracy to recognize the equality of (2). The general solution to the equation (1) is

$$y(x) = Asin(x) + Bcos(x)$$

where A and B are arbitrary constants. The structure definition does not provide any basis to determine A and B. Based on (2), the question related to the property can be rewritten in the following form: *is there a point x_0 where*

$$^{B}/_{10} = A\ sin(x_0) + B\ cos\ (x_0). \quad (3)$$

To answer this question, we have to perform some search on possible values of constants A and B. Though for the one dimensional structure this search can still be of polynomial complexity, nevertheless, the example considered is a good illustration of possible necessity

- for "additional determining the boundary conditions" which cannot be find in the structure and property definitions, or
- for replacing such conditions by some exhausting search.

"Additional boundary conditions" mean not only some possible values on structure boundary points, but also some externally defined relations and equations.

Now, let us consider another structure which is our interval (one dimensional structure) that is closured into a cycle. In this case, $y(0) = y(^{\pi}/_{2})$, constant $A = B$ and (3) is transformed into $^{1}/_{10} = sin(x_0) + cos\ (x_0)$ that does not require "additional boundary conditions" or corresponding search to answer the question (2).

These simple examples are just to demonstrate that on one dimensional structure, where local features in points are defined by an ordinary differential equation, there are two types of algorithms to compute a property on such structure: **with** or **without** "additional boundary conditions depending on the involvement of arbitrary constants."

However, if we consider two dimensional structures (for example, a $\frac{\pi}{2} \times \frac{\pi}{2}$ square) and a partial differential equation (for example, Laplace's equation) for defining local features in points, the algorithms to compute a property can depend on the involvement of arbitrary functions and on a fact that the superposition of any two solutions is also a solution. As a result, such algorithms

can be of non-polynomial time. It is important to note once more that "boundary conditions values related to the arbitrary functions" cannot be just extracted from definitions of the structure and the property. To be obtained, they need an additional search of possible values. This effect is behind $\mathbf{P \neq NP.}$

2. Underdetermined problems

Problems which require "boundary conditions" of the above mentioned type are examples of underdetermined problems. There are other forms of them. For example, a system of linear equations is considered underdetermined if there are fewer equations than unknowns. In fact, the number of equations can be equal to number of unknowns, but the maximum number of independent rows (or, the maximum number of independent columns) can be fewer.

The existence of determined and underdetermined problems is a fundamental basis of the existence of **P** and **NP** algorithms. For the **P** algorithms, which are related to determined problems, there is no necessity to obtain "boundary condition values." For the **NP** algorithms, which are related to underdetermined problems, there is no a basis for solution without such values. These differences separate **P** and **NP**. Within the framework of this view, let us consider one more example where a graph is used to represent a **structure** and a system of linear algebraic equations is used to represent a **property** which should be discovered on this structure.

Let we have a simple graph **G = G(V,E)** where **V** is a set of vertices and **E** is a set of edges. It is assumed that the graph is undirected one with unweighted edges. It is also assumed that the number of vertices is **N** and the number of edges is **M**. To simplify the explanation of our approach, we will use an example of graph **G** (with **M**=10 and **N**=6) depicted by Figures 1-3.

To represent the property of goal computing, a system of **N** equations (one equation for each vertex) is defined. Each equation is based on the graph definition and on a necessary condition for the Hamiltonicity, requiring two incident edges of a vertex to be involved into a Hamilton cycle.

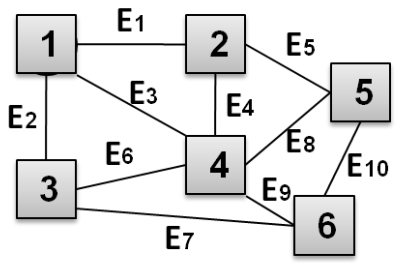

Fig.1. Graph **G** with **M**=10, **N**=6

0	1	1	1	0	0
1	**0**	0	1	1	0
1	0	**0**	1	0	1
1	1	1	**0**	1	1
0	1	0	1	**0**	1
0	0	1	1	1	**0**

Fig.2. Adjacency matrix of **G**

0	**E1**	**E2**	**E3**	0	0
E1	**0**	0	**E4**	**E5**	0
E2	0	**0**	**E6**	0	**E7**
E3	E4	E6	**0**	**E8**	**E9**
0	E5	0	E8	**0**	**E10**
0	0	E7	E9	E10	**0**

Fig.3. Edge labeling

For graph **G** specified by Figures 1-3, the system of such equations looks as follows:

$$
\begin{aligned}
E_1 + E_2 + E_3 \qquad\qquad\qquad\qquad &- 2H = F \\
E_1 \qquad + E_4 + E_5 \qquad\qquad\qquad &- 2H = F \\
E_2 \qquad\qquad + E_6 + E_7 \qquad\qquad &- 2H = F \\
E_3 + E_4 \quad + E_6 \quad + E_8 + E_9 \quad &- 2H = 3F \\
E_5 \qquad + E_8 \qquad + E_{10} \quad &- 2H = F \\
E_7 \qquad + E_9 + E_{10} \quad &- 2H = F
\end{aligned}
\qquad (4)
$$

where $E_i \in \{\mathbf{H}, \mathbf{F}\}$ (i=1,…,10) is a variable representing edge i**;** its value can be only **H** or **F** which are symbolic parameters (E_i is equal to **H** if it is involved in a Hamilton cycle and **F** if not).

Based on system (4), **the property** to be computed is defined as follows:

Find a solution of (4) such that a sub-graph of **G** *including only* **H***-edges is connected***.**

In fact, it is easy to see that **the existence of such solution is necessary and sufficient conditions for the graph Hamiltonicity**:

- Let a solution exist, then **N** variables are equal to **H**, **M-N** variables are equal to **F** and the set of **H**-edges represents a Hamilton cycle; this we can see after summation of all equations of (4): $\sum_1^M E_i$ – **NH** = (**M-N**)**F**

and after taking into account that a sub-graph of **G** including only **H**-edges is connected**.**

- Let a set of edges representing a Hamilton cycle exist, then the system of equations (4) has a solution with $E_i \in \{\mathbf{H}, \mathbf{F}\}$; this we can see after assigning value **H** to all edges from the set and value **F** to all other edges and after checking that all equations are valid with such assigning. Assuming that some equations are not valid leads to contradictions with the graph definition (with subsets of edges incident to each vertex) and/or with the existence of only two **H**-edges among each of these subsets.

Of course, the problem is how to find a solution. A standard linear solver (based on an elimination method of a cubic polynomial complexity) gives us the following expressions:

$$E_1 = r_1, \quad E_2 = -F + r_4 + r_3 + r_2 - r_1, \quad E_3 = 2H + 2F - r_4 - r_3 - r_2,$$
$$E_4 = r_4 + r_2 - r_1, \quad E_5 = 2H + F - r_4 - r_2, \quad E_6 = F - r_4 + r_1, \qquad (5)$$
$$E_7 = 2H + F - r_3 - r_2, \quad E_8 = r_4, \quad E_9 = r_3, \quad E_{10} = r_2.$$

These expressions provide a general solution depending on **M**-**N** arbitrary parameters r_i (in our example, 4 parameters) representing "boundary condition values." The graph and property definitions do not provide a basis for determining r_i. An additional search has to be done. We are interested in $\{\mathbf{H}, \mathbf{F}\}$ values, so the worst case scenario (which does not expect any specific features of the graph) for the corresponding search requires $\mathbf{O}(\mathbf{2^{M-N}})$ operations (of a polynomial type each). However, if we have a solution, it can be easily (in polynomial time) verified.

Within the framework of defining features of locality, there is an essential analogy between the (4) type equations and the (1) type equations. The graph as a set of vertices and a set of edges can be considered as a structure within a limited two dimensional area (for example, the $\frac{\pi}{2} \times \frac{\pi}{2}$ square) where distances (possibly not all) between vertices are decreased while the number of vertices is increased. As

a result, local features in vertices defined by sets of incident edges look like local features in points defined by differential equations. Therefore, computing some properties of the graph can lead us to the necessity of "boundary conditions."

Of course, for graphs with specific features, for example, *with* $\mathbf{M} - \mathbf{N} = \log_2 \mathbf{N}$ ($\mathbf{N} \to \infty$)*, the search will be of* $\mathbf{O}(\mathbf{N})$ *operations (of a polynomial type) and the total complexity of the problem will also be of a polynomial level.*

On the other hand, in spite of the exponential complexity of the worst case scenario, the (5) type expressions are a good basis for efficient heuristic search schemes where

- frequency of each parameter involvement in the expressions can be used as a priority for making decisions on the parameter values and
- symbolic analysis of the updated expression consistency after such decisions, as well as appearance of one edge vertices and connectedness of the graph, where edges equal to **F** are removed, can be applied as a special pruning technique.

The consistency mentioned requires that for each step of decision making, in all expressions involved, $\sum C_j = 1$, where C_j is a coefficient at parameter *j* that can be $\mathbf{H}, \mathbf{F}$ or r_i. For example, for E_3 in (5), $\sum C_j = 2 + 2 - 1 - 1 - 1 = 1$. In addition, the value of any $\mathrm{E_i}$ should be **H** or **F**. Assume, that at the first step a decision to assign **H** to r_2 and r_4 is made. Then all expressions including r_2 and r_4 are also updated. Before making decision at a second step, all updated expressions are checked for the consistency and the graph without edges equal to **F** is checked for connectedness and appearance of one edge vertices. In this case, the updating operations transforms expressions of (5) into a form allowing additional symbolic analysis:

$E_1 = r_1$, $E_2 = -F + 2H + r_3 - r_1$, $E_3 = 2F - r_3$

$E_4 = 2H - r_1$, $E_5 = F$, $E_6 = F$,

$E_7 = H + F - r_3$, $E_8 = H$, $E_9 = r_3$, $E_{10} = H$.

From the expression for E_3 we automatically obtain value F for r_3 and E_3, and from the expression for E_4 we automatically obtain value **H** for r_1 and E_4. After that all other E_i obtain their values, too.

3. Bridging gap between theory and practice

Specifying "boundary conditions" and, in this way, transforming, at least partially, underdetermined problems into determined ones are a way to make theoretical models be more practical (calls for such necessity, in spite of success of some practical approaches, are rather strong; see, for example, [3]). On the other hand, practice can help in discovering forms for such specifying and transforming. For example, to solve the Hamiltonicity problem on German (or European) networks of roads, which were created for centenaries, some history of the road construction can be extracted to divide the set of edges into subsets with special features allowing the introduction of systems of algebraic equations (with reasonable level of the determinant rank). Similar ideas can be used for designing devices/circuits where sets of connections can be technologically divided into subsets to help in various optimizations. Finally, the introduction of the edge hierarchy can also be used as a very practical technique for solving large-scale problems.

These simple observations provide good hints, for example, for enhancing the graph definition. This enhancing can be done by considering graph as a set of vertices and a set of edges that is a union of subsets representing a "history" or "hierarchy" of the edge appearance.

As an example, let us consider an idea of hierarchical edges which on the first level define connections between non-intersected regions, on the second level - between non-intersected sub-regions inside the regions, and so on:

$$E = \{\cup E_{ij}\}, i = 1, \dots, I; j = 1, \dots, J(i);$$

where I is the number of the hierarchical levels and $J(i)$ is the number of regions on level i. We can also assume that the number of sub-regions in a region is limited by constant C (if $C = 10$ we have $J(i) \leq 10J(i-1)$) and that there is a rule (including a full freedom) of assigning vertices in a sub-region for hosting edges representing connections of this sub-region with other sub-regions of the same hierarchical level.

Within such definition, we implicitly defined some “boundary condition values” to solve Hamiltonicity problem in polynomial time. Considering regions and sub-regions as super-vertices, at each step of algorithm we can work with graphs where a property we need can be easily find. This is because of small size graphs allowing exhausting search and because of appropriate freedom for assigning edges of inter-regional connections. In fact, to find a Hamilton cycle, we can find it between regions after that inside each region with necessary adjustments between hierarchical levels. Within a region we need only a Hamilton path covering sub-regions. In a case of applying a system of the (4) type, corresponding equations can be written not only for vertices, but also for regions and sub-regions. As a result, the number of arbitrary parameters to be considered is drastically decreased or disappeared at all.

4. Conclusion

NP problems are underdetermined problems where some arbitrary constants/ functions or relations are implicitly involved. In general cases such constants/ functions or relations cannot be extracted in a complete form from definitions of the structure and the computing property. And thus, they have to be exhaustively searched. In other words, external “boundary condition values” cannot be easily reconstructed through the definition of the structure and internal properties. **NP** problems can be considered as “ill-posed problems” requiring some “regularization” in a form of “boundary conditions.” That is why the **NP** class cannot be equal to the **P** class. Redefining models by introduction of “boundary conditions” is a way to solve serious practical problems.

References

1. Lance Fortnow, The Status of the P versus NP Problem, Communication of ACM, Vol.52, No.9, 2009, 78-86.
2. Richard J. Lipton, The P=NP Question and Gödels Lost Letter, Springer, 2010.
3. Moshe Y. Vardi, On P, NP, and Computational Complexity, Communication of ACM, Vol.53, No.11, 2010, p.5.