On Some Classes of Problems on Graphs

Volodymyr G. Skobelev

On Some Classes of Problems on Graphs

V.G. Skobelev[1,2][0000−0002−7018−2319]

[1] Glushlov Institute of Cybernetics of NAS of Ukraine,
Glushkov Ave., 40, Kyiv, 03187, Ukraine
[2] Institute of Applied Mathematics and Mechanics of NAS of Ukraine,
Gen. Batjuk Str., 19, Donetsk Region, Slov'jansk, 84116, Ukraine
skobelevvg@gmail.com

Abstract. The relevance to research the complexity of resolving Graph Theory problems is caused by its numerous applications. In the given paper this problem is investigated in terms of space complexity of data structures that represent analyzed graphs, orgraphs, and directed graphs. The following two non-trivial the simplest sets of problems of Graph Theory are investigated in detail. The first set consists of the problems that can be resolved by some algorithm with space complexity linear relative to the size of the data structure that represents the analyzed graphs. The second set consists of the following problems, such that the size of the solution significantly exceeds the size of the input data. To resolve the problem some algorithm that operates on space linear relative to the size of the data structure that represents the analyzed graphs can be applied. Besides, this algorithm uses some memory of the same size for sequential generation, one fragment after another, the solution of the problem. Some model problems that are not in these two sets of problems are considered briefly.

Keywords: Graphs · Algorithms · Complexity.

1   Introduction

At present finite graphs (i.e. ordinary graphs, orgraphs, and directed graphs) are used as mathematical models in resolving a wide class of theoretic and applied problems. Rather new areas for application of Graph Theory models are computer and social networks, agent-based technologies, and transition systems used for verification of the developed software. Therefore, analysis of the complexity for algorithms on graphs is an actual problem from the theoretic and applied point of view. In Algorithms Theory, the main attention is paid to the analysis of time complexity, while many questions for space complexity remain obscure.

The aim of the given paper is to investigate the classification of Graph Theory problems in terms of the memory size necessary to represent the analyzed graphs by basic data structures. These data structures are vertices-adjacency matrices, edges-adjacency matrices, vertices-adjacency lists, and edges-adjacency lists. The main ratios for the complexity of these data structures are established in terms "for almost all" and "on average".

Two sets of Graph Theory problems are investigated in detail.

The first set consists of the problems that can be resolved by some algorithm with space complexity linear relative to the size of data structure that represents the analyzed graphs. For these problems, the memory size necessary to represent the solution is also bounded above by some linear function of the memory size necessary to represent the input data.

The second set consists of the problems that satisfy the following three conditions.

Firstly, the memory size necessary to represent the solution significantly exceeds the memory size necessary to represent the input data.

Secondly, to resolve the problem some algorithm that operates on space linear relative to the size of data structure that represents the analyzed graphs can be applied.

Thirdly, this algorithm uses some memory of the same size to generate sequentially, one fragment after another, the solution of the problem in the explicit form.

Some problems of Discrete Mathematics and its applications that can be naturally reformulated in Graph Theory terms, and that are not in the investigated two sets of problems are briefly considered.

The basic concepts used in the given paper are the same as in [1-3].

2 Mathematical Background

In the given Section the basic concepts and definitions necessary for the presentation of the main results are introduced.

2.1 Data Structures for Graphs Representation

It is well-known that the main data structures used for representations of graphs are matrices of adjacency (either vertices or edges), incidence matrices, and lists of adjacency (either vertices or edges). We denote R any of these data structures.

Let $\mathsf{G}(n,m)$, $\overrightarrow{\mathsf{G}}^{or}(n,m)$, and $\overrightarrow{\mathsf{G}}^{dir}(n,m)$ be the set of all graphs, orgraphs and directed graphs $G = (V,E)$ such that $V = \{1,\ldots,n\}$ and $|E| = m$, $\mathsf{R}(G)$ be the representation of $G \in \mathsf{X}(n,m)$ ($\mathsf{X} \in \{\mathsf{G}, \overrightarrow{\mathsf{G}}^{or}, \overrightarrow{\mathsf{G}}^{dir}\}$) by the data structure R, and $v(\mathsf{R}(G))$ be the size of memory necessary for the representation $\mathsf{R}(G)$. We set

$$v(\mathsf{R}, \mathsf{X}, n, m) = \max_{G \in \mathsf{X}(n,m)} V(\mathsf{R}(G)) \ (\mathsf{X} \in \{\mathsf{G}, \overrightarrow{\mathsf{G}}^{or}, \overrightarrow{\mathsf{G}}^{dir}\}).$$

Due to the traditional approach in the Algorithms Theory, we will deal with asymptotic space complexity $V(\mathsf{R}, \mathsf{X}, n, m)$ of $v(\mathsf{R}, \mathsf{X}, n, m)$.

It is well-known that for all $\mathsf{X} \in \{\mathsf{G}, \overrightarrow{\mathsf{G}}^{or}, \overrightarrow{\mathsf{G}}^{dir}\}$:

$V(\mathsf{R}, \mathsf{X}, n, m) = O(n^2)$ $(n \to \infty)$, if R is the vertices-adjacency matrix;

$V(\mathsf{R}, \mathsf{X}, n, m) = O(m^2)$ $(m, n \to \infty)$, if R is the edges-adjacency matrix;

$V(\mathsf{R}, \mathsf{X}, n, m) = O(mn)$ $(m, n \to \infty)$, if R is the incidence matrix;

$V(\mathsf{R}, \mathsf{X}, n, m) = O(\max\{m, n\})$ $(m, n \to \infty)$, if R are the vertices-adjacency lists;

$V(\mathsf{R}, \mathsf{X}, n, m) = O(m \cdot \max\{1, \min\{m, n\}\})$ $(m, n \to \infty)$, if R are the edges-adjacency lists.

Since X is a dummy variable in $V(\mathsf{R}, \mathsf{X}, n, m)$, we will write $V(\mathsf{R}, n, m)$.

Let $v_{\mathcal{A}}(\mathsf{R}, \mathsf{X}, G)$ $(\mathsf{X} \in \{\mathsf{G}, \overrightarrow{\mathsf{G}}^{or}, \overrightarrow{\mathsf{G}}^{dir}\})$ be the size of memory necessary for an algorithm $\mathcal{A}$ to carry out the given processing of the data structure $\mathsf{R}(G)$ $(G \in \mathsf{X}(n, m))$ and

$$v_{\mathcal{A}}(\mathsf{R}, \mathsf{X}, n, m) = \max_{G \in \mathsf{X}(n,m)} v_{\mathcal{A}}(\mathsf{R}, \mathsf{X}, G).$$

In what follows, we will deal with asymptotic space complexity $V_{\mathcal{A}}(\mathsf{R}, \mathsf{X}, n, m)$ of $v_{\mathcal{A}}(\mathsf{R}, \mathsf{X}, n, m)$.

2.2 Some Classes of Problems in Graph Theory

Let's distinguish the following set of algorithms on graphs.

Definition 1. *An algorithm $\mathcal{A}$ is called a local algorithm on the set $\mathsf{X}(n, m)$ $(\mathsf{X} \in \{\mathsf{G}, \overrightarrow{\mathsf{G}}^{or}, \overrightarrow{\mathsf{G}}^{dir}\})$ and the data structure R, if*

$$V_{\mathcal{A}}(R, X, n, m) = V(R, n, m) \ \ (n \to \infty, m \to \infty). \tag{1}$$

Example 1. Let $\mathsf{X} = \mathsf{G}$ and R be the representation of a graph $G \in \mathsf{G}(n, m)$ by the vertices-adjacency matrix. It can be proved that the problem of checking the validity of the property "to be a connected graph" can be solved by some local algorithm.

It is evident that the validity of equality (1) can significantly depend on the law of the growth $m \to \infty$. To avoid this dependence, we will act as follows.

For any fixed positive integer n we set

$$\mathsf{X}(n) = \bigcup_m \mathsf{X}(n, m), \tag{2}$$

$$V_{\mathcal{A}}(\mathsf{R}, \mathsf{X}, n) = \max_m V_{\mathcal{A}}(\mathsf{R}, \mathsf{X}, n, m), \tag{3}$$

and

$$V(\mathsf{R}, n) = \max_m V(\mathsf{R}, n, m), \tag{4}$$

where union and maximum are over all admissible values of m.

On the base of formulae (2)-(4) we get the following definition.

Definition 2. *An algorithm $\mathcal{A}$ is called a local algorithm on the set $X(n)$ ($X \in \{G, \overrightarrow{G}^{or}, \overrightarrow{G}^{dir}\}$) and the data structure R for any admissible law of the growth $m \to \infty$, if*

$$V_{\mathcal{A}}(R, X, n) = V(R, n) \ (n \to \infty).$$

Example 2. Let $X = G$. It can be proved that following three statements are true:

1. Let R be the representation of a graph $G \in G(n, m)$ either by the vertices-adjacency matrix, or by the vertices-adjacency lists. Then the problem of checking the validity of the property "the given two graphs are isomorphic" can be solved by some algorithm that is a local algorithm for any admissible law of the growth $m \to \infty$.

2. Let R be the representation of a graph $G \in G(n, m)$ either by the vertices-adjacency matrix, or by the edges-adjacency matrix. Then the problem of checking the validity of the property "some sub-graphs of the given graph are the triangles" can be solved by some algorithm that is a local algorithm for any admissible law of the growth $m \to \infty$.

3. Let R be the representation of a graph $G \in G(n, m)$ by the vertices-adjacency matrix. Then the problem of checking the validity of the property "to be a bipartite graph", and each of the problems of computing for the given graph $G \in G(n, m)$ the radius, the diameter, the center, and the connected components can be solved by some algorithm that is a local algorithm for any admissible law of the growth $m \to \infty$.

On the base of Definition 2, we can distinguish the following set of Graph Theory problems.

Definition 3. *The set $Lcl(R, X)$ ($X \in \{G, \overrightarrow{G}^{or}, \overrightarrow{G}^{dir}\}$) is the set of all Graph Theory problems P, such that the problem P can be resolved by some local algorithm on the set $X(n)$ and the data structure R for any admissible law of the growth $m \to \infty$.*

Example 3. 1. Let A_n ($n = 1, 2, \ldots$) be the algebraic system with the basic set

$$G(n) = \bigcup_{m \geq 0} G(n.m),$$

the set of operations

$$F_{op}^{A_n} = \{\backslash, \cup, \cap, \oplus, \neg\},$$

and the set of relations

$$F_{rel}^{A_n} = \{=, \subset\}.$$

It can be proved that in the algebraic system A_n the problems of implementation of any operation $f \in F_{op}^{A_n}$ and checking the validity of any relation $\rho \in F_{rel}^{A_n}$ are elements of the set $Lcl(R, G)$ for any data structure R.

2. Let B be the algebraic system with the basic set

$$T = \bigcup_{n=1}^{\infty} \mathsf{G}(n),$$

the set of operations

$$F_{op}^B = \bigcup_{n=1}^{\infty} F_{op}^{A_n} \cup \{\circ, \rightarrow, \bullet, *, \wedge, \vee\},$$

and the set of relations

$$F_{rel}^B = \bigcup_{n=1}^{\infty} F_{rel}^{A_n} \cup \{\leq, \lhd\}.$$

It can be proved that for the algebraic system B the following two statements are true:

1. The problems of implementation of any operation $f \in \{\circ, \rightarrow\}$ and checking the validity of any relation $\rho \in \{\leq, \lhd\}$ are elements of the set $\mathsf{Lcl}(\mathsf{R}, \mathsf{G})$ for any data structure R.

2. For each operation $f \in \{\bullet, *, \wedge, \vee\}$ the problem of its implementation is not an element of the set $\mathsf{Lcl}(\mathsf{R}, \mathsf{G})$ for any data structure R.

Remark 1. The operations $\vee$ and $\wedge$ are often used in the design and analysis of algorithms. In particular, for parallel and concurrent computing. Indeed, let the graphs $G_1, G_2 \in T$ be the models of the two given sub-problems. The vertices present the separate stages of solving these problems, and the edges specify what stages are the neighbors.

The graph $G_1 \wedge G_2$ describes the situation when only simultaneous advance on stages of both sub-problems is admissible.

The graph $G_1 \vee G_2$ describes the situation when advance on stages at least of one of the sub-problems is admissible.

For a wide class of Graph Theory problems the size of the solution significantly exceeds the size of the input data. Due to this situation, it is reasonable to distinguish the following set of Graph Theory problems.

Definition 4. *The set $\mathsf{P\text{-}Work\text{-}Space}(R, X)$ $(X \in \{G, \overrightarrow{G}^{or}, \overrightarrow{G}^{dir}\})$ is the set of all Graph Theory problems P, such that the problem P can be resolved on the set $X(n)$ and the data structure R for any admissible law of the growth $m \to \infty$ by some algorithm $\mathcal{A}$ that satisfies to the following two conditions:*

Condition 1. The algorithm A operates on the memory size $V(R, n)$ $(n \to \infty)$.

Condition 2. The algorithm A explores as an output channel some additional memory of the size that is a polynomial of $V(R, n)$ to generate the solution sequentially, one fragment after another.

We denote $\mathsf{Work\text{-}Lcl}(\mathsf{R}, \mathsf{X})$ the set of all problems $P \in \mathsf{P\text{-}Work\text{-}Space}(\mathsf{R}, \mathsf{X})$, such that the size of additional memory, pointed in Condition 2 of Definition 4 is a linear polynomial of $V(\mathsf{R}, n)$. It can be proved that for any $\mathsf{X} \in \{\mathsf{G}, \overrightarrow{\mathsf{G}}^{or}, \overrightarrow{\mathsf{G}}^{dir}\}$ and any data structure R the following strict inclusions hold

$$\mathsf{Lcl}(\mathsf{R}, \mathsf{X}) \subset \mathsf{Work\text{-}Lcl}(\mathsf{R}, \mathsf{X}) \subset \mathsf{P\text{-}Work\text{-}Space}(\mathsf{R}, \mathsf{X}).$$

Example 4. It can be proved that the following three statements are true:

1. Let R be the representation either by the vertices-adjacency matrix, or by the vertices-adjacency lists. Then the problem of the design of the edge-to-vertex dual graph is an element of the set $\mathsf{Work\text{-}Lcl}(\mathsf{R}, \mathsf{G}) \backslash \mathsf{Lcl}(\mathsf{R}, \mathsf{G})$.

2. The problem of the design of some the longest path between the two given vertices in a graph $G \in \mathsf{G}(n)$, and consequently, the problem of the design of the set of all longest paths between the two given vertices in a graph $G \in \mathsf{G}(n)$, are elements of the set $\mathsf{Work\text{-}Lcl}(\mathsf{R}, \mathsf{X}) \backslash \mathsf{Lcl}(\mathsf{R}, \mathsf{X})$ for any data structure R.

3. In the algebraic system B (see example 3.2), the problem of implementation of any operation $f \in \{\bullet, *, \wedge, \vee\}$ is an element of the set $\mathsf{Work\text{-}Lcl}(\mathsf{R}, \mathsf{G}) \backslash \mathsf{Lcl}(\mathsf{R}, \mathsf{G})$ for any data structure R.

3 Analysis of Graphs Representations

Similarly to the algebraic systems A_n $(n = 1, 2, \ldots)$ and B (see example 3), there can be defined the algebraic system $\overrightarrow{A}_n^{or}$ $(n = 1, 2, \ldots)$ with the basic set $\overrightarrow{\mathsf{G}}^{or}(n)$, the algebraic system $\overrightarrow{B}^{or}$ with the basic set

$$\overrightarrow{T}^{or} = \bigcup_{n=1}^{\infty} \overrightarrow{\mathsf{G}}^{or}(n),$$

the algebraic system $\overrightarrow{A}_n^{dir}$ $(n = 1, 2, \ldots)$ with the basic set $\overrightarrow{\mathsf{G}}^{dir}(n)$, and the algebraic system $\overrightarrow{B}^{dir}$ with the basic set

$$\overrightarrow{T}^{dir} = \bigcup_{n=1}^{\infty} \overrightarrow{\mathsf{G}}^{dir}(n).$$

In Subsection 2.2, when we were speaking about complexity, we meant "complexity in the worst case". From the theoretic and applied point of view, both, a significant role also play "the average-case complexity" and "complexity for almost all input data".

The following factors form the strong base for the application of this approach for the detailed analysis of complexity for problems formulated in terms of the algebraic systems Y_n $(Y \in \{A, \overrightarrow{A}^{or}, \overrightarrow{A}^{dir}\})$ and Z $(Z \in \{B, \overrightarrow{B}^{or}, \overrightarrow{B}^{dir}\})$.

On the base of standard methods used for computing the mathematical expectation of the random variable defined on a finite set, the following theorem can be proved.

Theorem 1. *Let $X \in \{G, \overrightarrow{G}^{or}, \overrightarrow{G}^{dir}\}$. For chosen randomly element of the set $X(n)$ the average number of edges equals to:*

1) $0.25n(n-1)$, if $X = G$;

2) $\frac{1}{3}n(n-1)$, if $X = \overrightarrow{G}^{or}$;

3) $0.5n(n-1)$, if $X = \overrightarrow{G}^{dir}$.

Proceeding from this theorem, the following two propositions can be proved.

Proposition 1. *Let the elements of the basic set $X(n)$ ($X \in \{G, \overrightarrow{G}^{or}, \overrightarrow{G}^{dir}\}$) be chosen randomly. Then in the relevant algebraic system Y_n ($Y \in \{A, \overrightarrow{A}^{or}, \overrightarrow{A}^{dir}\}$) the average time for the implementation of any operation, as well as the average time for checking validity of any relation, is asymptotically the same for representations of elements of the basic set $X(n)$ by the vertices-adjacency lists and by the vertices-adjacency matrices.*

Proposition 2. *Let the elements of the basic set $X(n)$ ($X \in \{G, \overrightarrow{G}^{or}, \overrightarrow{G}^{dir}\}$) be chosen randomly. Then in the relevant algebraic system Y_n ($Y \in \{A, \overrightarrow{A}^{or}, \overrightarrow{A}^{dir}\}$) the average time for the implementation of any operation, as well as the average time for checking validity of any relation, for representations of elements of the basic set $X(n)$ by the edge-adjacency lists is less asymptotically than the appropriate time for representations of elements of the basic set $X(n)$ by the edges-adjacency matrices.*

On the base of standard methods used for computing the variance of the random variable defined on a finite set, and using Chebyshev's inequality, the following theorem can be proved.

Theorem 2. *Let $X \in \{G, \overrightarrow{G}^{or}, \overrightarrow{G}^{dir}\}$. For almost all elements of the set $X(n)$ ($n \to \infty$) the number m of edges satisfy to the following asymptotic equality*

$$m = \Theta(n^2) \ (n \to \infty).$$

Proceeding from this theorem, the following two propositions can be proved.

Proposition 3. *In the algebraic system Y_n ($Y \in \{A, \overrightarrow{A}^{or}, \overrightarrow{A}^{dir}\}$) for almost all elements of the relevant basic set $X(n)$ ($X \in \{G, \overrightarrow{G}^{or}, \overrightarrow{G}^{dir}\}$) the time for the implementation of any operation, as well as the time for checking validity of any relation, is asymptotically the same for representations of elements of the basic set $X(n)$ by the vertices-adjacency lists and by the vertices-adjacency matrices.*

Proposition 4. *In the algebraic system Y_n ($Y \in \{A, \overrightarrow{A}^{or}, \overrightarrow{A}^{dir}\}$) for almost all elements of the relevant basic set $X(n)$ ($X \in \{G, \overrightarrow{G}^{or}, \overrightarrow{G}^{dir}\}$) the time for the implementation of any operation, as well as the time checking validity of any relation, for representations of elements of the basic set $X(n)$ by the edge-adjacency lists is asymptotically less than the appropriate time for representations of elements of the basic set $X(n)$ by the edges-adjacency matrices.*

4 Some Remark About the Set Work-Lcl(R, X)\Lcl(R, X)

One of the main reason owing to which the considerable number of problems of Graph Theory are elements of the set Work-Lcl(R, X)\Lcl(R, X) is based on the following factor. The problem of the design of any object is an element of the set Lcl(R, X), but the number of objects, which are required to be designed as the solution of the analysed problem, is an exponent or some sub-exponent of n and m. This situation can be illustrated as follows.

Example 5. It can be proved that for all $\mathsf{X} \in \{\mathsf{G}, \overrightarrow{\mathsf{G}}^{or}, \overrightarrow{\mathsf{G}}^{dir}\}$ and for all data structures R the following problems are elements of the set Lcl(R, X):

1. The problem of the design of some the shortest path between the given two vertices in $G \in \mathsf{X}(n)$.

2. The problem of the design of some Hamiltonian path between the given two vertices in $G \in \mathsf{X}(n)$.

3. The problem of the design of some Hamiltonian cycle in $G \in \mathsf{X}(n)$.

4. The problem of the design of some cycle that visits the given vertex in $G \in \mathsf{X}(n)$.

5. The problem of the design of some spanning tree in $G \in \mathsf{X}(n)$.

6. The problem of the design of some minimum spanning tree in $G \in \mathsf{X}(n)$.

On the base of estimation the number of objects which can be designed in each of listed above case, it can be proved that for all $\mathsf{X} \in \{\mathsf{G}, \overrightarrow{\mathsf{G}}^{or}, \overrightarrow{\mathsf{G}}^{dir}\}$ and for all data structures R the following problems are elements of the set Work-Lcl(R, X)\Lcl(R, X):

1. The problem of the design of the set of all shortest paths between the two given vertices in $G \in \mathsf{X}(n)$.

2. The problem of the design of the set of all Hamiltonian path between the given two vertices in $G \in \mathsf{X}(n)$.

3. The problem of the design of the set of all Hamiltonian cycles in $G \in \mathsf{X}(n)$.

4. The problem of the design of the set of all cycles that visit the given vertex in $G \in \mathsf{X}(n)$.

5. The problem of the design of the set of all spanning trees in $G \in \mathsf{X}(n)$.

6. The problem of the design of the set of all minimum spanning trees in $G \in \mathsf{X}(n)$.

5 Out of the Set Work-Lcl(R, X)

Analysis of problems pointed in the Section 4, show that each of them can be solved by backtracking with linear space complexity. Unfortunately, there is a wide class of problems of Discrete Mathematics and its applications, such that the following two conditions hold:

1. Searching is the only known method of the solution of the given problem.

2. At implementation of searching there is an essential growth of lengths and the number of the designed and analyzed sequences of objects, both (just this factor results in exponential space complexity of searching).

All these problems formulated in terms of Graph Theory are the problems that are out of the set Work-Lcl(R, X).

An important non-trivial subset of the above-pointed problems of Discrete Mathematics consists of the problems that can be reduced to the design of some strategy for walks of special type on some graph. This subset includes in itself the problems, at least, of the following three types:

1. The problems that can be reduced to the design some unconditional, adaptive or cooperative strategy for some walk on the given graph, intended to identify the vertices covered by blots. In particular, to these problems can be reduced problems of identification of states for finite automata. Interpretation of these adaptive and cooperative strategies in terms of automata-experimenters demonstrates high complexity for computing the values of predicates defined on graphs by some finite automaton or some interacting group of finite automata.

2. The problems that are connected with the design of the supervisor intended to carry out adaptive control for discrete events systems presented by finite automata models.

3. The problems connected with the design of the winning strategies for considerable number of two persons games on graphs.

6   Conclusions

In the given paper we have analyzed two sets of Graph Theory problems. The first set consists of all Graph Theory problems that can be resolved by algorithms with linear space complexity. The second set consists of all Graph Theory problems, for which the size of the solution essentially exceeds the size of the input data, but there exists some algorithm that operates on space linear relative to the size of the input data, and this algorithm uses some additional memory of the same size, intended to generate the solution sequentially, one fragment after another.

It has been illustrated that these sets consist of sufficiently wide class of Graph Theory problems.

The carried-out analysis of sufficiently powerful algebraic systems on graphs gave the possibility to establish a number of estimates in terms "in average" and "for almost all objects".

Presented in the given paper results form some strong base for research of the structure of the set P-Work-Space(R, X), for analysis complexity, accuracy and efficiency of local search strategies on graphs, for investigation problems of design different strategies for walks on graphs, etc.

References

1. Harary, F.: Graph theory. Addison-Wesley Publishing Company, Inc., Reading, MA, USA (1969)
2. Aho, A.V., Hopcroft, J.E., and Ullman, J.D.: The design and analysis of computer algorithm. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1975)
3. Bollobas, B.: Modern graph theory. Springer-Verlag, New York, Berlin, Heidelberg (1998)