

Gainanov Damir N., Mladenovic NENAD, Rasskazova V. A.

Maximum independent set in planning freight railway transportation

© The Author(s) 2018. Published by Higher Education Press. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0>)

Abstract This work is devoted to the problem of planning of freight railway transportation. We define a special conflict graph on the basis of a set of acceptable train routes. The investigation aims to solve the classical combinatorial optimization problem in relation to the maximum independent set of vertices in undirected graphs. The level representation of the graph and its tree are introduced. With use of these constructions, the lower and upper bounds for the number of vertices in the maximum independent set are obtained.

Keywords independent set, algorithm, planning of transportation, two-sided estimate

1 Introduction

The problems that arise from managing transport processes have a special scientific and practical actuality in view of the importance of their effective solution for economic development and ensuring the connections of territories.

Planning of freight railway transportation was investigated in the context of scheduling theory in (Burdett and Kozan, 2010; Gholami et al., 2012; Gholami and Sotskov,

2014; Akimova et al., 2015; Gafarov et al., 2015; Gainanov et al., 2016; Lazarev et al., 2018). In this case, the schedules necessary for execution were built directly under the current conditions of the availability of network resources. Pei et al. (2017a, b) also developed several hybrid algorithms in scheduling, where a number of effective novel operators of variable neighborhood search was proposed. The computational results showed the good performance of these algorithms.

The development of railway network infrastructure is conservative and requires large-scale investments. By contrast, the demand for freight railway transportation is characterized by significant dynamics in terms of total traffic volume and the distribution of traffic across various sections of railway networks. Under these conditions, infrastructure resources or railway network capacity is deficient. In this regard, developing methods of planning of freight railway transportation, which will enable transportation without significant modernization of railway infrastructure, is becoming increasingly important.

In this work, the problem under investigation obtains a fundamentally new interpretation in terms of the classical combinatorial optimization problem. The main concept of the proposed approach is to form a set of potential train routes whose executions do not contradict each other. Thus, any subset of routes from this set can serve as a valid schedule. In terms of graph theory, this problem aims to explore the maximum independent set of vertices in undirected graphs.

Identifying the maximum independent set is a classical NP-hard problem. In this regard, developing algorithms that are effective in terms of computational complexity is actual. In other studies (Dasgupta et al., 2006; Korte and Vygen, 2008; Skiena, 2008), general approaches to developing exact, approximate, and heuristic algorithms were explored. In another set of studies (Duarte A et al., 2015; Todosijevic and Mladenovic, 2016; Hansen et al., 2017), significant results were obtained in the field of development of metaheuristic algorithms, such as solving

Received March 27, 2018; accepted June 30, 2018

Gainanov Damir N.
Ural Federal University, Ekaterinburg, Russia

Mladenovic NENAD
Serbian Academy of Sciences and Arts, Belgrade, Serbia

Rasskazova V. A. (✉)
Moscow Aviation Institute, Moscow, Russia
E-mail: varvara.rasskazova@mail.ru

This work is partially supported by the framework of grant number BR05236839. “Development of information technologies and systems for stimulation of personality’s sustainable development as one of the bases of development of digital Kazakhstan.”

applied problems of production control and transport processes.

In this research, we propose an approach in which two-sided estimates can be obtained for the number of vertices in the maximum independent set. This approach mainly aims to evaluate the accuracy of approximate solutions, which can be obtained by any available method, such as heuristic and metaheuristic algorithms. In addition, implementing the approach for different root vertices means that an exact solution can be expected when the lower and upper bounds coincide.

The paper is organized as follows. Section 1 presents a general formulation of the problem of planning of railway freight transportation. We also formalize a method for reducing the applied problem to finding the maximum independent set. Section 2 describes the approximate algorithm and its modification for the class of trees. In Section 3, special graph constructions of the level representation and its tree are introduced. These constructions underlie the proposed approach. Section 4 provides the lower and upper bounds for the number of vertices in the maximum independent set. An approximate algorithm and its modification for a class of trees are used to obtain these estimates. Section 6 presents the conclusion, an overview of the main results, and directions for further research.

2 Planning of freight railway transportation

In this section, the formulation of the private problem of planning of freight railway transportation is presented. In general, a set of potential train routes is required for the execution of a specific transportation plan. However, the selected routes cannot often be performed due to their intersections at stations and time of movement, which entail significant economic and time costs to execute the plan. In this connection, the problem of forming such a set of potential routes, that is, selecting any of the routes does not contradict the practical implementation, becomes actuality.

Consider the directed graph of the railway transport network

$$\vec{\Gamma} = (S, E),$$

where

$$S = \{s_1, s_2, \dots, s_n\}$$

is a set of stations and

$$E \subseteq \bigcup_{i,j} (s_i, s_j)$$

is a set of directed railroads that connect stations.

Definition 2.1. The normative thread of the train traffic schedule is a sequence of the form

$$n : \left(s_{i_1}(n), s_{i_2}(n), Nom_{i_1 i_2}(n), t_{i_1 i_2}^{dep}(n), t_{i_1 i_2}^{arr}(n), g_{i_1 i_2}(n, t) \right), \\ \left(s_{i_2}(n), s_{i_3}(n), Nom_{i_2 i_3}(n), t_{i_2 i_3}^{dep}(n), t_{i_2 i_3}^{arr}(n), g_{i_2 i_3}(n, t) \right), \dots, \\ \dots, \left(s_{i_{k-1}}(n), s_{i_k}(n), Nom_{i_{k-1} i_k}(n), t_{i_{k-1} i_k}^{dep}(n), t_{i_{k-1} i_k}^{arr}(n), g_{i_{k-1} i_k}(n, t) \right),$$

where $(s_{i_{j-1}}(n), s_{i_j}(n)) \in E$ and $Nom_{i_{j-1} i_j}(n), t_{i_{j-1} i_j}^{dep}(n), g_{i_{j-1} i_j}(n, t)$ are given path number, arrival time, departure time, and schedule of movement on the railroad $(s_{i_{j-1}}(n), s_{i_j}(n))$, respectively.

$$s^{beg}(n) = s_{i_1}(n), t^{beg}(n) = t_{i_1 i_2}^{dep}(n),$$

$$s^{fin}(n) = s_{i_k}(n), t^{fin}(n) = t_{i_{k-1} i_k}^{arr}(n).$$

Any normative thread represents a potential train route, which enables the execution of a certain transportation from the given plan.

Definition 2.2. Let n_1, n_2 be such threads that a directed railroad enters the sequence of each thread, and enable them to pass this railroad in the same or opposite directions yet remain on the same path.

$$\begin{cases} (s_i, s_j) = (s_k(n_1), s_{k+1}(n_1)) = (s_m(n_2), s_{m+1}(n_2)) \\ Nom_{ij}(n_1) = Nom_{ij}(n_2), \end{cases}$$

or

$$\begin{cases} (s_i, s_j) = (s_k(n_1), s_{k+1}(n_1)) = (s_{m+1}(n_2), s_m(n_2)) \\ Nom_{ij}(n_1) = Nom_{ji}(n_2). \end{cases}$$

Then, if time moment t exists from the planning period $[T_0, T]$ such that the distance between the trains following the threads n_1 and n_2 at this moment becomes less than the permissible distance d ; that is,

$$|g_{ij}(n_1, t) - g_{ij}(n_2, t)| \leq d,$$

or

$$|g_{ij}(n_1, t) - (l_{ij} - g_{ji}(n_2, t))| \leq d,$$

where l_{ij} is the given length ($l_{ij} = l_{ji}$) of the railroad (s_i, s_j) , then normative threads n_1 and n_2 have unidirectional or multidirectional conflict, respectively.

Let the set of all normative threads

$$\mathcal{N} = \{n_i\}, i = 1, 2, \dots,$$

be ordered with respect to the beginning time $t^{beg}(n)$ and finishing time $t^{fin}(n)$ of the movement by threads.

The following construction is a mathematical model of the applied problem under consideration, within the

framework of which the research can be reduced to solving the classical problem of combinatorial optimization.

Definition 2.3. An undirected conflict graph is a graph whose vertices are normative threads. These vertices are connected by edge if and only if a unidirectional or multidirectional conflict exists for the respective normative threads.

$$G = (\mathcal{N}, \varepsilon) : \{n_i, n_j\} \in \varepsilon \Leftrightarrow \text{there exists the conflict.}$$

The problem of planning of freight railway transportation is that of selecting the maximum conflict-free subset from an entire set of normative threads. Then, any subset of conflict-free normative threads can serve as an acceptable schedule for the organization of transportation.

Problem 2.1. For a given set of normative threads, the maximum subset has to be identified. The conflict graph generated by this subset must be empty; that is, it has no edges.

$$\mathcal{N}' \subseteq \mathcal{N}.$$

$$|\mathcal{N}'| = \max \{|\mathcal{N}''| : G(\mathcal{N}'') = (\mathcal{N}'', \emptyset)\}.$$

In terms of graph theory, the abovementioned problem is that of the maximum independent set of vertices in an undirected graph. In the following chapters, an approach is developed, within which estimates can be obtained for the number of vertices in the maximum independent set. Then, the solutions found by approximate methods can be characterized by their degree of closeness to the optimal solution. Conflict graphs that arise during railway operation are typically highly sparse. However, when an exact solution cannot be found, the proposed approach to estimating the approximate solution becomes rather actuality.

3 Algorithms for finding the maximum independent set

A previous study (Gainanov and Rasskazova, 2016) provided an algorithm for inferring a monotone Boolean function that is generated by an undirected graph; this algorithm can be interpreted in terms of independent sets of vertices. The following are necessary definitions.

Definition 3.1. For an integer $k \in [n-1]$, we call a vertex $v \in V$ of the graph $G = (V, \varepsilon)$ a k -vertex, if $|\mathcal{N}(v)| = k$ and the induced subgraph $G(\mathcal{N}(v))$ of the graph G is complete.

The algorithm described in the previous study (Gainanov and Rasskazova, 2016) is based on the abovementioned concept of k -vertices. That is, k -vertices are included at each step in the independent set being constructed. If the algorithm converges (that is, the current set of vertices of the graph becomes empty at certain steps), then the independent set found is the maximum.

Following the source, we denote the neighborhood of vertex v in the induced subgraph $G\langle V \rangle$ using $\mathcal{N}(v, V) \subset V$.

Algorithm $\mathcal{A}(G, V_0)$: Finding the maximum independent set of vertices in the undirected graph.

Input data: $G = (V_0, \varepsilon)$

Output data: V_0, S - independent set of vertices

$$S \leftarrow \{ \}$$

For $v \in V_0$ **do**

If v is a k -vertex in the subgraph $G\langle V_0 \rangle$ **then**

$$S \leftarrow S \cup \{v\}$$

$$V_0 \leftarrow V_0 - \{v\} - \mathcal{N}(v, V_0)$$

$$\mathcal{A}(G, V_0)$$

If $V_0 = \{ \}$, then the independent set found is the maximum. This condition is guaranteed if the original graph is a tree.

Indeed, k -vertices consistently exist in the tree, whereas the current set of vertices is not empty. These vertices are tree leaves (hanging vertices) or isolated vertices at certain steps of the algorithm. Thus, Algorithm $\mathcal{A}(G, V_0)$ can be modified for a special class.

Algorithm $IndSetTree(G)$: Finding the maximum independent set of vertices in the tree.

Input data: tree $G = (V, E)$

Output data: maximum independent set of vertices S

$$V_0 \leftarrow V$$

$$S \leftarrow \{ \}$$

While $V_0 = \{ \}$ **do**

For all $v \in V_0$ **do**

If v - is a leaf or isolated vertex **then**

$$S \leftarrow S \cup \{v\}$$

$$V_0 \leftarrow V_0 - \{v\} - \mathcal{N}(v, V_0)$$

Isolated vertices can appear after excluding from the current parent set the vertex selected for inclusion in the maximum independent set being constructed.

Consider an example of the implementation of the algorithm $IndSetTree(G)$.

Example 3.1 Let the tree $G = (V, E)$ be given (Fig. 1).

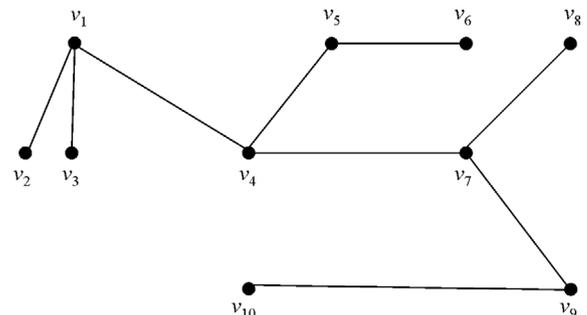


Fig. 1 Tree steps of the algorithm

1. $S = \{ \}$
 $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$
 v_1 — is not a leaf or an isolated vertex
 v_2 — is a leaf
 $S = \{v_2\}$
 $V = \{v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$
2. v_3 — is an isolated vertex
 $S = \{v_2, v_3\}$
 $V = \{v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$
3. v_4, v_5 — is not a leaf or an isolated vertex
 v_6 — is a leaf
 $S = \{v_2, v_3, v_6\}$
 $V = \{v_4, v_7, v_8, v_9, v_{10}\}$
4. v_4, v_7 — is not a leaf or an isolated vertex
 v_8 — is a leaf
 $S = \{v_2, v_3, v_6, v_8\}$
 $V = \{v_4, v_9, v_{10}\}$
5. v_4 — is an isolated vertex
 $S = \{v_2, v_3, v_4, v_6, v_8\}$
 $V = \{v_9, v_{10}\}$
6. v_9 — is a leaf
 $S = \{v_2, v_3, v_4, v_6, v_8, v_9\}$
 $V = \{ \}$

End of the algorithm

At each step, either a leaf or an isolated vertex is included in the independent set. That is, the k -vertices for $k = 1$ and $k = 0$. Thus, the independent set $S = \{v_2, v_3, v_4, v_6, v_8, v_9\}$ is the maximum.

The problem of the maximum independent set of vertices in a tree can be effectively solved using dynamic programming. However, structural differences remain in these approaches.

Let us consider the case when the current set of vertices at certain steps of Algorithm $\mathcal{A}(G, V_0)$ does not have a k -vertex.

Definition 3.2. For integers $k, m \in [n-1]$, we call a vertex $v \in V$ of the graph $G = (V, E)$ a (k, m) -vertex, if $k = |\mathcal{N}(v)|$ and $m = C_k^2 - \left| E \cap \binom{\mathcal{N}(v)}{2} \right|$.

In other words, a vertex is called a (k, m) -vertex if it has degree k and m edges are loosed in the neighborhood to be a complete induced subgraph.

Proposition 3.1. Let two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ have the same set of vertices $V_1 = V_2$ such that $E_1 \subseteq E_2$. Then, the following inclusion holds:

$$S_{\max}(G_2) \subseteq S(G_1),$$

where $S(G)$ is the set of all independent sets of the graph G and $S_{\max}(G)$ is the set of all maximum independent sets of the graph G .

The relation between the maximum independent sets of two undirected graphs follows from Proposition 3.1. Let us define the quantity $\max S(G) = |S|$, where $S \in S_{\max}(G)$. That is, the cardinality of the maximum independent set of graph G .

Corollary 3.1. Let $G_1 = (V, \varepsilon_1)$ and $G_2 = (V, \varepsilon_2)$ be graphs such that $\varepsilon_1 \subseteq \varepsilon_2$. Then,

$$\max S(G_1) \geq \max S(G_2).$$

Thus, if no applicants are to be included in the independent set under construction at certain steps of Algorithm $\mathcal{A}(G, V_0)$, then the neighborhood of any vertex for complete subgraphs can be artificially increased by consideration of the relation between the initial and new graphs. The following statement shows the details of the relation established in Corollary 1.1.

Proposition 3.2. Let $G = (V(G), E(G))$ be a graph in which vertices v_i and v_j are not adjacent. Let $G' = (V(G'), E(G'))$ be a graph in which the set of vertices is the same $V(G') = V(G)$ and that of edges is such that $E(G') = E(G) \cup \{(v_i, v_j)\}$. Hence, vertices v_i and v_j are adjacent in this graph. Then,

$$\max S(G) \geq \max S(G') \geq \max S(G) - 1.$$

The case where the neighborhood of any vertex is increased by unique edges can be continued for a larger number of edges.

Corollary 3.2. For graph $G = (V, E(G))$ let $e_1, \dots, e_k \notin E(G)$ be a subfamily of k vertex pairs that are not edges of graph G . In addition, let graph $G' = (V, E(G'))$, where $E(G') = E(G) \cup \{e_1, \dots, e_k\}$. Then,

$$\max S(G) \geq \max S(G') \geq \max S(G) - k.$$

According to Corollary 3.2, at each step of Algorithm $\mathcal{A}(G, V_0)$, an arbitrary vertex of the graph can be included into the independent set under construction. With consideration of the value of the parameter m of this vertex, the value of the deviation of the obtained approximate solution from the optimal one can be calculated. Let us consider the following approximate algorithm.

Algorithm $\mathcal{B}(G, V)$: Finding the independent set of vertices in the undirected graph.

Input data: $G = (V, E)$

Output data: S, Est - the estimate of the deviation of the approximate solution

$$S \leftarrow \{ \}; V_0 \leftarrow V; Est \leftarrow 0$$

While $V_0 \neq \{ \}$ **do**

For $v \in V_0$ **do**

$$k[v] \leftarrow \text{Calculate}(k, v)$$

$$m[v] \leftarrow \text{Calculate}(m, v)$$

$$v_0 \leftarrow \text{MinMaxParam}(v)$$

$$S \leftarrow S \cup \{v_0\}$$

$$Est \leftarrow Est + m[v_0]$$

$$V_0 \leftarrow V_0 - \{v_0\} - \mathcal{N}(v_0, V_0)$$

Algorithm $\mathcal{B}(G, V)$ operates with several functions. Function $\text{Calculate}(k, v)$ is used to calculate the values of parameter k for each vertex v . Function $\text{Calculate}(m, v)$ is similarly used for calculating the values of parameter m for each vertex v . Function $\text{MinMaxParam}(v)$ executes the choice of the vertex with minimal values of parameter m and choice of the vertex with the maximum value of parameter k among all previously selected vertices. If the unique vertex exists with the minimum value of parameter m , then selecting the vertex with the maximum value of parameter k is unnecessary. Thus, following Proposition 3.2, we obtain the next feature that is related to Algorithm $\mathcal{B}(G, V)$.

Proposition 3.3. Let S and Est be found by $\mathcal{B}(G, V)$. Then

$$|S| \geq |\max S(G)| - Est.$$

Let us consider an example of using Algorithm $\mathcal{B}(G, V)$ for finding an independent set of the arbitrary undirected graph. We also demonstrate the accumulation of the deviation estimate of the approximate solution from the optimal one.

Example 3.2. Let the undirected graph $G = (V, E)$ be given by the following list of adjacency vertices.

$v_1 : v_2, v_3, v_4, v_6, v_8, v_9$	$v_{12} : v_2, v_3, v_4, v_6, v_{11}, v_{17}$
$v_2 : v_1, v_3, v_4, v_6, v_{12}$	$v_{13} : v_{11}, v_{14}, v_{15}$
$v_3 : v_1, v_2, v_4, v_7, v_{12}$	$v_{14} : v_{11}, v_{13}, v_{15}$
$v_4 : v_1, v_2, v_3, v_5, v_6, v_8, v_9, v_{10}, v_{12}$	$v_{15} : v_{11}, v_{13}, v_{14}, v_{16}$
$v_5 : v_4, v_6, v_7, v_9, v_{10}$	$v_{16} : v_{15}, v_{17}$
$v_6 : v_1, v_2, v_4, v_5, v_7, v_8, v_9, v_{12}$	$v_{17} : v_{12}, v_{16}, v_{18}, v_{19}, v_{21}, v_{22}$
$v_7 : v_3, v_5, v_6$	$v_{18} : v_{10}, v_{17}, v_{19}, v_{21}, v_{22}$
$v_8 : v_1, v_4, v_6, v_9$	$v_{19} : v_{17}, v_{18}, v_{21}, v_{22}$
$v_9 : v_1, v_4, v_5, v_6, v_8, v_{10}$	$v_{20} : v_{10}, v_{21}, v_{22}$
$v_{10} : v_4, v_5, v_9, v_{11}, v_{18}, v_{20}$	$v_{21} : v_{17}, v_{18}, v_{19}, v_{20}$
$v_{11} : v_{10}, v_{12}, v_{13}, v_{14}, v_{15}$	$v_{22} : v_{17}, v_{18}, v_{19}, v_{20}$

Thus, we conclude that the maximum independent set of the initial graph contains less than 8 vertices. That is, $|\max S(G)| \leq |S| + Est = 8$.

Section 1 states that the applied problem of planning freight railway transportation can be solved in the frame of the problem of the maximum independent set of undirected conflict graphs. Graphs that arise in such applied areas are sparse. In this connection, the approximate algorithm $\mathcal{B}(G, V)$ acquires special interest for solving the applied problem under consideration.

4 Level representation of the graph

Consider an undirected connected graph $G = (V, E)$. Let $v, v \in V$ be an arbitrary vertex of the graph $G = (V, E)$.

Definition 4.1. The sequence of subsets of vertices of graph $G = (V, E)$ is called level representation with the root vertex v .

$$V(G, v) : V_1(G, v), V_2(G, v), \dots, V_k(G, v)$$

if it is constructed as follows:

$$V_1(G, v) = \{v\},$$

$$V_2(G, v) = \mathcal{N}(V_1) - V_1(G, v),$$

Table 1 Steps of Algorithm $\mathcal{B}(G, V)$

	V_0	S	Est
1	$\{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13}, v_{14}, v_{15}, v_{16}, v_{17}, v_{18}, v_{19}, v_{20}, v_{21}, v_{22}\}$	$\{ \}$	0
2	$\{v_2, v_3, v_5, v_7, v_{10}, v_{11}, v_{12}, v_{13}, v_{14}, v_{15}, v_{16}, v_{17}, v_{18}, v_{19}, v_{20}, v_{21}, v_{22}\}$	$\{v_8\}$	0
3	$\{v_2, v_3, v_5, v_7, v_{10}, v_{12}, v_{16}, v_{17}, v_{18}, v_{19}, v_{20}, v_{21}, v_{22}\}$	$\{v_8, v_{13}\}$	0
4	$\{v_5, v_7, v_{10}, v_{16}, v_{17}, v_{18}, v_{19}, v_{20}, v_{21}, v_{22}\}$	$\{v_2, v_8, v_{13}\}$	0
5	$\{v_{10}, v_{16}, v_{17}, v_{18}, v_{19}, v_{20}, v_{21}, v_{22}\}$	$\{v_2, v_7, v_8, v_{13}\}$	0
6	$\{v_{10}, v_{18}, v_{19}, v_{20}, v_{21}, v_{22}\}$	$\{v_2, v_7, v_8, v_{13}, v_{16}\}$	0
7	$\{v_{10}, v_{20}\}$	$\{v_2, v_7, v_8, v_{13}, v_{16}, v_{19}\}$	1
8	$\{ \}$	$\{v_2, v_7, v_8, v_{10}, v_{13}, v_{16}, v_{19}\}$	1

.....,

$$V_k(G, v) = \mathcal{N}(V_{k-1}) - V_{k-2}(G, v),$$

$$V_{k+1}(G, v) = \{ \},$$

where a neighborhood of a subset of vertices pertains to a union of neighborhoods of vertices entering this subset. That is,

$$\mathcal{N}(V_i) = \cup \{ \mathcal{N}(v) : v \in V_i(G, v) \}.$$

For definiteness, we assume that for any level in the representation $V(G, v)$, the following holds:

$$V(G, v) = \{ v_{i_1}, v_{i_2}, \dots, v_{i_k} : i_1 < i_2 < \dots < i_k \},$$

$$i_j \in [1, n],$$

where n is the number of vertices of the original graph. In other words, the vertices of any level in the representation $V(G, v)$ are ordered by increasing indexes.

Suppose that the level representation $V(G, v)$ of the graph $G = (V, E)$ with root vertex v has k levels. Consider the adjacent levels $V_n(G, v)$ and $V_{n-1}(G, v)$, where $n \in [2, k + 1]$. Let

$$E(v_i, V_n(G, v)) = \{ (v_i, v_j) : v_j \in V_{n-1}(G, v) \}$$

be the set of between-level edges incident to the vertex $v_i \in V_n(G, v)$. The edges incident to the vertices of level $V_{n+1}(G, v)$ for $n \in [2, k - 1]$ are not considered at this stage.

For each vertex $v_i \in V_n(G, v)$, we order the set $E(v_i, V_n(G, v))$ by increasing the indexes of vertices v_j . Thus, if

$$e_{m_{in}} = (v_i, v_{j_2}), e_{m_{in}} \in E(v_i, V_n(G, v))$$

and

$$e_{m_{2in}} = (v_i, v_{j_2}), e_{m_{2in}} \in E(v_i, V_n(G, v)),$$

then the following holds:

$$m_{1in} < m_{2in} \text{ if and only if } j_1 < j_2.$$

Indexes m_{in} of edges in the ordered set $E(v_i, V_n(G, v))$ are called relative indexes of edges, that is, edges incident to vertex v_i from level $V_n(G, v)$ and a few vertices from the previous level $V_{n-1}(G, v)$ in the representation $V(G, v)$.

Definition 4.2. The tree of level representation $TrV(G, v)$ is a construction obtained from $V(G, v)$ by the removal of all edges in each level and between-level edges whose relative indexes are more than 1.

$$TrV_1(v) = \{ v \},$$

$$TrV_2(v) = \{ v_i : v_i \in V_2(v) \},$$

and

$$TrV_k(v) = \{ v_i : v_i \in V_k(v) \}.$$

We note that the removal of multiples between-level edges is designed to preserve the tree structure, that is, the elimination of cycles.

Example 4.1. Figure 2 shows an undirected graph $G = (V, E)$, its level representation with root vertex v_1 , and the corresponding tree of the level representation.

By definition, the tree of the level representation contains the same number of levels as does the level representation itself.

5 Estimates for the number of vertices in the maximum independent set

On the basis of the level representation, the lower and upper bounds for the number of vertices in the maximum independent set are obtained in this section. In addition, the section illustrates that Algorithms $\mathcal{A}(G, V_0)$, $\mathcal{B}(G, V)$ and modified algorithm $IndSetTree(G)$ can be used to obtain these bounds as well as any other effective algorithm.

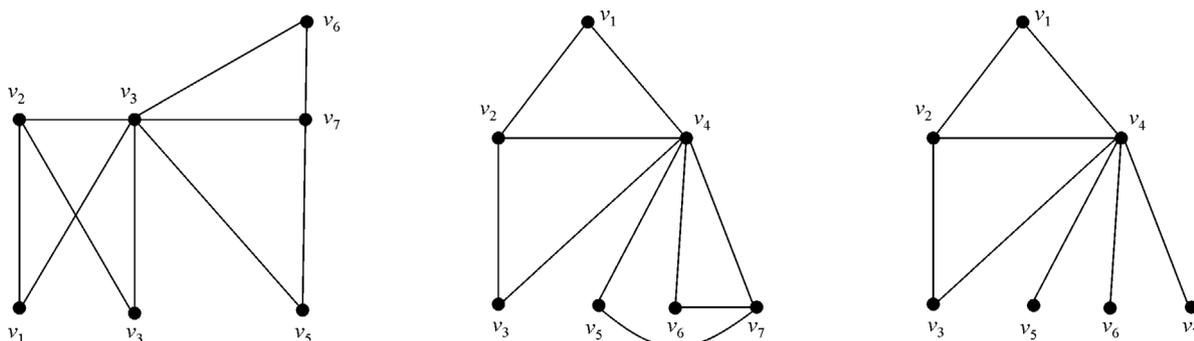


Fig. 2 From left to right: Undirected graph, level of representation, and tree of level representation

5.1 Lower bound

Let the undirected connected graph $G = (V, \varepsilon)$ be given and its level representation for an arbitrary vertex v .

Let $S(G)$ be the maximum independent set of vertices in graph $G = (V, \varepsilon)$, and $S(V(G, v))$ is the maximum independent set of vertices in the subgraph generated by the vertices of a certain level $V(G, v)$. For brevity, we denote

$$S(V) = S(V(G, v)),$$

which pertains to the maximum independent set of vertices in the corresponding level.

Let I be the set of indexes of non-adjacent levels in the level representation $V(G, v)$.

$$I = \{i : \text{for any } i_1, i_2 \text{ holds } |i_1 - i_2| > 1\}, \quad (1)$$

where $i \in [1, k]$. We denote the set of all maximal by inclusion sets of the form (1)

$$\mathcal{I} = \{I : \text{for any } i \notin I \text{ one can find } j \in I$$

$$\text{such that } |i - j| = 1\}. \quad (2)$$

The lower bound for the maximum independent set of vertices of the graph $G = (V, E)$ given in the form of the level representation with the root vertex v , $v \in V$ is the quantity

$$|S| \geq \max \left\{ \sum_{i \in I} |S(V_i)|, I \in \mathcal{I} \right\}, \quad (3)$$

All levels with indexes from any set $I \in \mathcal{I}$ are not adjacent to each other by construction. Consequently, the maximum of the sums (4) is an independent set in the graph $G = (V, \varepsilon)$.

In each level, the maximum independent set of vertices $S(V_i)$, $i = [1, k]$ can be found by either algorithms $\mathcal{A}(G, V_0)$ and $\mathcal{B}(G, V)$ or for each level, an analogous lower bound can be obtained. Then, the lower bound for the maximum independent set of vertices of the original graph will be no less than the greatest of the sums of the lower bounds obtained.

Thus, either a lower bound will be obtained for the maximum independent set of vertices of the original graph (if one can find the maximum independent sets of vertices for each level) or a substantial reduction in the dimension of the original problem will be achieved by moving to a series of simpler problems. The sets of indexes of even and odd non-adjacent levels and subsequent choice of the maximum of the sums of the form (3) can be naturally considered.

5.2 Upper bound

Consider the graph $G = (V, E)$ given by its level representation $\mathbb{V}(G, v)$ with an arbitrary root v . The tree of this level representation is $TrV(G, v)$.

The maximum independent set of vertices in the tree $TrV(G, v)$ can be found using the modified algorithm $IndSetTree(G)$ (or any other algorithm, such as dynamic programming method). Then, the upper bound for the maximum independent set of vertices in the original graph is

$$|S(G)| \leq |S(TrV(G, v))|. \quad (4)$$

The relation (4) easily follows from Proposition 3.1. By construction, the tree of the level representation $TrV(G, v)$ has the same vertices as the original graph. However, its set of edges is evidently smaller. Hence, inequality (4) holds under Proposition 3.1.

5.3 Two-sided estimate

Considering (3) and (4), we obtain a two-sided estimate for the maximum independent set of vertices in an undirected graph.

Proposition 5.1. Let $G = (V, E)$ be an undirected connected graph defined by its level representation $\mathbb{V}(G, v)$ for an arbitrary vertex v . Then,

$$\max \left\{ \sum_{i \in I} |S(V_i)|, I \in \mathcal{I} \right\} \leq |S(G)| \leq |S(TrV(G, v))|. \quad (5)$$

Proof. The proof follows from the arguments presented as comments on the validity of the relations (3) and (4).

An important feature of the two-sided estimate obtained is the possibility of using it for any level representation of the original graph.

We denote $LowB(v)$ and $UpB(v)$ as the lower and upper bounds, respectively, of the form that is obtained for level representation with root vertex v . Then, using the above-mentioned approach for each vertex of the graph, we obtain

$$\max \{LowB(v), v \in V\} \leq |S(G)| \leq \min \{UpB(v), v \in V\}.$$

Let us consider the example of using the approach described above for the undirected graph on the basis of Example 3.2.

Example 5.1. For each vertex of the graph as a root, level representations of the graph, and corresponding trees of level representations are constructed. To find the maximum independent set in each level V_i Algorithm $\mathcal{B}(G, V)$ is used. For finding the maximum independent set of the tree, Algorithm $IndSetTree(G)$ is used. Hence, in particular, from the level representation with the root vertex v_1 , a two-sided estimate of the maximum indepen-

dent set of form $8 \leq |S(G)| \leq 13$ is obtained. Among all upper bounds, we obtain the minimum one from the level representation with root vertex v_4 , and this bound is equal to 12. Similarly, the maximum of the lower bounds is equal to 8, and this bound is obtained from the level representation with root vertex v_8 . Thus, we obtain

$$\begin{aligned} \max \{LowB(v), v \in V\} &= LowB(v_8) = 8 \leq |S(G)| \leq 12 \\ &= UpB(v_4) = \min \{UpB(v), v \in V\}. \end{aligned}$$

According to the results obtained in Example 3.2, the cardinality of the maximum independent set of the initial graph is equal to 8.

Conflict graphs that arise in railway operation are usually sparse. In this connection, the implementation of the approach described above for solving applied problems means that an exact solution that is a coincidence of the upper and lower bounds can be expected for the maximum independent set of vertices.

6 Conclusions

This work investigates the applied problem of management of transport processes at the planning stage of freight railway transportation. Within the framework of the graph model, the investigation focuses on solving the problem of finding the maximum independent set of vertices in the undirected graph. To solve this problem, special graph constructions are introduced to the level representation and tree of the level representation. On the basis of the approximate algorithm and its modification for the class of trees, the lower and upper bounds for the number of vertices in the maximum independent set are obtained. The implementation of the approach is closely related to the abovementioned graph constructions. In particular, the upper bound is obtained using a modified algorithm for the tree of level representation.

The direction of further research is mainly connected with computational experiments and using the developed approach to solve the applied problem of planning of freight railway transportation.

Another aspect of further research concerns the excess normative threads in the conflict-free set; these threads are an important feature of the developed approach. In general, this case is sufficient for the execution of an arbitrary transportation plan. However, the normative threads are unsuitable for the execution of transportation. Thus, an adapted model where each vertex of the undirected conflict graph corresponds to a certain priority must be developed. Thus, further studies of the authors' team will be devoted to the development of such a model and effective algorithms for solving the planning problem while

considering the given transportation plan.

References

- Akimova E, Gainanov D, Golubev O (2015). The problem of scheduling for the linear section of a single-track railway with independent edges orientations. In: Proceedings of CEUR Workshop. 1513: 130–136
- Burdett O, Kozan E (2010). A disjunctive graph model and framework for constructing new train schedules. *European Journal of Operational Research*, 200(1): 85–98
- Dasgupta S, Papadimitriou C, Vazirani U (2006). *Algorithms*. New York: McGraw-Hill
- Duarte A, Pantrigo J, Pardo E, Mladenovic N (2015). Multi-objective variable neighborhood search: An application to combinatorial optimization problems. *Journal of Global Optimization*, 63(3): 515–536
- Gafarov E, Dolgui A, Lazarev A (2015). Two-station single-track railway scheduling problem with trains of equal speed. *Computers & Industrial Engineering*, 85: 260–267
- Gainanov D, Konygin A, Rasskazova V (2016). Modelling railway freight traffic using the methods of graph theory and combinatorial optimization. *Automation and Remote Control*, 77(11): 1928–1943
- Gainanov D, Rasskazova V (2016). An inference algorithm for monotone boolean functions associated with undirected graphs. *Bulletin of the SUSU*, 9(3): 17–30
- Gholami O, Sotskov Y N (2014). Scheduling algorithm with controllable train speeds and departure times to decrease the total train tardiness. *Int Journal of Industrial Engineering Computations*, 5(2): 281–294
- Gholami O, Sotskov Y N, Werner F (2012). Job-shop problems with objectives appropriate to train scheduling in a single-track railway. In: Proceedings of 2nd International Conference on Simulation and Modeling Methodologies, Roma. Technologies and Applications: 425–430
- Hansen P, Mladenovic N, Todosijevic R, Hanafi S (2017). Variable neighborhood search: basics and variants. *European Journal on Computational Optimization*, 5(3): 423–454
- Korte B, Vygen J (2008). *Combinatorial Optimization*. Berlin: Springer
- Lazarev A, Nekrasov I, Pravdivets N (2018). Evaluating typical algorithms of combinatorial optimization to solve continuous-time based scheduling problem. *Algorithms*, 4(11): 1–13
- Pei J, Cheng B, Liu X, Pardalos P M, Kong M (2017a). Single-machine and parallel-machine serial-batching scheduling problems with position-based learning effect and linear setup time. *Annals of Operations Research*, doi: 10.1007/s10479-017-2481-8
- Pei J, Liu X, Fan W, Pardalos P M, Lu S (2017b). A hybrid BA-VNS algorithm for coordinated serial-batching scheduling with deteriorating jobs, financial budget, and resource constraint in multiple manufacturers. *Omega*, doi: 10.1016/j.omega.2017.12.003
- Skiena S (2008). *The Algorithm Design Manual*. Berlin: Springer
- Todosijevic R, Mladenovic N (2016). Nested general variable neighborhood search for the periodic maintenance problem. *European Journal of Operational Research*, 252(2): 385–396