

Assigning airport gates during a pandemic using tolerance graphs

Dean Crnković, Andrea Švob[✉]

ABSTRACT

Tolerance graphs, introduced by M. C. Golumbic and C. L. Monma in 1982, are a generalization of interval graphs. In this paper, we propose an application of coloring of tolerance graphs together with machine learning, in particular supervised learning, in solving problems of airport gate assignment during a pandemic of an airborne disease. The idea is to minimize a contact between passengers at the gates, in order to slow down a spread of the disease. This application includes calculating the chromatic number of a graph and finding a clique of a given size in the graph. As a result, we obtain the minimum number of gates needed under the given assumptions and the corresponding gate assignment. Further, we propose an application of list coloring, a generalization of a coloring, of tolerance graphs in solving these problems. Besides the theoretical approach, the corresponding algorithms are given. The algorithms developed may take into account several parameters, such as the number of passengers on a flight, the number of newly infected people per 1000 inhabitants. A similar approach can be taken for classroom assignment during a pandemic, scheduling meetings, etc.

Keywords: gate assignment, classroom assignment, tolerance graph, chromatic number, list coloring, machine learning

1. Introduction

Recent pandemic of Covid-19 disease, that caused loss of many lives all around the world and also had a huge negative impact on economics, reminded us how important is to have

[✉] Corresponding author.

E-mail address: asvob@math.uniri.hr (A. Švob).

Received 16 Jun 2025; Revised 31 Jul 2025; Accepted 19 Aug 2025; Published Online 28 Sep 2025.

DOI: [10.61091/jcmcc127-20](https://doi.org/10.61091/jcmcc127-20)

© 2025 The Author(s). Published by Combinatorial Press. This is an open access article under the CC BY license (<https://creativecommons.org/licenses/by/4.0/>).

strategies to decrease the spread of a disease in case of a pandemic. Graph theory can offer models in order to help to fight the pandemic. For example, graphs can be used for tracing the contacts of infected people, doing some medical analysis for understanding the virus and disease dynamics, and in general predicting the dynamics of a pandemic. In this paper, we use tolerance graphs to assign airport gates and classrooms during a pandemic using tolerance graphs. This approach can also be used to prevent a spreading of an airborne disease before a pandemic occur, e.g. in a case of seasonal flu.

The problem of assigning gates to arriving and departing flights is one of the most important problems in airport operations. For some recent work on gate assignment problem we refer the reader to [1, 8, 11, 12, 13, 26]. The gate assignment problem becomes especially difficult during a pandemic. A pandemic also imposes major problems on classroom assignment when organizing live teaching at schools and universities. For the literature on classroom assignment problems we refer the reader to [5, 20, 22]. In this paper, we propose an application of colouring of tolerance graphs together with supervised learning in solving problems of airport gate assignment and classroom assignment in live teaching during a pandemic of an airborne disease. Further, we show that, in order to get better results, instead of a colouring one can use a list colouring, which is a generalization of classical colouring.

Solving problems of assigning gates and classrooms during a pandemic using tolerance graphs were considered in [6]. In this paper, we further develop that approach by combining colouring of tolerance graphs with machine learning and by using list colouring.

2. Preliminaries

All graphs in this paper are finite and simple. For basic definitions and further reading in graph theory we refer the reader to [23]. In the sequel, we give definitions related to vertex colourings, list colourings and tolerance graphs, that we later use in algorithms.

A (proper) *vertex colouring* of a graph $G = (V, E)$ is a map $c : V \rightarrow S$, where S is the set of colours, such that $c(v) \neq c(w)$ whenever v and w are adjacent. Given an integer k , a k -colouring c is a map that assigns to each vertex v of the graph G an integer $c(v)$ chosen in the set $\{1, 2, \dots, k\}$ (the set of colours). The smallest integer k such that G has k -colouring is the chromatic number that is usually denoted by $\chi(G)$. A graph G having that $\chi(G) = k$ is called k -chromatic and if $\chi(G) \leq k$ we call G k -colourable.

Given a set $L(v)$ of colours for each vertex v of a graph $G = (V, E)$, a *list colouring* is a choice function that maps every vertex $v \in V$ to a colour in the list $L(v)$, such that no two adjacent vertices receive the same colour. A set $L(v)$ is called a *list*. A graph is k -*list-colourable* (or k -choosable) if it has a proper list colouring no matter how one assigns a list of k colours to each vertex. The *list chromatic number* (or choosability) $ch(G)$ of a graph G is the least number k such that G is k -list-colourable. An important result with an elegant proof is given in [4], where C. Thomassen proved that every planar graph is 5-list-colourable. List coloring arises in practical problems concerning channel/frequency assignment (see [21, 25]).

Tolerance graphs were introduced in 1982 by Golumbic and Monma [14], in order to

generalize interval graphs. The motivation for this generalization were several scheduling problems. Properties and several variations of tolerance graphs were studied, e.g., in [16, 15].

Let $\mathcal{F}$ be a collection of set. The *intersection graph* of $\mathcal{F}$ is defined as the graph obtained by assigning a distinct vertex to each set in $\mathcal{F}$ and joining two vertices by an edge when their corresponding sets have nonempty intersection. We will deal with a collection of intervals on the real line.

Before defining tolerance graphs we will introduce another family of intersection graphs i.e. interval graphs which are actually the special case of tolerance graphs. An *interval graph* $G = (V, E)$ is a graph for which each vertex $v \in V$ is associated with an real interval I_v , and two vertices are connected by an edge in G if the associated intervals have nonempty intersection. The set of intervals $\{I_v \mid v \in V\}$ is an interval graph representation of G . In other words,

$$uv \in E(G) \iff I_u \cap I_v \neq \emptyset, \text{ for all } u, v \in V(G).$$

Tolerance graphs are defined in a similar manner as interval graphs, but here the sizes of intersecting intervals will have the crucial role. A graph $G = (V, E)$ is a *tolerance graph* if each vertex $v \in V$ can be assigned a closed interval I_v and a tolerance $t_v \in \mathbb{R}^+$ such that $xy \in E$ if and only if $|I_x \cap I_y| \geq \min\{t_x, t_y\}$. A collection $\langle I, t \rangle$ of intervals and tolerances is called a *tolerance representation*, $I = \{I_x \mid x \in V\}$, $t = \{t_x \mid x \in V\}$. If for all $v \in V$, $t_v \leq |I_v|$, the graph is called a *bounded tolerance graph* and the representation is called a *bounded tolerance representation*.

An important property of tolerance graphs is that they are perfect graphs (see [10, 17]). Perfect graphs have the property that for some problems that are in general NP-complete, such as determining the chromatic number and the clique number, they allow polynomial-time algorithms for obtaining the solution (see [18]). For more information see [7].

Machine learning is a branch of Artificial Intelligence that is using models and known examples in order to solve problems for which it does not include huge amount of known data. The algorithms in machine learning belong to one of the types of learning i.e. supervised, unsupervised and reinforcement learning algorithms. In this paper, we are interested in supervised learning. The main set here must be defined in the start of an algorithm and is called training set data. The set of possible outputs is also given. These algorithms learn from the training set and after training on some amount of data, predict accurate outputs. The most important thing here is to make good training set. This can be done by one of the algorithms: Linear Regression, Decision tree, K-Nearest Neighbours, Support Vector Machine, K-means clustering, Naive Bayes, etc.

For the computations dealing with the problems given in this paper one can use Python [9] and SageMath [24].

3. Application of machine learning and coloring of tolerance graphs

In this section we propose an application of supervised learning together with a coloring of tolerance graphs in solving problems of airport gate assignment and classroom assignment during a pandemic, in order to combat a pandemic of an airborne disease, such as COVID-19 or SARS-CoV-1, by limiting the spread of the disease. We present a modification of the algorithms given in [6], which involve the use of machine learning algorithms, in particular supervised learning. In the subsections below, we will refer to the problems given in [6] and focus on the modifications of the algorithms which use machine learning to determine tolerances in the tolerance graphs used. For the sake of completeness we will explain these real-life problems which could be solved by the algorithms.

3.1. Application to airport gate assignment problem

Suppose that the airport X has a rule that the passengers that are using domestic flights must be at the gates 2 hours before the flight, and those getting international flights 3 hours before. In order to decrease the possibility of spreading a disease among passengers, the airport wants to schedule flights x and y for different gates if the period of time when the passengers for these two flights should be at the gates overlap for more than D_{xy} minutes. One of the problems that airports are facing during the pandemic is to determine the minimum number of gates needed for a particular day and assign gates to the flights, and at the same time to prevent the spread of a disease at the airport. In order to solve that problem, we can use a machine learning algorithm to optimize the values D_{xy} . The main idea is to build a tolerance graph based on the information about the flights and use a machine learning algorithm to determine tolerances. Then for two vertices (flights) x and y the overlapping time D_{xy} will be $\min\{t_x, t_y\}$.

In order to determine the tolerances, we propose using of machine learning, more precisely we will use supervised learning. The idea of supervised learning as a type of machine learning algorithms lies in a fact that given input variables, and predicted output variables, the algorithm learns how to map input variables to output variables. The goal is that the mapping from inputs to outputs is approximated well so that for a new input, algorithm gives very precise output. In this problems, we will use regression as the procedure that we find useful. Regression is a technique usually used when dealing with real-valued data, which is our case. The main formula of the regression used for the problem described in this section is

$$f(x_1, x_2) = a + b \cdot x_1 + c \cdot x_2,$$

where $v = (x_1, x_2)$ is an input vector and a, b and c are the parameters that should be learned. Let v be a vector $v = (x_1, x_2)$, where x_1 denotes the number of people planned to be on the flight, and x_2 denotes the number of newly infected patients on 1000 inhabitants. The value $f(x_1, x_2)$ is tolerance expressed in minutes. As the variables x_1 and x_2 are increasing, the tolerance $f(x_1, x_2)$ will be decreasing. After proceeding with machine learning algorithm, we get the tolerance $f(x_1, x_2)$ depending on the health

situation and the number of passengers on the flight. Collecting the data from the airport we get "the tested values" from which the parameters are learned. In each of the "new values", the tolerance is immediately calculated. After this procedure we have the value needed for further calculations. Below we give a pseudo code of this procedure.

3.1.1. Function - predicting the tolerances - machine learning model.

1. input1 = dataset $[f(x_1, x_2), x_1, x_2]$
2. split dataset into TestSet and TrainSet
3. parameters $[a, b, c] = \text{LinearRegression}(\text{TrainSet})$
4. for each sample in TestSet do
5. Check(sample, parameters)
6. end for
7. return parameters $[a, b, c]$
8. x_2 = number of newly infected patients per 1000 inhabitants
9. for each flight j on fixed day do
10. x_1 = number of passengers on flight j
11. input2 = NewValue $[x_1, x_2]$
12. LinearRegression(input2)
13. $t_j = f(x_1, x_2)$
14. return tolerance t_j
15. end for

Remark 3.1. The main formula of the regression used for the problem described in this section is a function depending on two variables. Depending on the problem, the formula can be changed and modified in a way that more variables and parameters are used and different type of function is applied.

After obtaining the tolerances, we can proceed with a search for the minimum number of gates that can be used at the airport X in a particular day. Let $I_j = [s_j - 2, s_j]$, where s_j is the departure time of the flight j scheduled for the day if the flight is domestic and let $I_j = [s_j - 3, s_j]$, where s_j is the departure time of the flight j if the flight is international. Further, let $I = \{I_1, \dots, I_{m_1}, I_{m_1+1}, \dots, I_{m_1+m_2}\}$ be the set of intervals for each flight scheduled for the exact day with m_1 domestic and m_2 international flights. Each interval represents a vertex of the graph G . If $|I_x \cap I_y| \leq \min\{t_x, t_y\} = D_{xy}$, for flights x and y the same gates can be used. The minimum number of gates that must be used is equal to the chromatic number of the corresponding tolerance graph. With this approach we also find a colouring with the minimum number of colours and thereby assign the minimum number of gates to the flights (each colour corresponds to a gate). Below we give a pseudo code of an algorithm for determining the minimum number of gates described in [6].

3.1.2. Minimum number of gates.

1. for j in 1 to n :
2. input = (s_j, e_j, t_j)
3. Gamma = graphs.ToleranceGraph(input)
4. from sage.graphs.graph_coloring import chromatic_number
5. ChromaticNumber = chromatic_number(Gamma)
6. from sage.graphs.graph_coloring import all_graph_colorings
7. GraphColoring = all_graph_colorings(Gamma, ChromaticNumber)
8. **return** [Gamma, ChromaticNumber, next(GraphColoring)]

Remark 3.2. Instead of looking for the minimum number of gates needed, and the corresponding gate assignment, the airport staff may be interested in assigning k gates to the flights, for a certain fixed number k . In that case we can skip the line 4. in the algorithm and proceed by taking k instead of the variable ChromaticNumber.

3.2. Application to classroom assignment during a pandemic

An important issue during the pandemic crisis is the organization of live teaching and meetings. One of the main problems that appears with this item is the use of common classrooms within different courses or meetings. In [6], we propose the use of tolerance graphs to solve this type of problems. In this paper, we introduce the use of machine learning in solving the problem of calculating the minimum number of classrooms that have to be used.

For the sake of completeness we give here the problem proposed in [6]. Suppose that in a particular day there are v courses that should be organized in a campus building having exactly x classrooms, and that the schedule is given. The time slot reserved for a course includes one extra hour which is supposed to be used for cleaning the classroom between two lectures. The problem is to determine the minimum number of classrooms that have to be used on that day at the campus building, and assign classrooms to the lectures.

Let $I_j = [s_j, e_j + 1]$, where s_j and e_j are the starting and ending time, respectively, of the class j scheduled for the day. We suppose here that one hour is added for the cleaning. Further, let $I = \{I_1, \dots, I_n\}$ be the set of intervals for each of the n courses scheduled for the particular day. Each interval represents a vertex of the graph G . To each vertex we add the tolerance t_j , $j = 1, \dots, n$, that depends on the number of students enrolled in a course.

In order to determine the tolerances, we propose using of machine learning, more precisely we will use supervised learning similarly as in Section 3.1. The main formula of the regression used for the problem described here is

$$f(x_1, x_2) = a + b \cdot x_1 + c \cdot x_2,$$

where $v = (x_1, x_2)$ is an input vector and a, b and c are the parameters that should be learned. Let v be a vector $v = (x_1, x_2)$, where x_1 denotes the number of students enrolled

in the course, and x_2 denotes the number of newly infected patients on 1000 inhabitants. The value $f(x_1, x_2)$ is tolerance expressed in minutes. After proceeding with machine learning algorithm, we get the tolerance $f(x_1, x_2)$ depending on the health situation and the number of students enrolled in the course. Below we give a pseudo code of this procedure.

3.2.1. Function - predicting the tolerances - machine learning model-classroom.

1. input1 = dataset $[f(x_1, x_2), x_1, x_2]$
2. split dataset into TestSet and TrainSet
3. parameters $[a, b, c] = \text{LinearRegression}(\text{TrainSet})$
4. for each TestSample in TestSet do
5. Check(TestSample, parameters)
6. end for
7. return parameters $[a, b, c]$
8. $x_2 =$ number of newly infected patients on 1000 inhabitants
9. for each course j on fixed day do
10. $x_1 =$ number of people at course j
11. input2= NewValue $[x_1, x_2]$
12. LinearRegression(input2)
13. $t_j = f(x_1, x_2)$
14. return t_j
15. end for

After obtaining the tolerances, we define the tolerance graph. If $|I_x \cap I_y| \leq \min\{t_x, t_y\}$, for classes x and y the same classroom can be used. The minimum number of classrooms that must be used is equal to the chromatic number of the corresponding tolerance graph (each colour corresponds to a classroom). We remark here that the same approach can be used for organizing meeting and many other events.

In a similar way as in Remark 3.2, instead of using the minimum number of classrooms a university may be interested to assign a certain number k of classrooms to the lectures.

4. Application of list colouring

Sometimes the use of ordinary graph colouring in the algorithm given in Section 3.1 may give us a gate assignment that is not suitable in practice. For example, it is possible that a flight with a large number of passengers is assigned to a gate that is built for a smaller number of passengers. To overcome this obstacle, instead of a colouring one can use a list colouring. In order to use a list colouring in a gate assignment problem, for every flight v a list $L(v)$ is built, that consists of all the gates that are suitable for v . In such a way, the colouring obtained by using these lists will represent a gate assignment such that, to

each flight v a suitable gate from the list $L(v)$ is assigned. When using a list colouring, the minimum number of gates that must be used is equal to the minimum number of colours needed for proper colouring of the corresponding graph with respect to the list $L(v)$, where each colour corresponds to a gate.

Example 4.1. Let the vertices u_1 and u_2 represent domestic flights, and let v_1 and v_2 represent international flights. Further, let u_1u_2 , u_1v_1 and u_2v_2 be the edges of the corresponding tolerance graphs. The chromatic number of this tolerance graph is 2. For example, we can colour the vertices u_1 and v_2 with the colour 1, and the vertices u_2 and v_1 with the colour 2. Suppose that the gates 1 and 2 are intended for domestic flights, and the gates 3 and 4 are intended for international flights. Then we cannot use the gate assignment given above. In order to obtain a gate assignment that assigns gates 1 and 2 to domestic flights and gates 3 and 4 to international flights, we will use list colouring. The corresponding lists are given as follows

$$L(u_1) = L(u_2) = \{1, 2\}, L(v_1) = L(v_2) = \{3, 4\}.$$

With respect to these lists, we need at least 3 colours for (proper) colouring of the tolerance graph, i.e. we need at least 3 gates. For example, we can colour the vertex u_1 with the colour 1, the vertex u_2 with the colour 2, and the vertices v_1 and v_2 with the colour 3.

Similarly, a use of ordinary colouring to assign classrooms in Section 3.2 (and [6]) can result with a schedule where a course with large number of students is scheduled for a small classroom. To prevent that to happen, to each course we assign a list of suitable classrooms. The minimum number of classrooms required is equal to the minimum number of colours needed for proper colouring of the corresponding graph with respect to the given list, where each colour corresponds to a classroom.

While the colouring problem is polynomially solvable for perfect graphs, the list colouring problem is NP-complete for general perfect graphs, including interval graphs (see [2, 3, 19]), and tolerance graphs too. So, using list colouring to solve the problems described in this paper can give better results than the classical colouring, but can also require more time to obtain a solution.

5. Conclusion

In this paper, we propose two applications of machine learning together with tolerance graphs in fighting pandemic of an airborne disease by slowing down a spread of the disease. The first application is airport gate assignment during a pandemic, and the second one is optimizing classroom assignment during a pandemic. These applications are based on supervised learning, which is a machine learning algorithm and on calculating the chromatic number of a graph, the problem that is in general NP-complete but for tolerance graphs can be solved in polynomial time. Further, we propose that instead of a colouring of a tolerance graph one can use a list colouring, which can give better results

in some cases. As a result, we obtain the minimum number of gates or classrooms, needed under the given assumptions. Moreover, we get the corresponding assignments of gates or classrooms. The same approach can be used for other similar problems. This approach can also be used to prevent a spreading of an airborne disease before a pandemic occur, e.g. in a case of seasonal flu.

Acknowledgement

This work has been fully supported by Croatian Science Foundation under the projects HRZZ-IP-2022-10-4571 and HRZZ-UIP-2020-02-5713. The authors thank the anonymous referee for helpful suggestions that improved the paper.

References

- [1] U. Benlic, E. K. Burke, and J. R. Woodward. Breakout local search for the multi-objective gate allocation problem. *Computers & Operations Research*, 78:80–93, 2017. <https://doi.org/10.1016/j.cor.2016.08.010>.
- [2] M. Biro, M. Hujter, and Z. Tuza. Precoloring extension. i. interval graphs. *Discrete Mathematics*, 100:267–279, 1992. [https://doi.org/10.1016/0012-365X\(92\)90646-W](https://doi.org/10.1016/0012-365X(92)90646-W).
- [3] F. Bonomo, G. Durán, and J. Marenco. Exploring the complexity boundary between coloring and list-coloring. *Annals of Operations Research*, 169:3–16, 2009. <https://doi.org/10.1007/s10479-008-0391-5>.
- [4] C. Thomassen. Every planar graph is 5-choosable. *Journal of Combinatorial Theory Series B*, 62:180–181, 1994. <https://doi.org/10.1006/jctb.1994.1062>.
- [5] W. Carter and C. A. Tovey. When is the classroom assignment problem hard? *Operations Research*, 40(No. 1, Suppl. 1):S28–S39, 1992. <https://doi.org/10.1287/opre.40.1.S28>.
- [6] D. Crnković and A. Švob. Application of tolerance graphs to combat covid-19 pandemic. *SN Computer Science*, 2 (83):8 pages, 2021. <https://doi.org/10.1007/s42979-021-00462-6>.
- [7] D. Jungnickel. *Graphs, Networks and Algorithms*. Springer Berlin Heidelberg New York, 2005.
- [8] U. Dorndorf, F. Jaehn, and E. Pesch. Flight gate assignment and recovery strategies with stochastic arrival and departure times. *OR Spectrum*, 39:65–93, 2017. <https://doi.org/10.1007/s00291-016-0443-1>.
- [9] P. S. Foundation. Python language reference. *version 3.9*:Available at <http://www.python.org>.
- [10] L. Lovász. A characterization of perfect graphs. *Journal of Combinatorial Theory, Series B*, 13:95–98, 1972. [https://doi.org/10.1016/0095-8956\(72\)90045-7](https://doi.org/10.1016/0095-8956(72)90045-7).
- [11] H. Li, X. Ding, J. Lin, and J. Zhou. Study on coloring method of airport flight-gate allocation problem. *Journal of Mathematics in Industry*, 9:16 pages, 2019. <https://doi.org/10.1186/s13362-019-0068-5>.

- [12] M. Li, J.-K. Hao, and Q. Wu. Learning-driven feasible and infeasible tabu search for airport gate assignment. *European Journal of Operational Research*, 302:172–186, 2022. <https://doi.org/10.1016/j.ejor.2021.12.019>.
- [13] J.-G. Liu, M.-L. Zhang, .- C. P, J.-W. Xie, and H.-F. Zuo. An integrative approach with sequential game to real-time gate assignment under cdm mechanism. *Mathematical Problems in Engineering*, Article ID 143501:13 pages, 2014. <https://doi.org/10.1155/2014/143501>.
- [14] M. C. Golumbic and C. L. Monma. A generalization of interval graphs with tolerances. *Congressus Numerantium*, 35:321–331, 1982.
- [15] M. C. Golumbic and R. E. Jamison. Rank-tolerance graph classes. *Journal of Graph Theory*, 52:317–340, 2006. <https://doi.org/10.1002/jgt.20163>.
- [16] M. C. Golumbic and A. N. Trenk. *Tolerance Graphs*. Cambridge University Press, Cambridge, 2004.
- [17] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas. The strong perfect graph theorem. *Annals of Mathematics*, 2 (164):352–360, 2006.
- [18] M. Grötschel, L. Lovász, and A. Schrijver. Polynomial algorithms for perfect graphs. *Annals of Discrete Mathematics*, 21:325–356, 1984.
- [19] D. Marx. Precoloring extension on unit interval graphs. *Discrete Applied Mathematics*, 154:995–1002, 2006. <https://doi.org/10.1016/j.dam.2005.10.008>.
- [20] J. M. Mulvey. A classroom/time assignment model. *European Journal of Operational Research*, 9:64–70, 1982. [https://doi.org/10.1016/0377-2217\(82\)90012-1](https://doi.org/10.1016/0377-2217(82)90012-1).
- [21] N. Garg, M. Papatriantafyllou, and P. Tsigas. Distributed long-lived list colouring: how to dynamically allocate frequencies in cellular networks. *Wireless Networks*, 8:49–60, 2002. <https://doi.org/10.1023/A:1012719525108>.
- [22] A. E. Phillips, H. Waterer, M. Ehrgott, and D. M. Ryan. Integer programming methods for large-scale practical classroom assignment problems. *Computers & Operations Research*, 53:42–53, 2015. <https://doi.org/10.1016/j.cor.2014.07.012>.
- [23] R. Diestel. *Graph Theory*. Springer-Verlag New York, 2012.
- [24] T. S. Developers. Sage mathematics software. *version 7.3*:2016. (<http://www.sagemath.org>).
- [25] W. Wang and X. X. Liu. List-coloring based channel allocation for open-spectrum wireless networks. *VTC-2005-Fall. 2005 IEEE 62nd Vehicular Technology Conference*:690–694, 2005.
- [26] L. Wang, Q.-L. Zhu, and X.-F. Xu. Optimised assignment of airport gate configurations using an immune genetic algorithm. *Mathematical Structures in Computer Science*, 24, Article ID e240517:14 pages, 2014. <https://doi.org/10.1017/S0960129513000765>.

Dean Crnković

Faculty of Mathematics, University of Rijeka, Radmile Matejčić 2, 51000 Rijeka, Croatia

Andrea Švob

Faculty of Mathematics, University of Rijeka, Radmile Matejčić 2, 51000 Rijeka, Croatia