Redundant 1-cells in Multi-labeled 2-Gmap Irregular Pyramids

Majid Banaeyan, Walter G. Kropatsch and Jiří Hladůvka

Pattern Recognition and Image Processing Group, TU Wien

1040 Wien, Favoritenstr. 9/5, E193-03, Vienna, Austria

{majid,krw,jiri}@prip.tuwien.ac.at

Abstract

Nowadays the amount of generated digital data is growing faster and faster in a broad spectrum of application domains such as biomedical and biological imaging, document processing, remote sensing, video surveillance, etc. Processing such big data encourages efficient data structure and powerful processing algorithms. The n-dimensional generalized map is a useful structure that completely represents the topological structure of an image. Their advantages have been widely proved in the literature. Nevertheless, the main disadvantage of these structures is the high rate of memory requirement. This paper, first proposes an efficient method that implicitly encodes two of the three involutions in the 2-Gmap that dramatically reduces the amount of required memory. Second, it introduces a new formalism to define and detect redundant 1-cells (edges), in the 2-Gmap. Removing such redundant information the reduced memory is further decreased approximately by half. Finally, experiments show the advantage of the proposed method in a real database of high-resolution X-ray micro-tomography (μCT) and fluorescence microscopy.

1. Introduction

We are live in the era of *Big Data*. In 2018 it was stated "Data volumes are exploding; more data has been created in the past two years than in the entire history of the human race [9]." Nowadays, the data volume and velocity is growing even faster [15]. Processing such a huge amount of data requires efficient data structures and efficient processing algorithms. In addition, currently we are working on the *Water's gateway to heaven* project[1] dealing with high-resolution X-ray micro-tomography (μCT) and fluorescence microscopy. The size of the labeled cross slice of a leaf scan is more than 2000 in each dimension. To correctly preserve the structure of the elements in the image, in this paper we employ 2-dimensional generalized map (2-

Gmap) [13].

Although the n-Gmap is an efficient structure for describing an n-dimensional orientable or non-orientable quasi-manifold [13] it suffers from requiring a huge amount of memory storage. To remedy this problem, this paper first introduces an efficient encoding to implicitly preserve elements of the 2-Gmap without taking extra space of memory. Second and more important, it introduces a new formalism to define and detect redundant elements of the 2-Gmap structure of the multi-labeled image.

By removing the redundant elements, the resulted 2-Gmap not only has the same structure to the original one but it would be also computationally more efficient to be used in upcoming processing. In particular, to process both general and local information of the structure we use the irregular graph pyramid. Removing such redundant elements in the hierarchical structure, simplifies and speeds up the construction of the pyramid. In this paper we are dealing with *multi-labeled* images. The multi-labeled image is defined as an image consists of different connected components (CCs) where each CC has a unique label (color).

1.1. Irregular Pyramid

Pyramids are powerful and efficient hierarchical structures in pattern recognition that were introduced by Rosenfeld [16]. They are able to propagate local information from the base level into global and abstract information at top of the pyramid [14]. Irregular image pyramids consist of a series of successively smaller images constructed over a base image [10]. By presenting a digital image as a 4-adjacent neighborhood graph, each pixel in image P corresponds to a vertex $v \in V$ of the graph $G = (V, E)$. Each edge, $e \in E$, of the graph encodes the neighborhood relationship between pixels. In addition, the gray-value of a pixel $g(p)$ becomes an attribute of the corresponding vertex v, $g(v) = g(p)$ and the $contrast(e) = |g(u) - g(v)|$ becomes an attribute of an edge $e(u, v)$ where $u, v \in V$. In an irregular pyramid, in order to produce the smaller graph at the upper level, two operations are performed at each level: edge contraction and edge removal [5, 6]. The former re-

---

[1]https://waters-gateway.boku.ac.at/

moves one edge and one vertex while preserving the connectivity of a graph and the latter removes one edge. Vertices (edges) of the current level that will be disappeared at the upper level are called *non-surviving* vertices (edges) while those that remain at the upper levels are called *surviving* vertices (edges). The decision of which vertices (and consequently which edges) must be selected as the surviving vertices (edges) is taken by introducing the *contraction kernel* (CK).

Definition 1 (Contraction Kernel (CK)) *A CK is a tree consisting of a surviving vertex as its root and some non-surviving neighbors with the constraint that every non-survivor can be part of only one CK.*

An arrow over an edge is commonly used to indicate the direction of contraction, i.e., from non-survivor to survivor vertex. Using the 4-adjacent neighborhood relationship results in the plane graph. A ***plane*** graph is a graph embedded in the plane such that its edges intersect only at their endpoints [18]. In a plane graph, a *face* is the connected spaces between edges and vertices where its degree is the number of edges bounding the face. A face bounded by a cycle is called an *empty* face.

1.2. Gmap

An n-dimensional generalized map (n-Gmap) is a combinatorial data structure allowing to describe an n-dimensional orientable or non-orientable quasi-manifold with or without boundaries [13]. An n-Gmap is defined by a finite set of darts $\mathcal{D}$ on which act $n + 1$ involutions[2] α_i, satisfying composition constraints of the following definition [7]:

Definition 2 (n-Gmap) *An n-dimensional generalized map, or n-Gmap, with $0 \leq n$ is an $(n + 2)$-tuple $G = (\mathcal{D}, \alpha_0, ..., \alpha_n)$ where:*
1. $\mathcal{D}$ is a finite set of darts,
2. $\forall i \in \{0, ..., n\}$: α_i is an involution on $\mathcal{D}$
3. $\forall i \in \{0, ..., n - 2\}$, $\forall j \in \{i + 2, ..., n\}$: $\alpha_i \circ \alpha_j$ is an involution.

A 2-Gmap $(\mathcal{D}, \alpha_0, \alpha_1, \alpha_2)$ represents the structure of a set of surfaces. Darts as the fundamental elements of the 2-Gmap are linked together by involution functions. For example in Fig. 1, $\alpha_0(21) = 22$, $\alpha_1(21) = 10$ and $\alpha_2(21) = 23$.

Definition 3 (i-cell) *Let $G = (\mathcal{D}, \alpha_0, ..., \alpha_n)$ be an n-Gmap, $d \in D$, and $i \in \{0, ..., n\}$. The i-dimensional cell (or i-cell) containing d is:*

$$c_i(d) = < \alpha_0, ..., \alpha_{i-1}, \alpha_{i+1}, ..., \alpha_n > (d) \qquad (1)$$

---
[2]self-inverse permutations

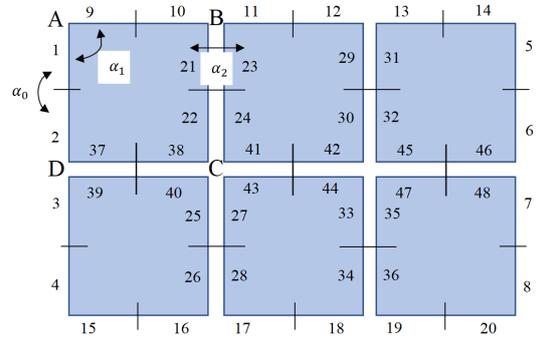Figure 1. An example of a 2-Gmap

where

1. $\langle \alpha_1, \alpha_2 \rangle(d)$ denotes the propagation of $(\alpha_1^*, \alpha_2^*)^*(d)$ and identifies the 0-cell (a point), the eight darts surrounding C in Fig. 1.

2. $\langle \alpha_0, \alpha_2 \rangle(d)$ denotes the propagation of $(\alpha_0^*, \alpha_2^*)^*(d)$ and identifies the 1-cell consisting of the four darts between B and C in Fig. 1.

3. $\langle \alpha_0, \alpha_1 \rangle(d)$ denotes the propagation of the orbit $(\alpha_0^*, \alpha_1^*)^*(d)$ and identifies the 2-cell between A, B, C and D in Fig. 1.

Based on the definition 3, in Fig. 1, $c_0(22) = \{22, 24, 41, 43, 27, 25, 38\}$ means this set of darts represents the 0-cell of the $d = 22$. In addition, the set $\{22, 21, 23, 24\}$ and the set $\{22, 21, 10, 9, 1, 2, 37, 38\}$ represent 1-cell and 2-cell corresponding to $d = 22$, respectively.

2. Corresponding graph of a 2-Gmap

Let G be a corresponding graph of a 2-Gmap. 0-cells and 1-cells of the 2-Gmap correspond to the vertices and edges of G, respectively. The 2-cells of the 2-Gmap correspond to the faces of degree 4 in the G. Fig. 2 shows G as the corresponding graph of the 2-Gmap of Fig. 1. Each edge of G consists of two half-edges or *darts*. There are three involutions, α_0, α_1 and α_2 encoding the relationships between darts (Fig. 3). To store the involutions one may consider an array of darts encoding each involution. However, we introduce specific encoding such that only one of these three involutions, i.e. α_1, explicitly be stored in the 1D array of darts. The remaining two involutions, α_0 and α_2, are implicitly encoded.

Assume G consists of M by N vertices containing $n_d = 2 \times 2(M + N) + 2 \times (2MN - M - N)$ darts. The first term, $2 \times 2(M + N)$, indicates the number of darts in the
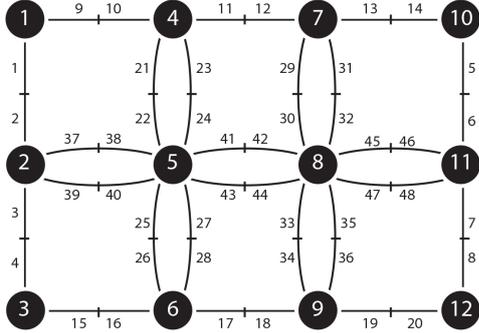
Figure 2. Corresponding graph G of a 2-Gmap

d	1	2	3	4	5	6	...	21	22	23	24	...
$\alpha_0(d)$	2	1	4	3	6	5	...	22	21	24	23	...
$\alpha_1(d)$	9	37	39	15	14	46	...	10	38	11	41	...
$\alpha_2(d)$	1	2	3	4	5	6	...	23	24	21	22	...

Figure 3. Array of darts in the 2-Gmap of Fig. 2

boundary where $\alpha_2(k) = k$ and $k \in [1, 2 \times 2(M + N)]$. In Fig. 2 these darts are indicated by numbers 1 to 20. The second term, $2 \times (2MN - M - N)$, illustrates the remaining darts where $\alpha_2(4k + 1) = 4k + 3$ and $\alpha_2(4k + 2) = 4k + 4$ where $k \in [(2 \times 2(M + N)) + 1, n_d]$. In this manner, α_2 is implicitly encoded. In Fig. 2, these darts are indicated by numbers 21 to 48. Furthermore, we consider $\alpha_0(2k - 1) = 2k$, where $k \in [1, n_d/2]$. Therefore, the α_0 can be implicitly encoded as well.

2.1. Edge Classification

A multi-labeled image consists of different labels where each label represents an object (connected component). In the neighborhood graph of an input image, each connected component (CC) consists of a set of vertices with the same label (color). In this regard, we partition the edges of the neighborhood graph into two categories: intra-CC and inter-CCs as follows:

Definition 4 *Intra-CC edge: an edge $e = (u, v)$ is **intra-CC** iff $g(u) = g(v)$.*

Definition 5 *Inter-CCs edge: an edge $e = (u, v)$ is **inter-CCs** iff $g(u) \neq g(v)$.*

Based on the definitions above, the contrast of an intra-CC edge is equal to zero, $c(e) = 0$. We show the intra-CC edge by $e_0 \in E_0$. On the other hand, the contrast of an inter-CCs edge is larger than zero, $c(e) > 0$. The inter-CCs edge is shown by $e_i \in E_i$, $i \in \mathbb{N}$. Fig. 4 illustrates an example of multi-labeled image containing 4 CCs where each CC has a different color. We illustrate the E_0 and E_i edges with

black and red color, respectively. The edges are partitioned as follows:

$$E = E_0 \cup E_i \qquad (2)$$

2.2. Selecting the CKs

Selecting the CKs is the main procedure in building the irregular pyramid. In construction of the pyramid, a CC at the base level will be reduced into a single vertex at top of the pyramid. In other words, all vertices of a CC will be contracted through the pyramid until to reach a corresponding surviving vertex at the top level. To this aim, we select the CKs only from the E_0 edges. In addition, in order to select a unique set of CKs, a **total order** is used over the indices of vertices [1, 2]. Consider the corresponding graph G of the 2-Gmap with M by N vertices. Let $(1, 1)$ be the coordinate of the vertex at the upper-left corner and (M, N) at the lower-right corner. The vertices of G receive a unique index as follows:

$$Idx : [1, M] \times [1, N] \mapsto [1, M \cdot N] \subset \mathbb{N} \qquad (3)$$
$$Idx(r, c) = (c - 1) \cdot M + r \qquad (4)$$

The total order has two main properties [8]. First, any two elements (indices of vertices) are comparable. Second, every subset of vertices has one minimum and one maximum. Since the CKs are selected from E_0, a neighborhood $\mathcal{N}(v)$ is defined as follows [1]:

$$\mathcal{N}(v) = \{v\} \cup \{w \in V | e_0 = (v, w) \in E_0\} \qquad (5)$$

If the neighborhood has at least one member ($|\mathcal{N}(v)| > 1$), then the surviving vertex is selected as follows [1]:

$$v_s = \arg\max\{Idx(v_s) | v_s \in \mathcal{N}(v), |\mathcal{N}(v)| > 1\} \qquad (6)$$

Because there is only one maximum number in every subset of the total order, there is only one unique surviving vertex for each non-surviving vertex.

3. Redundant edges in multi-labeled images

Graphs as a versatile representative tool may have many unnecessary edges [1, 2]. The definition of these unnecessary edges is different based on the specific application. In this paper, we study the redundant edges in multi-labeled images. In particular, a new formalism is defined to detect the redundant edges in the hierarchical structure of the irregular pyramid.

In constructing the irregular pyramid [6, 10], the neighborhood graph of an input image forms the base level of the pyramid. To reach the smaller graph at the upper level, a set of vertices are selected for contractions. The contraction operation reduces the number of vertices and edges in the resulting graph. The resulting graph may have empty self-loops or double edges that we define later as redundant
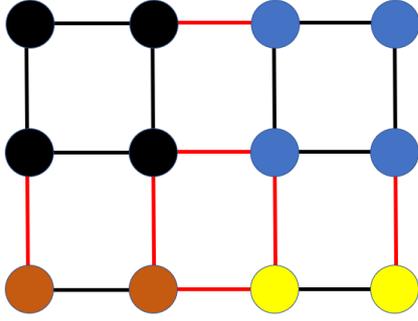
Figure 4. Edge classification in a 3×4 multi-labeled neighborhood graph.

edges. The simplification procedure removes these redundant edges after the contractions. The edge contraction and edge removal are consecutively performed till the pyramid reaches to its top level [11, 12]. However, the simplification procedure can be performed before the contraction operation [4] where it facilitates the construction of the irregular pyramid.

In [1,4] the redundant edges in binary images are defined as follows:

Definition 6 (Redundant-Edge (RE)) *In an empty face, the non-oriented edge incident to the vertex with lowest Idx is redundant iff:*

- *The empty face is bounded by only non-oriented edges with the same contrast value.*

- *The empty face is bounded by non-oriented edges with the same contrast value and oriented edges.*

By defining the new edge classification in Sec 2.1, the definition of redundant edges of binary images would be valid for the corresponding graph of the multi-labeled image. Fig. 5 shows all possible configurations of E_0 and E_i in a face of degree 4 in the grid structure. The right column of this figure illustrates the resulting graph after the edge contraction. The RE after the contraction are either empty self loops or one of the double edges of a face of degree 2.

4. Removing redundant 1-cells

In the previous section, it was shown that the RE can be predicted before constructing the pyramid. Since these RE have no rules in pyramid construction, they can be removed without harming the structure. Therefore, removing the RE reduces the memory space of the pyramid. In a binary image it is proved that up to 50% of the edges are redundant [4]. Considering the E_i edges as the category of

Figure 5. The configuration of all possible redundant edges in a face of degree 4.

E_1 edges, it is concluded that the upper bound of the RE in the multi-labeled images is 50% as well.

Having sufficient independent processing elements, the redundant edges are removed with parallel $\mathcal{O}(1)$ complexity [3,4]. To this aim, a set of independent edges (darts) are selected to be removed at the same time. By definition, two edges not sharing an endpoint are considered as independent edges [4]. Therefore, redundant edges or equivalently redundant 1-cells in the 2-Gmap are removed in a constant time.

8

	#Images	size	$|RE_\mu|$	std($|RE|$)
RE	120	1350×1142	48.43%	1.04

Table 1. The amount of redundant edges (RE) in multi-labeled image.

5. Results

A 2-Gmap is completely defined by encoding its $\alpha_i, i = 0, 1, 2$ involutions. We have shown in Sec .2 that by only preserving the α_1 darts the 2-Gmap is completely encoded. By using the canonical encoding [17], the memory consumption is equal to the size of the initial generalized map independent to the number of pyramid's level. To build up the whole pyramid and use only darts at the base level, the history of contractions is preserved in a 1D array of darts. Two operations of the pyramid, edge contraction and edge removal, modify the α_1 while the α_0 and α_2 do not change in the entire pyramid. By detecting the redundant edges (darts), we put all the redundant darts on the left side of the array. These redundant darts have no role in constructing the pyramid. Fig. 6 shows an example of a 2-Gmap where the array of $\alpha_1(d)$ encodes the entire of the 2-Gmap.

The redundant edges are illustrated by dashed-line in Fig. 6-b. These redundant edge (darts) are highlighted in the array of Fig. 6-d. By putting the redundant darts into the left side of the array, the remaining darts preserve the structure of the simplified 2-Gmap. As it was proved (Sec .4) up to 50% of the whole darts in a 2-Gmap would be redundant.

To exploit the advantage of the proposed method in a real application, we calculate the percentage of RE through a labeled 2D cross slice of a leaf scan (Fig. 7). The multi-labeled input image (Fig. 7) has six different labels illustrating different regions inside the leaf. The size of the original 2D slice is 2560×2560 and there are 2160 slices in the volume of the 3D imaging. After cropping the unnecessary parts of the original image, the proposed algorithm was tested over 120 multi-labeled images[3] with the size 1350×1142.

Tab .1 displays the outcome of the proposed method. The first column shows number of images (#Images) of our multi-label data base. The second column displays the size of the 2D input image. The last two columns give the average amount of RE ('$|RE_\mu|$') along with the standard deviation ('std($|RE|$)') over all images. The results show that the proposed method enormously reduces the size of the input image approximately by half.

---

[3]The images are from the *Water's gateway to heaven* project, https://waters-gateway.boku.ac.at/

6. Conclusion

The paper presents a novel formalism to define redundant 1-cells in the 2-Gmap of a multi-labeled image. It defines a corresponding graph of the 2-Gmap and detects the redundant edges in the graph. The obtained formalism then translated into the 2-Gmap structure where the redundant 1-cells are detected. We proved that up to half of the whole 1-cells (edges) would be redundant in theory. Having sufficient processing elements by employing the set of independent edges, all the redundant edges (1-cells) are removed in constant complexity. The experiments show almost 48% of the 1-cells in the 2-Gmap are structurally redundant on average. By removing these redundant 1-cells the memory consumption is dramatically reduced. Moreover, we introduced an efficient encoding of involutions in the 2-Gmap where the two third of the involutions can be implicitly encoded. Finally, using the generalized map structure the proposed method can be extended to higher dimensional n-Gmaps.

References

[1] Majid Banaeyan, Darshan Batavia, and Walter G. Kropatsch. Removing redundancies in binary images. In *International Conference on Intelligent Systems and Patterns Recognition (ISPR), Hammamet, Tunisia, March 24-25, 2022*, pages 221–233. Springer, 2022.

[2] Majid Banaeyan and Kropatsch Walter G. Fast Labeled Spanning Tree in Binary Irregular Graph Pyramids. *Journal of Engineering Research and Sciences*, 1(10):69–78, 2022.

[3] Majid Banaeyan and Walter G. Kropatsch. Pyramidal connected component labeling by irregular graph pyramid. In *2021 5th International Conference on Pattern Recognition and Image Analysis (IPRIA)*, pages 1–5. IEEE, 2021.

[4] Majid Banaeyan and Walter G. Kropatsch. Parallel $\mathcal{O}(log(n))$ computation of the adjacency of connected components. In *International Conference on Pattern Recognition and Artificial Intelligence (ICPRAI), Paris, France, June 1-3, 2022*, pages 102–113. Springer, 2022.

[5] Luc Brun and Walter Kropatsch. Introduction to combinatorial pyramids. In *Digital and Image Geometry*, pages 108–128. Springer, 2001.

[6] Luc Brun and Walter G. Kropatsch. Hierarchical graph encodings. In Olivier Lézoray and Leo Grady, editors, *Image Processing and Analysis with Graphs: Theory and Practice*, pages 305–349. CRC Press, 2012.

9

(a) 2-Gmap   (b) redundant darts (dashed line)   (c) the resulted 2-Gmap
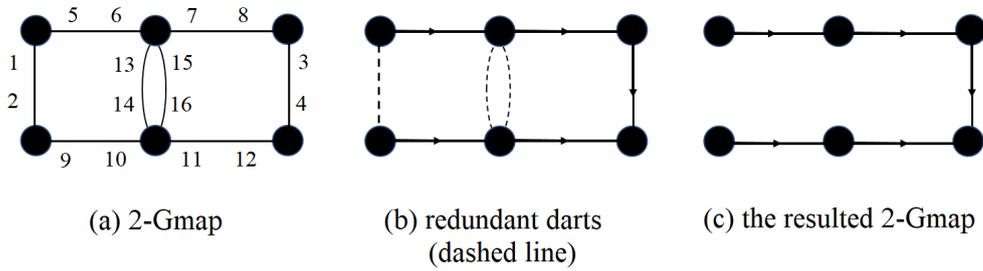
d	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\alpha_1(d)$	5	9	8	12	1	13	15	3	2	14	16	4	6	10	7	11

(d) array of darts in memory

d	1	2	13	14	15	16	3	4	5	6	7	8	9	10	11	12
$\alpha_1(d)$	5	9	6	10	7	11	8	12	1	13	15	3	2	14	16	4

redundant darts   required darts to preserve 2-Gmap in (c)

(e) removing the array of redundant darts

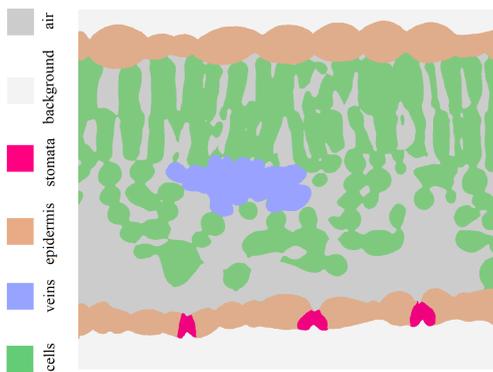Figure 6. Removing redundant darts in the canonical encoding

Figure 7. Multi-labeled input image

[7] Guillaume Damiand and Pascal Lienhardt. *Combinatorial maps: efficient data structures for computer graphics and image processing*. CRC Press, 2014.

[8] Brian A Davey and Hilary A Priestley. *Introduction to lattices and order*. Cambridge university press, 2002.

[9] Anand Deshpande and Manish Kumar. *Artificial intelligence for big data: complete guide to automating big data solutions using artificial intelligence techniques*. Packt Publishing Ltd, 2018.

[10] Walter G. Kropatsch. Building irregular pyramids by dual graph contraction. *IEE-Proc. Vision, Image and Signal Processing*, Vol. 142(No. 6):pp. 366–374, 1995.

[11] Walter G Kropatsch, Yll Haxhimusa, and Pascal Lienhardt. Hierarchies relating topology and geometry. In *Cognitive Vision Systems*, pages 199–220. Springer, 2006.

[12] Walter G. Kropatsch, Yll Haxhimusa, Zygmunt Pizlo, and Georg Langs. Vision pyramids that do not grow too high. *Pattern Recognition Letters*, 26(3):319–337, Feb. 2005.

[13] Pascal Lienhardt. Topological models for boundary representation: a comparison with n-dimensional generalized maps. *Computer-aided design*, 23(1):59–82, 1991.

[14] Peter Meer. Stochastic image pyramids. *Computer Vision, Graphics, and Image Processing*, 45(3):269–294, 1989.

[15] Alessandro Negro. *Graph-powered machine learning*. Simon and Schuster, 2021.

[16] Azriel Rosenfeld and Mark Thurston. Edge and curve detection for visual scene analysis. *IEEE Transactions on computers*, 100(5):562–569, 1971.

[17] Fuensanta Torres and Walter G. Kropatsch. Canonical encoding of the combinatorial pyramid. In *Proceedings of*

the 19th Computer Vision Winter Workshop, pages 118–125, 2014.

[18] R.J. Trudeau. *Introduction to Graph Theory*. Dover Books on Mathematics. Dover Pub., 1993.