DOI 10.4467/21995923GP.18.009.9639 GEOINFORMATICA POLONICA
17: 2018

Justyna Dębicka, Stanisław Szombara

COMPARATIVE ANALYSIS OF ARCGIS AND QGIS IN TERMS OF THE TRANSFORMATIONS' RUNTIME

AGH University of Science and Technology, Faculty of Mining Surveying and Environmental Engineering,
Department of Integrated Geodesy and Cartography
AGH Akademia Górniczo-Hutnicza w Krakowie, Wydział Geodezji Górniczej i Inżynierii Środowiska,
Katedra Geodezji Zintegrowanej i Kartografii

Keywords: ArcGIS, QGIS, GIS, spatial analysis runtime

Abstract

These days there is a variety of software on the market that enables spatial analysis. Spatial data has been becoming increasingly large. More and more, the analyses are done repetitively at a mass-scale and consist of many distinct transformations. Thus, the time of single one proves to be essential. The aim of the research is to compare the execution time of the selected transformations between two geographical information system programs: ArcGIS Desktop 10.5.1 vs. QGIS 2.18.20 ‚Las Palmas'. Buffer, convex hull and intersection were selected as transformations. The measurements were carried out on a specially prepared representative dataset for GIS vector analyses. At the data exploration stage, the influence of size, geometry type, no. of vertices/objects of the input data and the characteristics of the output data on the measured time were examined. In addition, computational complexity was investigated for the point layers. The results of the experiment can be taken into account when selecting the program that will be the most suitable for a particular GIS user.

ANALIZA PORÓWNAWCZA CZASU WYKONANIA PRZEKSZTAŁCEŃ GEOMETRYCZNYCH W PROGRAMACH ARCGIS I QGIS

Słowa kluczowe: ArcGIS, QGIS, GIS, czas wykonania analiz przestrzennych

Abstrakt

Współcześnie istnieje na rynku wiele programów umożliwiających przeprowadzenie analiz przestrzennych. Coraz częściej analizy wykonywane są iteracyjnie oraz na masową skalę a przetwarzane dane mają większe rozmiary. Ponadto tworzy się rozbudowane modele rzeczywistości składające się z wielu pojedynczych analiz przestrzennych. Z tego powodu czas wykonania takiej pojedynczej analizy jest istotny. Celem badania jest porównanie czasu wykonania wybranych przekształceń geometrycznych między dwoma programami systemu informacji geograficznej, tj. ArcGIS Desktop 10.5.1 oraz QGIS 2.18.20 „Las Palmas". Jako przekształcenia wybrano ekwidystantę, otoczkę wypukłą i nakładanie. Pomiary przeprowadzono na specjalnie przygotowanym reprezentatywnym w analizach przestrzennych zbiorze danych. Na etapie eksploracji danych sprawdzono wpływ rozmiaru, typu geometrii, liczby wierzchołków i liczby obiektów danych wejściowych oraz charakterystykę danych wyjściowych na uzyskany w pomiarze czas analizy. Dodatkowo dla warstw punktowych zbadano złożoność obliczeniową. Wyniki eksperymentu mogą być brane pod uwagę podczas wyboru programu, który będzie najbardziej odpowiedni dla konkretnego użytkownika GIS.

1. INTRODUCTION

Currently, in the era of information society, due to modern technologies of data processing, transmission and storage, data with spatial reference are becoming more and more important. The analysis of the data provides a lot of interesting information that would be invisible had only numerical data been taken into account. Systems which handle collecting, storing, processing and visualization of such data are called Geographic Information System (GIS). The GIS software is a specialistic tool to fulfil the above-mentioned tasks. Moreover, spatial information has become so vital that spatial extensions have been added to ordinary database management systems (DBMS). The functionality of these extensions vastly improves each year. Thus, spatial DBMS begin to resemble GIS software.

Both GIS and spatial DBMS can perform spatial analyses. One of the types of such analyses is transformation. There are comparisons of GIS software, i.e. (Smith, Lazzarato and Carette, 2018), within the scope of installation process, maintainability, interoperability etc. The aim of the research is to perform a comparative analysis of two GIS programs, i.e. ArcGIS Desktop 10.5.1 (ed. 29.06.2017) and QGIS Desktop 2.18 'Las Palmas' (ed. 20.05.2018), except in terms of the execution time of the chosen transformations for the dataset representative for common GIS tasks. The research was conducted as a part of a master thesis and is inspired by (Piórkowski and Krawczyk, 2011), where authors are examining the influence of the objects' generalisation for the execution time of two queries in spatial DBMS, i.e. MySQL Spatial and PostGIS.

The results of the experiment should be helpful both for private users of tested software as well as for smaller enterprises since the execution time of a single analysis has lately become crucial to effective workflow. This is caused by progressively larger sizes characterising the processed data (the so-called *Big Data*) and complex models of the reality consisting of many individual spatial analyses. Moreover, this comparison of the GIS programs is particularly important due to the fact that one is commercial and the other is open source, which means that it can be used to create customized GIS software for free.

1.1. The speed of algorithms

The theory of computation deals with computational complexity (CC), which serves estimation of the memory capability and time efficiency of the algorithms. CC is used because it does not depend on changing factors such as the size of the RAM module, processor nor the type of the disk. It allows matching a given algorithm to a specific class – model. The speed of such models is not measured in time units. At the beginning it should be determined how many operations are carried out on a model, then the adjustment of the function describing their number depending on the input data of the algorithm should follow. By examining the performance of algorithms, these functions are compared to each other. In other words, the speed of the algorithms is recognised by how quickly the amount of work increases with the increasing number of operations. In practice however, instead of determining the exact CC, only the order of its magnitude is estimated. Parameters that describe this are notations presented by specific functions g(n), where n is the number of input data elements, c – a constant:

- Notation O (big O notation) – $\forall n \geqslant n0 : f(n) \leqslant c * g(n)$,
  Notation Θ – $\forall n \geqslant n0 : c1 * g(n) \geqslant f(n) \geqslant c2 * g(n)$
- Notation Ω – $\forall n \geqslant n0 : f(n) \geqslant c * g(n)$,

Often the speed of algorithms is presented in the big O notation, which determines the most pessimistic case of the number of performed operations. The popular functions in the big O notation are i.e. linear O(n), logarithmic O(log n), factorial O(n!), linear-logarithmic (nlog n) etc. It is apparent that the algorithm with the notation $O(\log_2 n)$ is much faster than O(n) since the function is asymptotic. Even if two algorithms have the same big O notation, in practice they can sometimes differ by a certain constant value c e.g. c*O(n). On the other hand, if two algorithms have different notation, the constant is usually insignificant, especially when dealing with a large number of input data elements. It is also worth mentioning the average speed of the algorithm. It might happen that one algorithm has a better big O notation, but on average it is slower (notation Θ). The above considerations show the importance of a deliberate construction and application of the algorithms in order to obtain an optimal solution in terms of the execution time (*Notacja dużego O – Encyklopedia Algorytmów*, 2018; *Podstawy złożoności* *obliczeniowej – Samouczek Programisty*, 2018; Bhargava, 2017)

The field of science regarding this issue within the scope of geometric algorithms, which are commonly used in GIS, is computational geometry. It handles data structures and algorithms for geometric objects and searches for fast, accurate and asymptotic algorithms. The algorithm is asymptotic when the execution time grows slowly as the number of objects increases.

Currently, there exists a large set of geometric algorithms to choose from. In order to solve the geometric problem of an algorithm one should understand its geometrical properties. The next step is the appropriate usage of data structures and algorithmic techniques (de Berg *et al.*, 2007).

1.2. Chosen transformations

Spatial analyses include transformations (Longley *et al.*, 2006). Ultimately, their goal is to find regularities that were not previously visible in the input data. In this experiment, three types of transformations were used to test the performance of selected programs: buffer, convex hull, and intersection. Below, only the vector data of the transformations is discussed.

Buffer is a transformation which is applicable to layers with point, line and polygon objects. The result of the analysis is the layer with a polygon object(s) whose edge is equal (or for the computational reasons minimally smaller) to the distance set by the user from the input object. The algorithm for calculating the buffer for the point layer has a **linear** function in the big O notation. That transformation facilitates classifying e.g. the impact zones of a given phenomenon or the protection area. Moreover, it is commonly used as an element of the model to find the best location. Depending on the software used, this analysis may have various additional parameters (*How Buffer (Analysis) works—Help | ArcGIS for Desktop*, 2018; *Vector Spatial Analysis (Buffers)*, 2018).

Convex hull is an analysis that allows the calculation of the smallest convex polygon containing all the vertices that define the input object(s) (Figure 1). The boundary of the polygon consists of existing vertices of the input data. In (de Berg *et al.*, 2007) in a pseudocode the authors present two different algorithms for calculating convex hull – slow and fast. In the "slow" algorithm, big O notation has a **polynomial** function $O(n^3)$. The polynomial function is not asymptotic, which indicates the slow operation of this algorithm. The function describing the big notation O of the "fast" algorithm is **linear-logarithmic** $O(n \log_2 n)$.

Intersection is a transformation in which a new object(s) is created that is a common part of the two (pos-

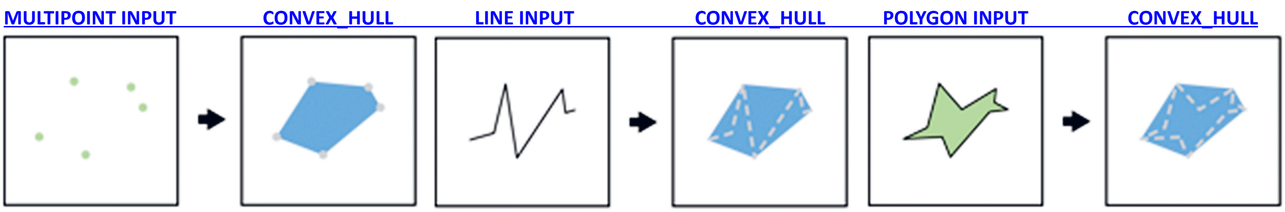

Fig. 1. Convex hull output (*Minimum Bounding Geometry—Data Management toolbox | ArcGIS Desktop*)
Rys. 1. Wynik działania otoczki wypukłej (*Minimum Bounding Geometry-Data Management toolbox | ArcGIS Desktop*)

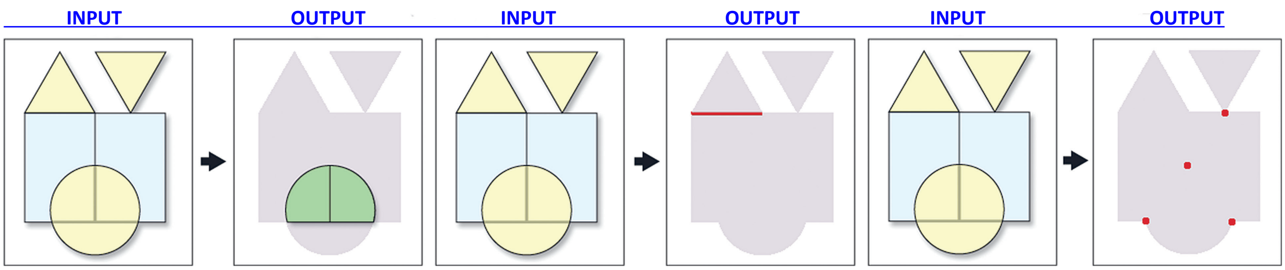

Fig. 2. Intersection of two polygon layers; polygon/polyline/point output (*How Intersect works*)
Rys. 2. Nakładanie dwóch warstw poligonowych; poligonowy/liniowy/punktowy wynik (*How Intersect works*)

sibly more) input geometry objects. The resulting object belongs to a geometry class of input dataset objects with a smaller geometry dimension (Figure 2). Intersection algorithm inserts new vertices at the places where objects intersect.

1.3. Software characteristics

ArcGIS is a complex, commercial GIS software created by ESRI – a market leader in many countries. ArcGIS for Desktop is a software designed for a desktop user. The workflow in the program can be automated by using the Python console in the user interface (UI) or in an external IDE after importing the arcpy module and using Python installed with the program. ArcGIS Desktop is compatible with Python e.d. 2.X. Created scripts can be loaded as an additional tool for the software [9, 10].

QGIS is an open source GIS software, whose development as a social project began in 2002. Version 1.0 was released in 2009. The first meeting of QGIS users in Poland took place on 19.06.2018 at the Cracow University of Technology. This program is written in C ++, based on GDAL / OGR and GEOS libraries and uses the Qt library for the UI. The speed of an operation is very good, it even surpasses other programs of this type available on the market (Lawhead, 2013). There is a Python console in QGIS where one can perform instructions. It is also possible to create scripts and test them in an external IDE. Due to the fact that all software is open source, virtually everything can be adapted to the user's own needs with the application of the published source code ('PyQGIS developer cookbook', 2013; *QGIS Python Programming Cookbook – Second Edition*, 2017). QGIS ed. 2.X is compatible with Python version 2.X.

2. METHODOLOGY

As a part of the study the GIS programs' desktop versions were selected. The choice was induced by numerous queries on the Internet forums about the superiority of one software over the other, as well as the emerging comparative articles, e.g. (*27 Differences Between ArcGIS and QGIS – The Most Epic GIS Software Battle in GIS History – GIS Geography*). As the transformations buffer, convex hull and intersection were tested due to being frequently used as part of the more complex models. In the experiment **buffer** is performed separately for each object and its distance is equal to 500 m. **Convex hull** is calculated for all objects together. For the above analyses, it was decided to examine the notation function for point layers. In the **intersection** analysis, a singular polygon in the shape of a triangle was used as the second argument of the function and the input and output data is stored on an external disk in the ESRI Shapefile format. Spatial indexes were created for the dataset and the output, i.e. for ArcGIS – **grid** and for QGIS – **quadtree**.

In the experiment selected transformations were performed in the tested software on the prepared dataset consisting of 29 layers in the ESRI Shapefile format (projection: EPSG 2180). The layers varied in number of vertices/objects, size and type of geometry in order to observe overall patterns in the selected software's ways of conduct depending on the particular data types. Part of the data was artificially generated, whereas the rest was downloaded from the Polish Central Centre for Geodetic and Cartographic Documentation (pol. CODGIK).

The names of layers in the set indicate their geometry: P – stands for point, L – polyline, A – polygon. For each type of geometry, the layers were numbered consecutively according to the increasing number of objects. Layers P8, P10, L19, L20, A28, A29 have regular shapes, i.e. grid, parallel lines etc. The size of the data files ranges from 0.006 to 59.878 MB, while the number of vertices ranges from 235 to 2 242 375. The smallest number of objects totals 11 for A21. Examples of selected layers are shown in the figure (Figure 3). All unnecessary attributes have been removed.

Python programming language was chosen as a time-testing tool since it has recently become very popular in various fields, not only in the GIS environment (*The Best Programming Languages for GIS – FreelancingGig Blog – Freelancer Job Tips and Hiring Insights*; Garbade). The usage of this language also enables exploration of the data results of the process along with visualization (Gągolewski, Bartoszuk and Cena, 2016).

The time of the transformations for each layer was measured iteratively – for the smaller sized layers 50 times, for the larger ones less due to the fact that disturbances in the time measurement caused by background processes are relatively smaller. Finally, the minimal measured value, considered the least disturbed, was viewed as the analysis time (Pilgrim, 2004). Buffer analysis was performed on a PC with a Windows 10 64bit operating system with an Intel® Core ™ i5-6500

a) b) c)

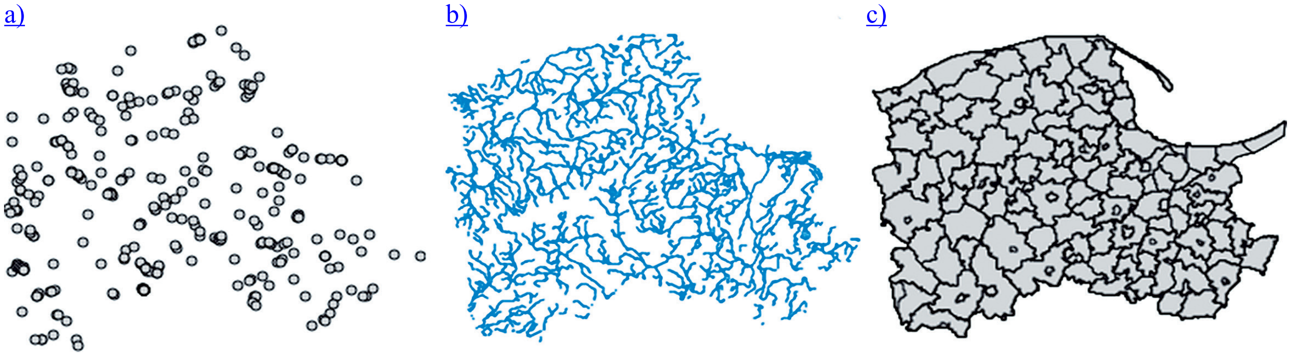

Fig. 3. Examples of the input data; a) P1-P7, P11-randomly generated points; b) L13-16 – polish network of streams; c) A23-A27 – administrative division in Poland

Rys. 3. Przykładowe dane wejściowe; a) P1-P7, P11 – losowo wygenerowane punkty; b) L13-16 – sieć rzeczna w Polsce; c) A23-A27 – podział administracyjny w Polsce

3.2 GHz processor equipped with 32 GB of RAM and SSD. The remaining analyses were carried out on a PC with a Windows 10 64bit operating system with an Intel® Core ™ i3-3110M 2.4 GHz processor with 6 GB RAM and HDD disk, which was defragmented before performing the analyses. The defragmented HDD Seagate 7200 rpm 2 TB drive with USB 3.0 for fast data transfer was also used.

3. RESULTS

The analyses' time obtained in the experiment are shown in diagrams (Figure 4–6). Because of the prolonged time of calculating buffer in ArcGIS, no measurements were made for the 6 layers of the largest size, i.e. 17–59 MB. In the intersection analysis in ArcGIS results for L19 and L20 were not calculated, despite waiting for them over 7 hours and calling the analysis several times. The layers are a regular line grid and L19 is a slice of L20. This was most likely caused by the occurrence of a degenerate case during the calculations, which was not included in the algorithm used by the software.

On average the time of calculating **buffer** in QGIS is equal to 30% of its time in ArcGIS, and for the layers with the longest time: 12%. The difference may be caused by the saving speed of the results to the ESRI shapefile file. However, it could be an outcome of different programs' default number of segments used to approximate a quarter circle (Figure 7). Calculation of

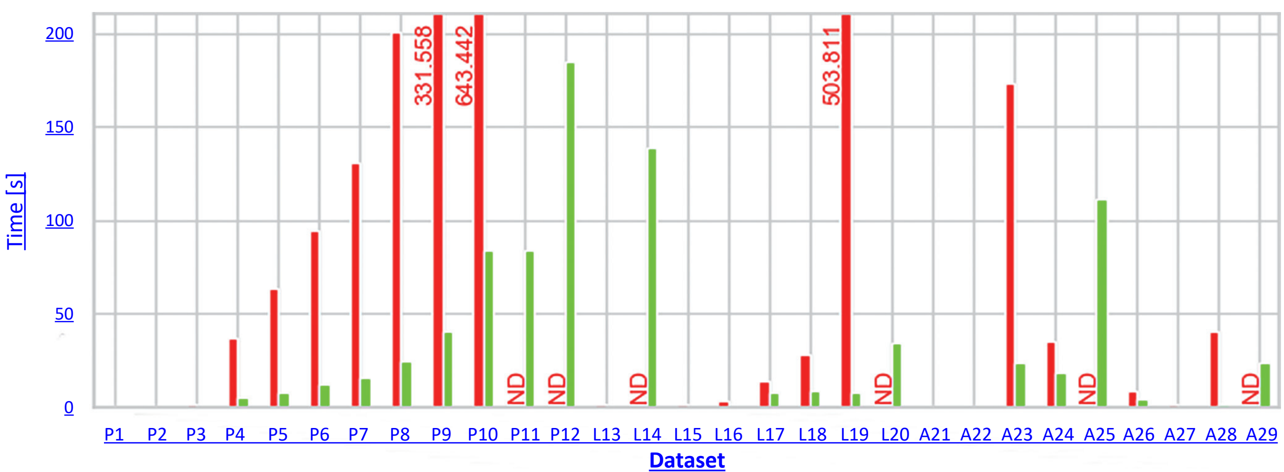

Fig. 4. Buffer analysis time in ArcGIS (red) and QGIS (green); ND – No Data

Rys. 4. Czas ekwidystanty w ArcGIS (czerwony) i QGIS (zielony); ND – brak danych

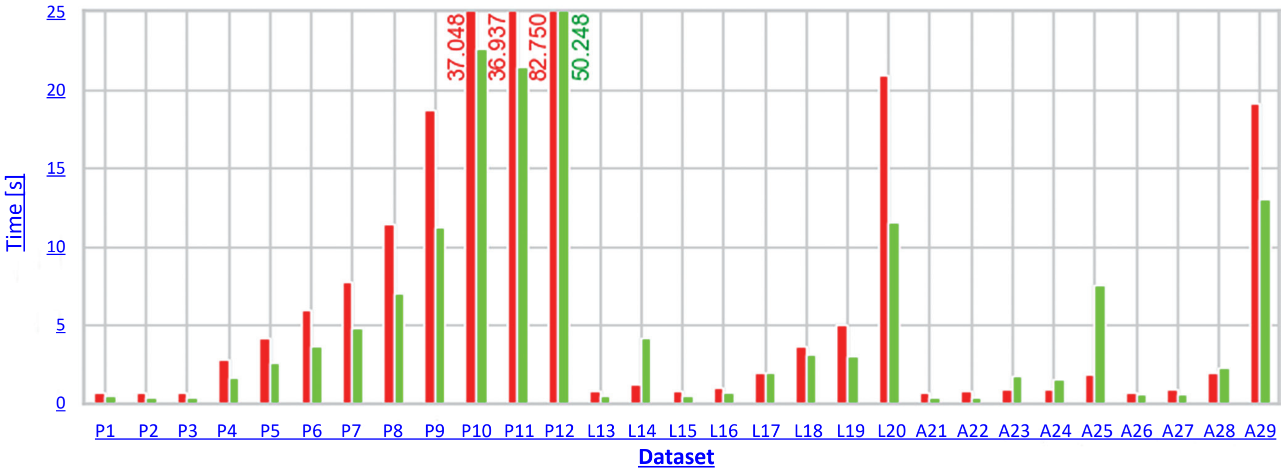

Fig. 5. Convex Hull analysis time in ArcGIS (red) and QGIS (green)
Rys. 5. Czas otoczki wypukłej w ArcGIS (czerwony) i QGIS (zielony)

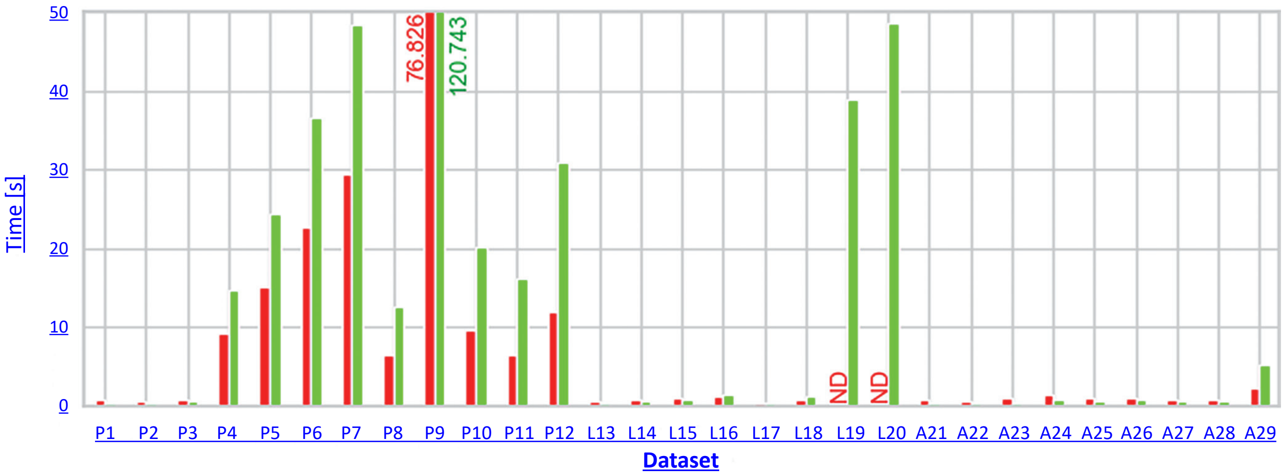

Fig. 6. Intersection analysis time in ArcGIS (red) and QGIS (green); ND – No Data
Rys. 6. Czas nakładania w ArcGIS (czerwony) i QGIS (zielony); ND – brak danych

the **convex hull** was 79% faster in QGIS and measured time constituted on average 63% of the analysis time carried out in ArcGIS. For the remaining layers, the analysis time in QGIS was on average twice as large as in ArcGIS. The common feature of these layers was their area, which was the whole country instead of only one province. It is likely that this diversity is due to the spatial index used in the program. In the **intersection** analysis, QGIS came out a bit less favourably – 56% faster results and the time obtained in QGIS was on average 66% of the calculation time of the ArcGIS. Generally, for the points layers the analysis was faster in ArcGIS whereas for the polygon layers in QGIS. It is apparent that the effectiveness of spatial indexes depends on the characteristics of input data. Overall, 73% of the three transformations made on the same layers were faster in QGIS

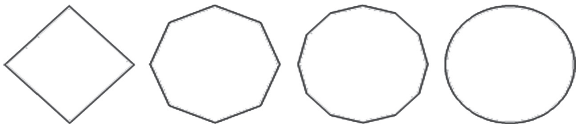

Fig. 7. Number of segments used to approximate a quarter circle: 1, 2, 3, 30
Rys. 7. Liczba segmentów ćwiartkowych aproksymujących koło: 1, 2, 3, 30

At the stage of data mining, illustrative charts were created showing the dependence of the obtained time from measurements on the geometry of the input data and the no. of vertices, the no. of objects and the size of the data. The most important part of them is presented in the charts (Figure 8–10). Due to the lack of visible connections in **intersection**, the charts for that analysis are not shown. For each analysis, the different behaviour of layers with distinct geometry is noticeable. This is best seen on the point layers from which deduction of notation is

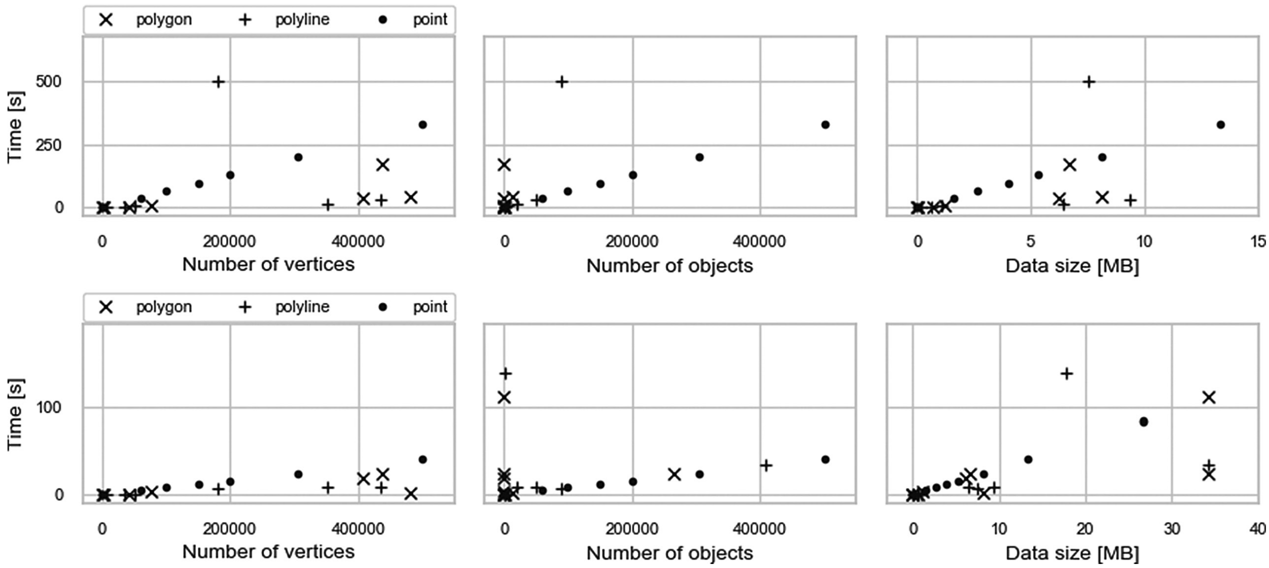

Fig. 8. Buffer; ArcGIS (top), QGIS (bottom); analysis time depending on the input geometry, no. of vertices/objects and data size
Rys. 8. Ekwidystanta; ArcGIS (góra), QGIS (dół); czas analizy w zależności od geometrii, liczby wierzchołków/obiektów i rozmiaru danych wejściowych

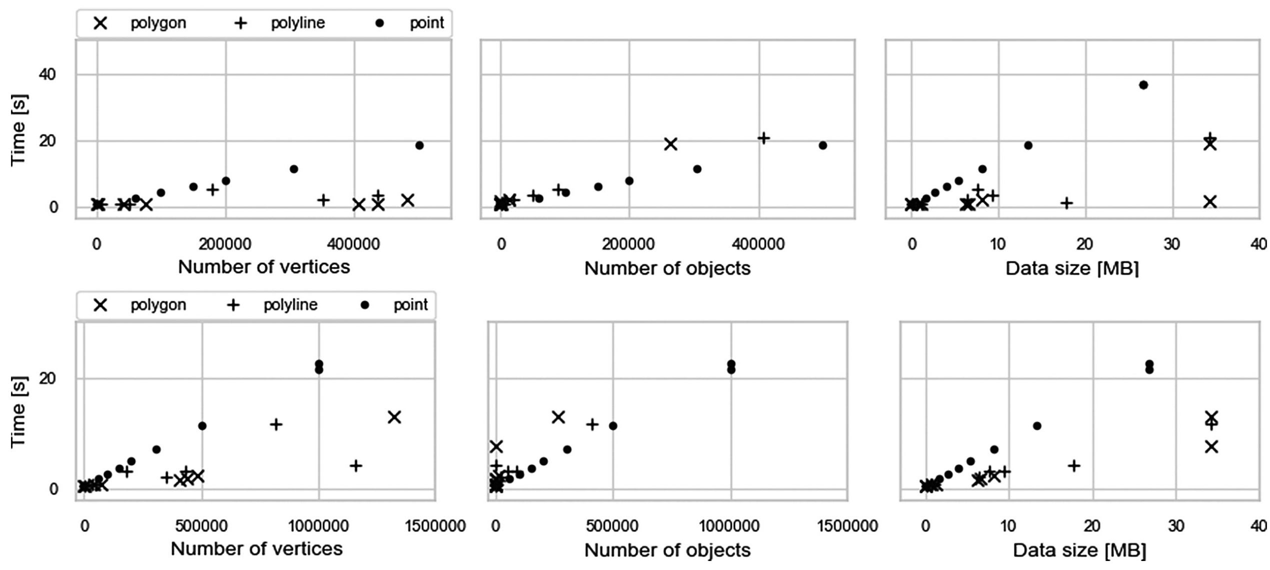

Fig. 9. Convex hull; ArcGIS (top), QGIS (bottom); analysis time depending on the input geometry, no. of vertices/objects and data size
Rys. 9. Otoczka wypukła; ArcGIS (góra), QGIS (dół); czas analizy w zależności od geometrii, liczby wierzchołków/obiektów i rozmiaru danych wejściowych

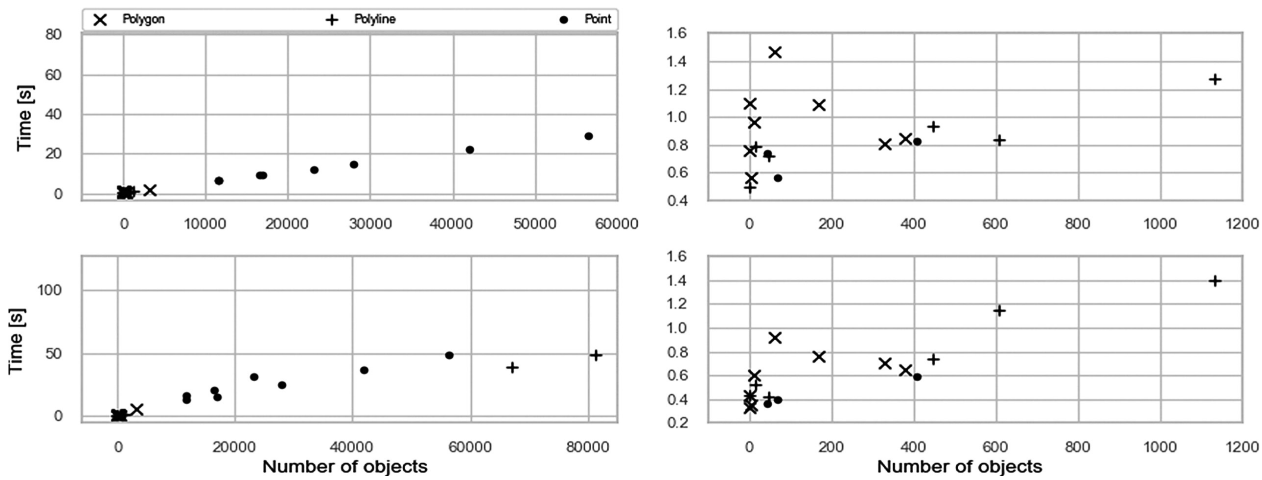

Fig. 10. Intersection; ArcGIS (top), QGIS (bottom); analysis time depending on the input geometry and no. of the output objects; the charts in the second column are the zoomed versions of the charts in the first column
Rys. 10. Nakładanie; ArcGIS (góra), QGIS (dół); czas analizy w zależności od geometrii i liczby obiektów wynikowych; wykresy w drugiej kolumnie są przybliżeniem wykresów w pierwszej

Tab. 1. Intersection – number of the output objects
Tab. 1. Nakładanie – liczba wynikowych obiektów

Dataset	**L17**	**A21**	**A23**	**A22**	**A25**	**L14**	**P1**	**L13**	**A24**	**P2**	**A26**	**A28**	**A27**	**P3**	**L15**
Objects	0	1	1	3	10	15	43	49	60	68	167	329	379	409	443
Dataset	**L18**	**L16**	**A29**	**P11**	**P8**	**P10**	**P4**	**P12**	**P5**	**P6**	**P7**	**L19**	**L20**	**P9**	
Objects	607	1130	3240	11586	11586	16461	16890	23106	27990	41984	56404	67106	81316	139752	

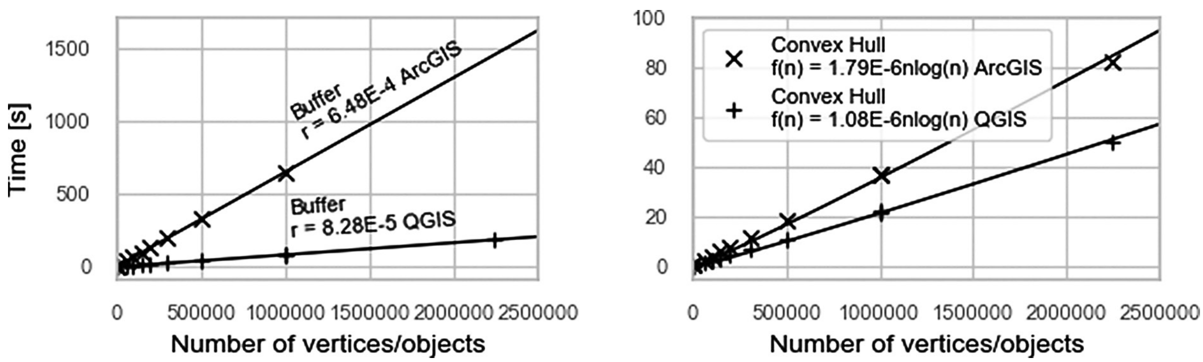

Fig. 11. ArcGIS, QGIS; point data set; time complexity of the buffer (left) and Convex hull (right) algorithm
Rys. 11. ArcGIS, QGIS; punktowe warstwy testowe; złożoność obliczeniowa algorytmu ekwidystanty (lewy wykres) i otoczki wypukłej (prawy wykres)

possible. To accurately examine notation in cases of more complex geometries more test data would be required.

In the case of the buffer analysis (Figure 8), the time is clearly dependent on the geometry of the input data and the number of the vertices, number of which is directly proportional to the size of the data. In ArcGIS, the linear, regular L19 layer is an outlier in terms of time-out value.

In the convex hull analysis, it can be noticed (Figure 9) that in ArcGIS there is a dependence of time on the number of input data objects regardless of the other characteristics of the data, whereas in QGIS the greatest influence on the result has the size of the input data and its geometry.

Due to the possibility of varying the speed of saving the results, it was additionally checked whether the number of vertices in the result object of convex hull and intersection is significant to the calculation. In the intersection analysis the impact of the no. of result objects was also examined. From the above tests, only the dependence of the number of result objects on the time measured in intersection was detected (Figure 10) for both programs. In (Table 1) there is a list of number of the output objects in ascending order.

The results of the function of the notation testing for point layers in the buffer and the convex hull are presented in the diagrams (Figure 11). The algorithm for calculating the **buffer** for n vertices of input data in both programs has a **linear** notation. The probable execution time of the buffer in ArcGIS for the P12 layer, which has 2 242 375 points, is 24.21 min whereas the time in QGIS is only 3.09 min. The **convex hull** algorithm has likely a **linear-logarithmic** notation.

4. CONCLUSION

The purpose of the research, which was to compare the execution time of selected spatial transformations for the GIS software was achieved. Exploration of the obtained results revealed the diversity of the measured values, dependent on the geometry of the input data, the number of their vertices/objects and, in case of intersection, also from the characteristics of the output data. It should be noted that the software is often characterised by different behaviour. These differences are most likely caused by programming language in which they are written, its architecture, applied spatial index, used algorithm and taking into account degenerate cases in a given algorithm. In the case of buffer, the analysis' default parameter i.e. number of segments used to approximate a quarter circle might be meaningful. The applied index influence is particularly visible in intersection.

While comparing the GIS programs, QGIS 2.18.20 "Las Palmas" was performing significantly better than ArcGIS Desktop 10.5.1. Among the analyses carried out on the same dataset, 73% had shorter time in QGIS. The result can also be caused by the input data format, since the default format for QGIS is used in the experiment shapefile, while the dedicated format for ArcGIS is geodatabase.

It would be valuable to precisely determine patterns in behaviour for the line/polygon layers. To achieve this an artificially controlled dataset with i.e. fixed ratio of vertices to the number of objects should be created. Runtime of the analyses conducted on such input data might manifest usage of modified variants of the algorithms for the specific geometry input.

Obtained results can be the basis for choosing an optimal, in the scope of time and the input data characteristics, spatial data processing software for a particular task performed by a GIS user. Apart from faster analysis performance, the use of QGIS, which has an open code, can help to save company capital.

LITERATURE

27 Differences Between ArcGIS and QGIS – The Most Epic GIS Software Battle in GIS History – GIS Geography. Available at: https://gisgeography.com/qgis-arcgis-differences/ (Accessed: 23 April 2018).

de Berg, M., van Kreveld, M., Overmars, M. and Schwarzkopf, O. (2007) *Geometria obliczeniowa – algorytmy i zastosowania*. II. Warszawa: Wydawnictwo Naukowo-Techniczne.

Bhargava, A. Y. (2017) *algorytmy – Ilustrowany przewodnik*. Gliwice: Helion.

Gągolewski, M., Bartoszuk, M. and Cena, A. (2016) *Przetwarzanie i analiza danych w języku Python*. I. Warszawa: Wydawnictwo Naukowe PWN SA.

Garbade, M. J., *Top 3 most popular programming languages in 2018 (and their annual salaries)*. Available at: https://hackernoon.com/top-3-most-popular-programming-languages-in-2018-and-their-annual-salaries-51b4a7354e06 (Accessed: 7 November 2018).

How Buffer (Analysis) works—Help | ArcGIS for Desktop. Available at: http://desktop.arcgis.com/en/arcmap/10.3/tools/analysis-toolbox/how-buffer-analysis-works.htm (Accessed: 17 May 2018).

How Intersect works—Help | ArcGIS for Desktop. Available at: http://desktop.arcgis.com/en/arcmap/10.3/tools/analysis-toolbox/how-intersect-analysis-works.htm (Accessed: 19 May 2018).

Lawhead, J. (2013) *Learning Geospatial Analysis with Python*.

Longley, P. A., Goodchild, M. F., Maguire, D. J. and Rhind, D. W. (2006) *GIS Teoria i praktyka (GIS Theory and Practice)*. Warszawa: Wydawnictwo Naukowe PWN SA.

Minimum Bounding Geometry—Data Management toolbox | ArcGIS Desktop. Available at: http://pro.arcgis.com/en/

pro-app/tool-reference/data-management/minimum-bounding-geometry.htm (Accessed: 19 May 2018).

Notacja dużego O – Encyklopedia Algorytmów. Available at: http://algorytmy.ency.pl/artykul/notacja_duzego_o (Accessed: 18 May 2018).

Pilgrim, M. (2004) *Dive Into Python*.

Piórkowski, A. and Krawczyk, A. (2011) 'Wpływ generalizacji obiektów na optymalizację zapytań w bazach danych przestrzennych', *STUDIA INFORMATICA*, 32(297), pp. 119––129.

Podstawy złożoności obliczeniowej – Samouczek Programisty. Available at: http://www.samouczekprogramisty.pl/podstawy-zlozonosci-obliczeniowej/ (Accessed: 18 May 2018).

'PyQGIS developer cookbook' (2013) *Handbok*, p. 86.

QGIS Python Programming Cookbook – Second Edition (2017). Available at: https://www.amazon.com/QGIS-Python-Programming-Cookbook-Second/dp/1787124835.

Smith, W. S., Lazzarato, D. A. and Carette, J. (2018) 'State of the Practice for GIS Software'. Available at: https://arxiv.org/pdf/1802.03422.pdf (Accessed: 16 April 2018).

The Best Programming Languages for GIS – FreelancingGig Blog – Freelancer Job Tips and Hiring Insights. Available at: https://www.freelancinggig.com/blog/2017/08/23/best-programming-languages-gis/ (Accessed: 21 May 2018).

Vector Spatial Analysis (Buffers). Available at: https://docs.qgis.org/2.8/en/docs/gentle_gis_introduction/vector_spatial_analysis_buffers.html#vector-spatial-analysis-buffers (Accessed: 30 June 2018).