International Electronic Journal of Mathematics Education
2026, 21(3), em0889
e-ISSN: 1306-3030
https://www.iejme.com

MODESTUM

Research Article

OPEN ACCESS

From paper to software: Teaching polygon-separability problems using BichromaticSolver

Ruben Molano [1,2]*, Mohammadhossein Homaei [2], Mar Avila [2], Pablo G. Rodriguez [2], Andres Caro [2]

[1] International University of La Rioja (UNIR), SPAIN
[2] University of Extremadura, SPAIN
***Corresponding Author:** ruben.molano@unir.net

Citation: Molano, R., Homaei, M., Avila, M., Rodriguez, P. G., & Caro, A. (2026). From paper to software: Teaching polygon-separability problems using BichromaticSolver. *International Electronic Journal of Mathematics Education, 21*(3), em0889. https://doi.org/10.29333/iejme/18916

ARTICLE INFO

Received: 22 Dec. 2025

Accepted: 01 Jun. 2026

ABSTRACT

This paper examines how bichromatic separability problems, a classic topic in computational geometry, can be adapted for secondary mathematics education through the use of BichromaticSolver. The software computes simple or convex polygons that separate two finite sets of points under different optimisation criteria, including maximum area, minimum area, maximum perimeter, or minimum perimeter. Unlike traditional approaches, the number of polygon sides k is not fixed in advance but chosen by the user, enabling the exploration of diverse and potentially more effective configurations. Three classroom tasks were designed in which students alternated between paper-and-pencil methods and digital exploration with the software. This two-phase structure encouraged them to verify constructions, compare alternative outcomes, and refine their strategies. Classroom observations from this exploratory study document how these activities created opportunities for students to express and refine geometric reasoning while making computational thinking (CT)-related practices visible, for example decomposition, abstraction, strategic planning, and comparative evaluation of solutions. The findings suggest that integrating computational geometry problems with digital tools can enrich traditional mathematics instruction, highlight the relevance of geometry in authentic contexts, and offer a promising and transferable context for developing CT alongside core geometry content in secondary mathematics education.

Keywords: geometry education, computational geometry, computational thinking (CT), technology-enhanced learning, area, perimeter

INTRODUCTION

Computational geometry has experienced steady growth over recent decades, driven by its applicability in a wide range of fields such as computer vision (Peters, 2017; Szeliski, 2022), pattern recognition (Braß, 2002; Toussaint, 1982), robotics (Whitesides, 1985), geographic information systems (GIS) (De Floriani et al., 2000; Panigrahi, 2014), computer-aided design (CAD) (Farin et al., 2002), and medical image analysis (Antiga et al., 2003; Ebalunode et al., 2008; Pu et al., 2008). At its core, this discipline focuses on designing algorithms capable of solving geometric problems using mathematical and computational tools. Within this domain, geometric optimisation has become an especially active area of research. Its goal is to identify the best possible solution among all feasible options within a given search space (Augustine et al., 2008; Bae & Yoon, 2023; Naamad et al., 1984).

In parallel with the expansion of this field, the concept of computational thinking (CT) has gained recognition as a fundamental competence in mathematics and science education for the 21st century. Initially introduced by Wing (2006) as a framework for formulating problems and designing solutions through algorithmic thinking, CT goes well beyond programming: it encompasses decomposing complex problems, identifying recurring patterns, and constructing systematic, logical procedures. These skills resonate deeply with computational geometry, a field that requires not only precise understanding of geometric structures but also the capacity to convert spatial ideas into rigorous algorithmic representations.

To avoid treating CT as a broad label, we align our indicators with established frameworks that define CT components for school settings and mathematics/science classrooms (Brennan & Resnick, 2012; Grover & Pea, 2013; Weintrop et al., 2016). In this study, CT-related claims are made only when at least one observable indicator is present in students' artefacts, utterances, or software actions, as specified later in the *Data Collection* and *Analysis* section and further discussed in the *Computational Thinking in Action* section.

In this sense, integrating computational geometry problems into classroom instruction promotes not only deeper engagement with mathematical content related to space, measurement, and optimisation, but also cultivates a structured way of thinking consistent with the central aims of CT (Shute et al., 2017). Recent research further emphasises that incorporating such tasks can support the development of transversal competencies among both lower and upper-secondary students, by bridging logical reasoning with real-world or simulated decision-making scenarios (Lee et al., 2020; Ye et al., 2023).

This convergence between CT and geometry becomes evident when addressing problems that require both mathematical analysis and algorithm design. Tasks such as finding a polygon of maximum area that encloses a given set of points, or minimizing the perimeter of a fence, introduce spatial concepts and optimisation strategies simultaneously. Solving them involves representing geometric data efficiently, formulating procedures, and evaluating possible solutions. Computational geometry is not standard curricular content, but it provides a suitable context for developing CT in the classroom. By engaging with such problems, students are required to use precise representations, apply spatial constraints, and design algorithmic solutions. This discipline enables connections between mathematical concepts and computational strategies, fostering an integrated understanding of both domains (Chytas et al., 2024; Maharani et al., 2021).

From a pedagogical standpoint, this synergy between geometry and CT opens up valuable opportunities for learning. Solving geometric-computational problems provides a bridge between traditional curricular content and the forms of reasoning that are fundamental to computer science and algorithmic logic. These tasks demand not only the identification of key geometric properties, but also the planning of coherent, step-by-step strategies for achieving effective solutions. For instance, when designing an algorithm to compute a simple or convex polygon of maximum area under given spatial constraints (Boyce et al., 1982; Jin, 2015; Lee et al., 2021), students are required to interpret visual representations, generate and test ideas, identify relevant patterns, and encode them as algorithmic instructions. In doing so, they activate both mathematical skills and CT-related competencies, particularly the design of procedural steps and the comparison of alternative solutions.

From a mathematics education perspective, progress in geometry involves coordinating representations, articulating properties and constraints, and moving from visual trial to justification, often supported by purposeful use of digital tools (Jones, 2002; Sinclair et al., 2016; Sinclair & Bruce, 2015). This lens motivates our focus on how students formulate, test, and revise constraints across the paper-and-pencil and software phases, and it clarifies why short utterances are interpreted together with artefacts and software outputs rather than in isolation.

In response to this context, BichromaticSolver (Molano, 2025) was developed, a tool that computes simple or convex k-gons separating two-point sets within a lattice polygon (Molano et al., 2022). Users can select k, the polygon type, and the optimisation criterion (area or perimeter, maximum or minimum). A fuller description of the interface and workflow is provided in the *BichromaticSolver: General Functionality* section. Recent research on technology integration in mathematics classrooms also suggests that the educational value of digital tools depends on structured classroom implementation and task design, rather than on the tool alone (Alkouri & Wardat, 2025).

To explore this potential in a secondary classroom, we implemented three contextualised tasks with students aged 13–14. Each task followed a two-phase sequence: students first worked in paper-and-pencil mode (PPB) and then used BichromaticSolver to verify and refine their solutions. The implementation and data sources are detailed in the *Methodology* section.

One particularly relevant class of problems within this context is the separability of point sets, which forms the conceptual foundation of numerous applications and constitutes the central focus of the tasks explored in this study.

This classroom study is descriptive and exploratory. Its aim is to characterise how students engage with bichromatic separability tasks when they work first in paper-and-pencil mode and then verify and refine their solutions using BichromaticSolver. In particular, we examine the solution strategies students generate, the difficulties they encounter, and the CT-related practices that become visible when constraints and optimisation criteria are made explicit.

RQ1. What solution strategies and difficulties do students exhibit during the paper-and-pencil phase and during software-based verification and refinement?

RQ2. What CT-related practices are evidenced across tasks, and how do these practices differ between the paper-and-pencil and software phases?

RQ3. What forms of teacher support and whole-class validation are salient when students interpret and use software feedback as part of their reasoning?

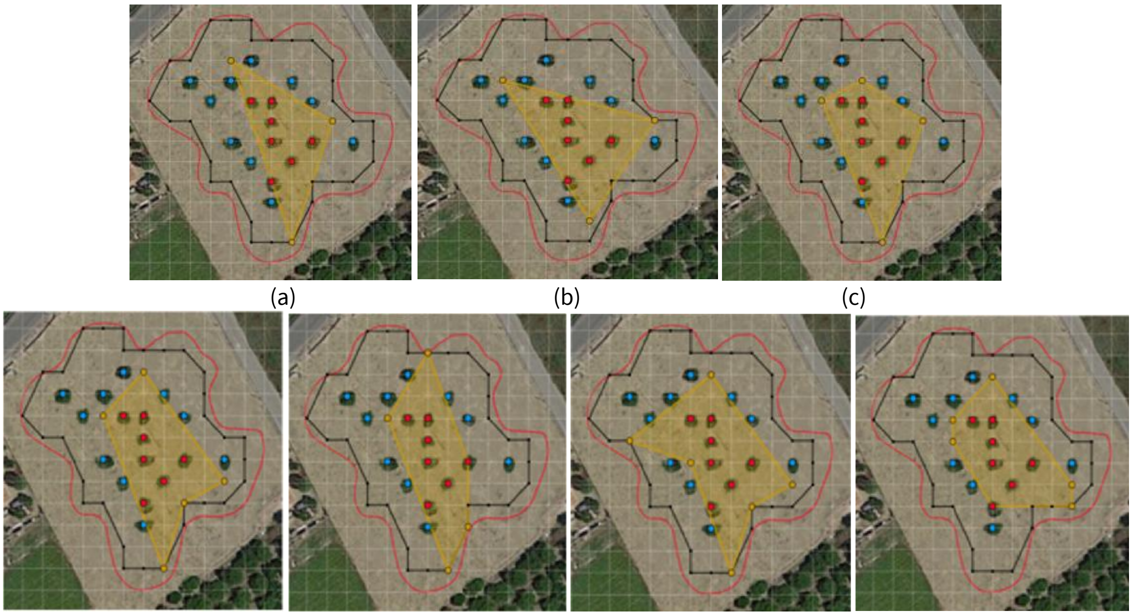
Practical Applications

Computational geometry can play a valuable role in agricultural applications, particularly in the planning and design of greenhouses. These enclosed structures enable the creation of an artificial microclimate conducive to the year-round cultivation of selected plant species. Among their primary benefits are improved pest control and enhanced crop productivity. One critical design challenge involves determining which portion of a given plot should be covered by the greenhouse structure, as not all trees within the area are suitable for inclusion under controlled conditions.

In the proposed classroom task, students begin with an aerial image of a cultivated plot. From this image, a closed contour C is defined (**Figure 1a**), and the trees are classified based on their suitability for greenhouse cultivation: type A trees (suitable) are represented as red points, while type B trees (unsuitable) are marked in blue. A lattice polygon P is then constructed to enclose the area of interest (**Figure 1b**). The main objective is to compute a polygon, of any number of sides, that maximizes the enclosed area while containing all red points and excluding all blue ones. **Figure 2** illustrates several polygonal solutions obtained using BichromaticSolver, clearly reflecting the potential dimensions of the selected greenhouse. Furthermore, **Table 1** summarizes the results for various k-gons, both simple and convex, indicating the number of solutions and the corresponding area in each case.

(a) (b)

Figure 1. Image extraction process for the practical greenhouse design task (a) Original aerial image and extracted closed contour C; (b) Lattice polygon P with seventeen points on closed contour C (Source: Aerial orthophotography from SIGPAC (visorSigPac v4.8, https://sigpac.mapa.gob.es/fega/visor), Fondo Español de Garantía Agraria, Ministerio de Agricultura, Pesca y Alimentación, Gobierno de España, licensed under CC BY 4.0; annotations by the authors)



(a) (b) (c)

(d) (e) (f) (g)

Figure 2. Solutions for the practical task: maximum-area simple and convex *k*-gons; (a) Triangle; (b) Simple Quadrilateral; (c) Convex Quadrilateral; (d) Simple Pentagon; (e) Convex Pentagon; (f) Simple Hexagon; (g) Convex Hexagon (Source: Aerial orthophotography from SIGPAC (visorSigPac v4.8, https://sigpac.mapa.gob.es/fega/visor), Fondo Español de Garantía Agraria, Ministerio de Agricultura, Pesca y Alimentación, Gobierno de España, licensed under CC BY 4.0; annotations by the authors)

Table 1. Solutions: maximum-area simple and convex *k*-gon

***k*-gon**	**Number of solutions**	**Area**
Triangle	1	18
Rectangle	0	—
Simple Quadrilateral	2	20
Convex Quadrilateral	3	19.5
Simple Pentagon	2	23
Convex Pentagon	2	20.5
Simple Hexagon	1	26
Convex Hexagon	1	21

In this task, students engage with authentic geographic data, make informed design decisions, construct visual representations of their proposals, and evaluate their effectiveness using the software.

RELATED WORK ON BICHROMATIC SEPARABILITY PROBLEMS

The concept of separability has long been a central topic in computational geometry, especially when the goal is to distinguish between two distinct categories of spatial data. This section provides an overview of the key ideas related to separability, with a particular focus on its polygonal formulation, which constitutes the core of the present study.

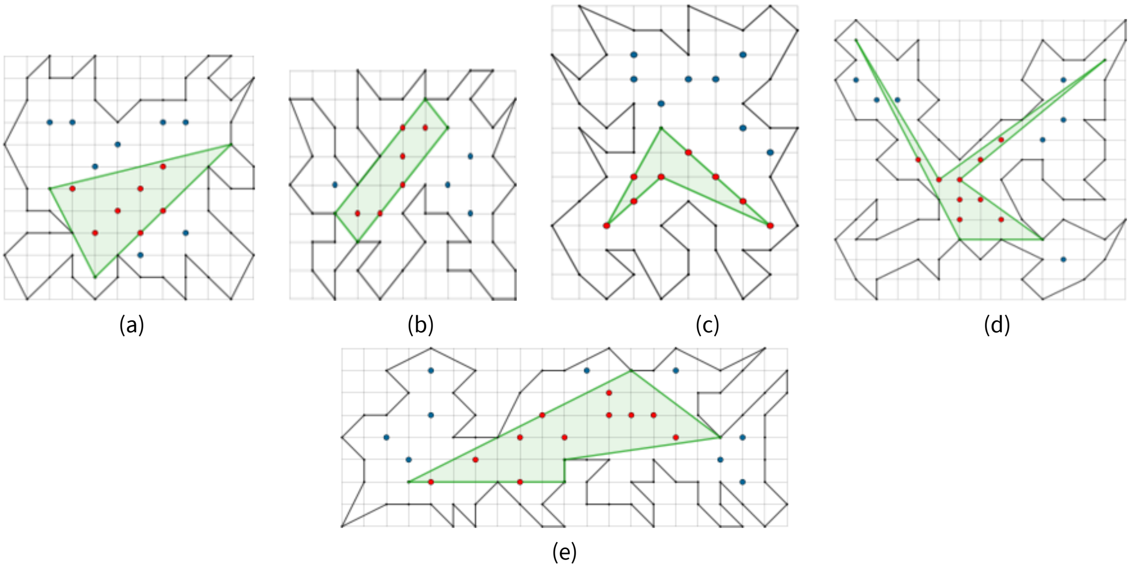


Figure 3. Polygonal configurations computed using different values of k; (a) Triangle maximum area; (b) Rectangle maximum area; (c) Simple quadrilateral minimum area; (d) Simple hexagon maximum perimeter; (e) Simple pentagon maximum area (Source: Authors' own elaboration, using GeoGebra)

Among its many variations, point set separability has received considerable attention due to its versatility and strong relevance to practical applications. It naturally arises in classification tasks in machine learning, robotic motion planning, and even the analysis of biological structures using medical or computational imaging techniques (Dobkin et al., 1996; Duda & Hart, 2000; Eckstein et al., 2002).

In the academic literature, such problems are typically referred to as bichromatic separability problems. The term describes scenarios in which two disjoint sets, often represented as red and blue points, must be separated using a geometric structure. The central challenge is to determine whether such a separation is possible, and under what spatial or geometric conditions.

In our context, we consider two finite point sets in R^2: a red set R and a blue set B, each representing a distinct category. The problem consists in determining whether there exists a simple geometric structure, such as a line (Aronov et al., 2012; Hurtado et al., 2004), a circle (Armaselu & Daescu, 2019; Bitner et al., 2010), or a polygon (Acharyya et al., 2020; Bandyapadhyay & Banik, 2017; Edelsbrunner & Preparata, 1988; Sheikhi & Alipour, 2018), capable of separating all red points from blue ones without overlap or intersection.

Whereas classical formulations of this problem emphasise the mere existence of any separating shape, our approach takes a more specific direction: we examine polygonal separability under optimisation constraints. More precisely, we investigate whether it is possible to construct a simple or convex polygon that encloses all red points while excluding the blue ones (or vice versa) and does so in a way that satisfies a predefined optimality criterion, such as maximizing or minimizing its area or perimeter.

A particularly relevant feature of the approach presented here is that the number of sides k of the polygon is not fixed in advance. Instead, it is defined by the user, which opens the door to a more flexible and exploratory use of the tool, both in didactic and experimental terms. An illustrative set of configurations computed under different values of k and optimisation criteria is presented later when describing the software workflow (the *BichromaticSolver: General Functionality* section, **Figure 3**).

BichromaticSolver: GENERAL FUNCTIONALITY

Functional Overview of the Software

BichromaticSolver is an interactive software tool developed to explore bichromatic separability problems in the plane. The complete software package is openly accessible via the Zenodo repository (Molano, 2025). It computes simple or convex polygons that separate two finite sets of points, displayed in red and blue. The first step in using the program is to define a lattice polygon P (Molano et al., 2022), that is, a polygon whose vertices lie on integer coordinates and is defined over a regular partition Π. We denote by $GL = \{(x_i, y_j): 0 \le i \le r, 0 \le j \le s\}$ the square grid composed of points of the partition Π. We define the partition size, $L = |x_{i+1} - x_i| = |y_{j+1} - y_j|$, the length of the side of each square formed by the square grid. This initial polygon determines the working domain: the point sets to be separated are placed inside it, and the search algorithms operate within this bounded region. **Figure 4** shows the lattice polygon P constructed from the closed contour C introduced in the *Practical Applications* section.

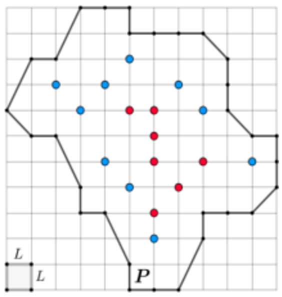


Figure 4. Initial stage of problem setup in *BichromaticSolver* (Source: Authors' own elaboration; output from BichromaticSolver (Molano, 2025))

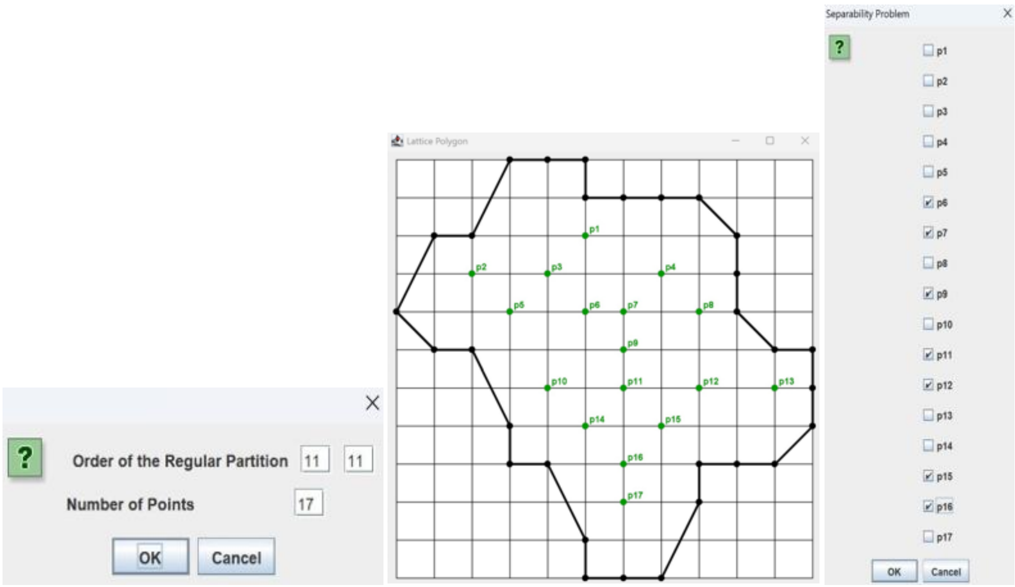


Figure 5. Initial configuration in BichromaticSolver (a) Initial required data; (b) Lattice polygon P; (c) Classification of red and blue points (Source: Authors' own elaboration; output from BichromaticSolver (Molano, 2025))

Based on the user-defined input, the software explores the space of feasible solutions and generates one or more separating polygons that meet the specified conditions. The configurable parameters include the number of sides k, the polygon type (simple or convex), and the geometric quantity to optimise, either area or perimeter, in its maximal or minimal form. The interface enables users to adjust these parameters intuitively and to visualize the resulting configurations in real time, thus supporting both exploratory experimentation and comparative analysis of different strategies.

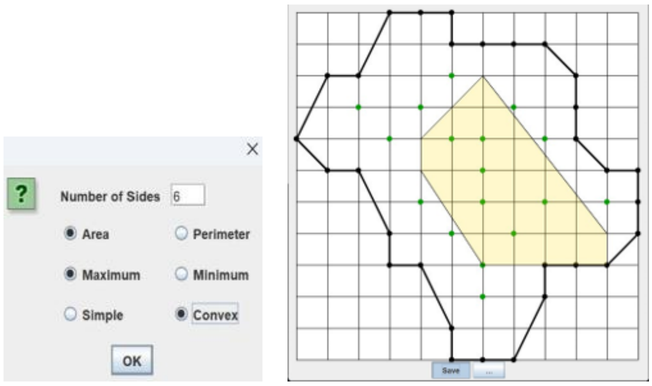
In this study, the software's interactive feedback was used mainly for verification and refinement, while CT-related interpretations are reported later in the *Computational Thinking in Action* section.

Example of Use with Graphical Interface

Continuing with the example introduced earlier, we now present the full process for solving a bichromatic separability problem using the graphical interface of BichromaticSolver. Within this environment, the user configures the problem step by step and obtains a solution that satisfies the defined constraints.

The procedure begins by selecting the order of the regular partition and the number of points to be generated within the lattice polygon (**Figure 5a**). In the illustrated case, an 11 × 11 order is used, and 17 integer-coordinate points are distributed inside it. Once this initial setup is complete, the user draws a closed contour to define the lattice polygon and places the desired points within it (**Figure 5b**). The user then manually selects the subset of points to be classified as red. This selection is performed via a dropdown menu, enabling accurate classification of the points to be separated (**Figure 5c**).

In the next step, the user specifies the criteria for computing the separating polygon: the number of sides k, the geometric measure to optimise (area or perimeter), the optimisation type (maximum or minimum), and the polygon shape (simple or convex). These settings are configured through an intuitive and user-friendly interface that allows for flexible combinations (**Figure 6a**). Once all parameters are set, the algorithm executes and returns the computed solution. In the example shown, a convex hexagon is generated that maximizes the area while enclosing the red points and excluding all others (**Figure 6b**).



(a) (b)

Figure 6. Solution process in BichromaticSolver; (a) Initial configuration options; (b) Convex hexagon maximum area (Source: Authors' own elaboration; output from BichromaticSolver (Molano, 2025))

METHODOLOGY

This study is a classroom-based, descriptive and exploratory intervention that documents how students work on bichromatic separability tasks across a designed two-phase sequence (PPB first, then software). Our focus is on process-level evidence, that is, the strategies students generate, revise, and justify while interacting with the task constraints and the software feedback.

BichromaticSolver was developed as a research tool in computational geometry. The pedagogical application reported here was designed independently of its algorithmic development, and the classroom analysis was carried out with reference to established CT frameworks (Weintrop et al., 2016).

A descriptive and exploratory design is appropriate for RQ1–RQ3 because these questions call for characterising observable patterns of student activity rather than testing theoretical predictions or generating transferable design principles. RQ1 and RQ2 require documenting what strategies emerge, what difficulties arise, and which CT-related practices become visible across phases, aims that are addressed through systematic, episode-based analysis of students' artefacts, utterances, and software interactions. RQ3 requires attending to classroom interaction and teacher moves as they occurred within the designed sequence, for which field notes and whole-class exchanges supply the relevant data. The descriptive stance thus reflects a deliberate alignment between the study's aim (to characterise how students engage with a novel instructional arrangement in its first classroom implementation) and the methods used to examine it.

Classroom Design and Participants

This study adopts a descriptive and exploratory classroom design. Before the intervention, students had already explored essential geometric ideas, such as plotting points on the Cartesian plane, identifying and classifying polygons, and calculating areas and perimeters of basic shapes like triangles and quadrilaterals. This foundational knowledge served as a springboard for tackling the proposed tasks with confidence.

The intervention involved twenty students aged between 13 and 14, all of whom participated voluntarily with informed consent from their legal guardians. The activities were conducted in a standard classroom equipped with individual personal computers, one per student, and an Interactive Digital Whiteboard (IDW), which provided an appropriate setting for implementing the project (De Vita et al., 2014; Holmes, 2009). Before the sessions began, the BichromaticSolver software was installed on all devices to ensure proper functionality and full availability throughout the study.

Classroom observations were documented through field notes taken by the classroom teacher during each session, focusing on students' verbal exchanges, written work, and interactions with the software. Students' paper-and-pencil productions were collected at the end of each session, and screenshots of BichromaticSolver outputs were recorded during the software phase.

The intervention was structured as three 55-minute sessions, each dedicated to a distinct bichromatic separability task. The activities followed a carefully structured progression, enabling students to begin with basic geometric configurations and gradually move toward more complex problem settings. Additional constraints were introduced incrementally, for instance, specifying the number of polygon sides or requiring area/perimeter optimisation. This gradual increase in complexity expanded the mathematical content and promoted the use of increasingly diverse and sophisticated solution strategies.

Each session was structured into two clearly differentiated yet complementary phases. In the first phase, students tackled the problem using traditional paper-and-pencil (PPB) strategies, combining geometric drawing with group discussion. During this stage, Pick's theorem (Papadopoulos & Iatridou, 2010; Pick, 1899) proved particularly useful, as it offered a straightforward and precise method for calculating the area of lattice polygons. Its application allowed students to explore the concept of area through an intuitive counting-based approach, focused on interior and boundary points, without relying on more complex or abstract formulas (Equation 1):

$$A(P) = \left(\#(iP) + \frac{\#(\partial P)}{2} - 1\right)L^2 \quad (1)$$

where A(P) denotes the area of lattice polygon P, # represents the cardinality of a set, ∂P is the set of boundary nodes of P, and its complementary in P, iP, the interior points, so that P = ∂P ∪ iP.

Thus, for the polygon shown in **Figure 6b**, we obtain:

$$A(P) = \left(17 + \frac{10}{2} - 1\right)L^2 = 21L^2$$

To calculate the perimeter, the Cartesian coordinates of the polygon's vertices were used. The Pythagorean theorem was applied to compute the Euclidean distance between each pair of consecutive vertices. By summing these distances, the total perimeter was obtained. Specifically, for two consecutive vertices with coordinates A = (x_1, y_1) and B = (x_2, y_2), the length of the side connecting them was calculated using Equation 2:

$$d_{AB} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2)$$

In the second phase, students used BichromaticSolver to test and refine the solutions previously developed during the paper-and-pencil stage. This digital phase allowed for a more systematic exploration of the problem, enabling them to modify parameters in a controlled manner and critically evaluate the resulting configurations. Students then replicated their polygonal constructions using the program's graphical interface, obtaining the solutions in a visual and automated fashion. Throughout the sessions, an Interactive Digital Whiteboard (IDW) was used as a support tool. It helped clarify questions, foster participation, and maintain student engagement.

The coordinated use of manual and digital methods, along with the connection between visual reasoning and geometric algorithms, made the activity a particularly rich educational experience. Taken as a whole, the methodology not only supported the exploration of curricular geometry content but also offered a meaningful context for integrating computational thinking (CT) into the mathematics classroom in a practical and applied manner, aligned with current educational priorities.

The PPB phase elicited students' initial constraints, candidate constructions, and justifications under limited feedback, whereas the software phase provided systematic feedback that made feasibility and optimisation explicit. This contrast supports RQ1–RQ3 by allowing us to trace strategy refinement and to identify when CT-related indicators become observable in artefacts, utterances, and software actions.

Geometric Tasks and Classroom Implementation

This subsection presents the three geometry tasks designed for the intervention, organized according to a progression of increasing complexity. Each task was addressed through a combination of traditional paper-and-pencil methods and digital tools, allowing students to compare multiple solution strategies using BichromaticSolver.

Task 1: Lattice polygon

The first task aimed to reinforce students' understanding of lattice polygons and familiarize them with the functionalities of BichromaticSolver. Two different configurations were presented, each consisting of a lattice polygon P and a corresponding solution that separates the red points from the blue ones. Depending on the task instructions, students were asked to calculate either the area or the perimeter. Specifically, Task 1a (**Figure 7a**) focused on computing the perimeter of a given simple quadrilateral (maximize perimeter), while Task 1b (**Figure 7b**) involved finding the area of a given simple pentagon (minimize area). Students first applied traditional paper-and-pencil (PPB) techniques, using Pick's theorem for area and the Pythagorean theorem for perimeter, and then verified their results using the software.

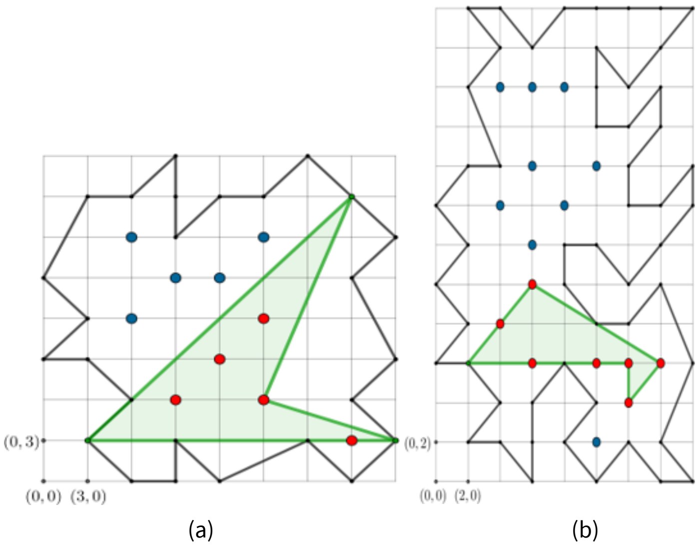


Figure 7. Solution Lattice polygons with different L values; Task 1a) Simple quadrilateral maximum perimeter; (b) Task 1b) Simple pentagon minimum area (Source: Authors' own elaboration, using GeoGebra)

Figure 8. Spatial distribution of animals (Source: Aerial orthophotography from SIGPAC (visorSigPac v4.8, https://sigpac.mapa.gob.es/fega/visor), Fondo Español de Garantía Agraria, Ministerio de Agricultura, Pesca y Alimentación, Gobierno de España, licensed under CC BY 4.0; annotations by the authors)

Task 2: Optimal fencing in an agricultural setting

In livestock management, it is common practice to divide a large estate into smaller enclosures to optimise animal handling. This strategy serves multiple purposes: facilitating pasture rotation, organizing animal movement, increasing productivity, and promoting the conservation of soil and water resources.

Based on this scenario, the proposed task simulates a simplified representation of a livestock estate in which two types of animals coexist. In **Figure 8**, these are represented as red and blue circles, distributed within a bounded region defined by a closed contour C. This spatial distribution reflects a typical layout and serves as the basis for the problem.

The objective of the task is to design a fence that completely encloses one group of animals while excluding the other, using the shortest possible length of fencing. In other words, the challenge is to compute a polygon that separates both point sets while minimizing the total perimeter. To delimit the working area, a lattice polygon P was defined within the contour C, as shown in **Figure 9**. Within this domain, students are expected to explore and compare geometric solutions, first through manual reasoning and then using the BichromaticSolver software.

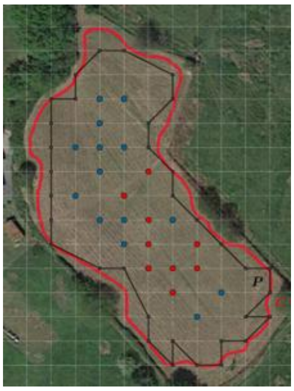


Figure 9. Lattice polygon P defined within the contour C (Source: Aerial orthophotography from SIGPAC (visorSigPac v4.8, https://sigpac.mapa.gob.es/fega/visor), Fondo Español de Garantía Agraria, Ministerio de Agricultura, Pesca y Alimentación, Gobierno de España, licensed under CC BY 4.0; annotations by the authors)

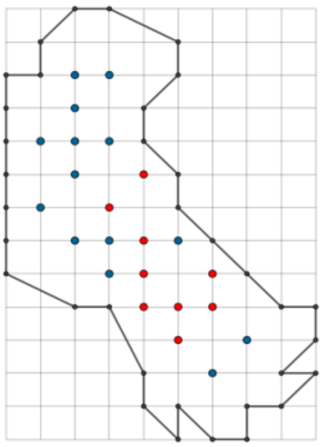


Figure 10. Task 2b. Lattice polygon P (Source: Authors' own elaboration; output from BichromaticSolver (Molano, 2025))

Task 2a. Analyze whether it is possible to enclose all red points using a triangle, a rectangle, or any quadrilateral, without including any blue points. Justify your answer using the BichromaticSolver software.

Task 2b. Compute a simple pentagon with the smallest possible perimeter that successfully separates the red points from the blue ones. Start by proposing a manual solution using the paper-and-pencil (PPB) method. Then, use the BichromaticSolver software to obtain the optimal configuration. Refer to **Figure 10** as a visual reference for both stages of the task. Finally, compare your manual solution with the software-generated result in terms of accuracy, geometric simplicity, and total perimeter.

Task 3: Tumor tissue in medical imaging

Medical imaging offers a particularly compelling application of computational geometry. When analyzing diagnostic images, pathologists examine various visual features, such as color, shape, texture, and density, that may suggest the presence of clinically significant structures.

In many cases, it becomes necessary to distinguish tumor tissue from healthy tissue to plan surgical interventions or administer radiation therapy. In such images, specialists may annotate tumor cells using one color (e.g., red) and healthy cells using another (e.g., blue). Isolating the tumor region based on these annotations gives rise to geometric optimisation problems, for example, computing the triangle of minimal area that encloses all red points, or the polygon of maximal perimeter that separates red and blue points. These geometric challenges lend themselves to digital tools such as BichromaticSolver, while also deepening students' understanding of fundamental geometric concepts and fostering computational thinking (CT) through real-world, healthcare-related scenarios.

Figure 11a depicts a cellular tissue sample in which tumor cells (red points) and healthy cells (blue points) are clearly distinguishable. This spatial arrangement presents a clinically relevant problem: the need to isolate the tumor region in preparation for surgery or radiation treatment. From a geometric perspective, this task can be formalized as a bichromatic separability problem under optimisation constraints. Specifically, the goal is to compute a pentagon of minimal area, either simple or convex, that encloses only the red points while excluding all blue ones (**Figure 11b**).

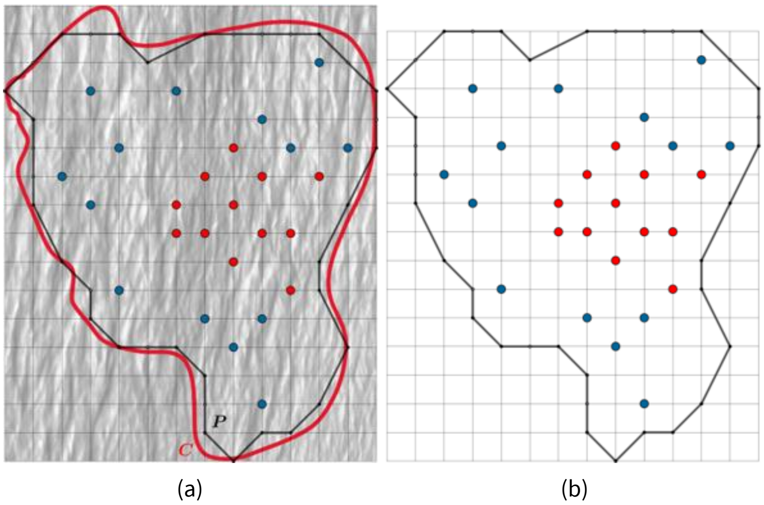


Figure 11. Task 3. Tumor tissue; (a) Tumor (red) and healthy (blue) cells represented within the lattice polygon P; (b) Lattice polygon P (Source: Authors' own elaboration; (a) created with GeoGebra, (b) output from BichromaticSolver (Molano, 2025))

Data Collection and Analysis

Data sources and collection

Data were gathered across three sessions from (i) students' paper-and-pencil work (sketches, lattice counts, and calculations), (ii) artefacts produced in BichromaticSolver (selected parameters and resulting polygons), and (iii) classroom observations. Observation notes focused on key episodes: extracting constraints, proposing candidate polygons, testing feasibility, comparing outcomes under the optimisation criteria, and revising solutions after feedback from peers, the teacher, or the software.

Analysis

The analysis was qualitative and iterative. First, we segmented classroom activity into short episodes aligned with the working cycle implied by the tasks (constraints, conjectures, feasibility testing, optimisation, and comparison). Next, we produced summaries by task and phase to identify recurring strategies, difficulties, and teacher interventions. Finally, we examined CT practices using a set of observable indicators, triangulating students' written work, software outputs, and the associated classroom interactions.

These indicators were operationalised using the CT framework for mathematics and science classrooms proposed by Weintrop et al. (2016). Episodes were tagged only when at least one explicit indicator was present in students' utterances, written artefacts, or software-based actions. For instance, we coded decomposition when students split the task into subgoals (e.g., feasibility first, then optimisation), abstraction when they stated constraints and filtered relevant information (e.g., "no blue points inside"), evaluation when they compared candidate polygons under the optimisation criterion (area/perimeter), and debugging when they revised a construction after detecting a mismatch between a manual attempt and the software output. This procedure aligns the interpretation of CT with concrete classroom evidence rather than with general impressions.

Analytical framework and excerpt selection

Geometric reasoning was analysed by attending to how students articulated and revised geometric constraints (e.g., inclusion/exclusion of points), justified feasibility, and coordinated representations (sketches, lattice counts, coordinates), drawing on established perspectives in geometry education (Battista, 2007; Duval, 2006; van Hiele, 1986).

Quoted utterances in the *Results* section were selected as representative of recurrent episode types and are interpreted together with the surrounding episode description and the corresponding artefacts and software outputs. This was done to avoid treating fragmentary statements as evidence on their own.

RESULTS

The implementation of the three proposed tasks provided valuable insight into how students engage with geometric-computational problems when alternating between traditional and digital learning environments. Each activity was designed to activate prior knowledge and stimulate collaborative reasoning. The combined use of paper-and-pencil (PPB) methods and the BichromaticSolver software allowed for a dual approach: manual strategies fostered conceptual understanding and visual reasoning, while digital tools enabled the verification, refinement, and extension of the initial solutions.

To structure the reporting, the results are organised with reference to RQ1–RQ3, focusing on (i) strategies and difficulties across phases, (ii) CT-related practices evidenced through observable indicators, later synthesised in the *Computational Thinking in Action* section, and (iii) forms of teacher support and whole-class validation.

Task 1: Lattice Polygon

This initial activity was designed to introduce the concepts of area and perimeter in the context of lattice polygons. During the manual phase, most students applied Pick's theorem correctly, although some confusion emerged when counting the number of boundary and interior points.

Determining the perimeter proved more challenging than calculating the area. While some students attempted to estimate side lengths visually, others correctly applied the Pythagorean theorem using vertex coordinates. This diversity of strategies supported a deeper exploration of the concept of distance and revealed differing levels of familiarity with basic algebraic and analytic-geometry tools.

Viewed through a van Hiele lens, Pick's theorem is compatible with Level 2 (Analysis) reasoning, since students identify a countable property of the polygon without yet reasoning about relationships between constraints. Computing perimeter via the distance formula, which requires coordinating vertex coordinates and cumulative summation, begins to demand Level 3 (Ordering) thinking. The coexistence of both approaches within the same task group made the activity accessible across a range of geometric reasoning levels (van Hiele, 1986). It should be noted that Level 4 (Formal Deduction) was not evidenced in this task, as the activity did not include proof-construction components.

During the digital phase, the use of BichromaticSolver helped students validate their initial estimations and receive immediate feedback. Many were surprised to discover discrepancies between manual and digital results, prompting additional verification and adjustments. This interplay between methods prompted students to verify, adjust, and re-check their initial solutions; in our analysis, these observed sequences were tagged as CT-related practices such as decomposition (separating feasibility checks from optimisation) and evaluation, understood as the systematic comparison of outcomes.

A participatory classroom atmosphere was evident throughout the activity, accompanied by spontaneous use of collaborative strategies during both the PPB and digital phases. Peer interaction facilitated the exchange of ideas and the mutual validation of results. **Figures 12** and **13** illustrate some of the solutions developed during the sessions. These figures are included as concrete examples of students' PPB productions, showing how perimeter and area were computed and where counting or measurement errors tended to occur, which informed the teacher prompts reported in **Table 2**.

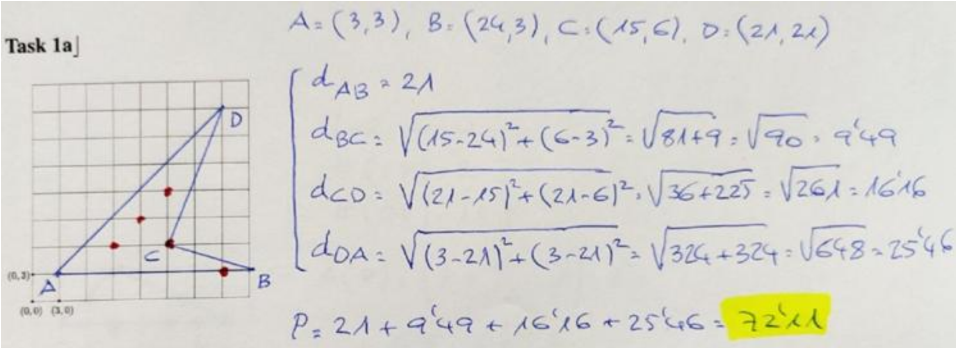


Figure 12. Task 1a. Simple quadrilateral, maximum perimeter (PPB method) (Source: Student's paper-and-pencil work produced during the study, reproduced with the participants' informed consent (anonymised))

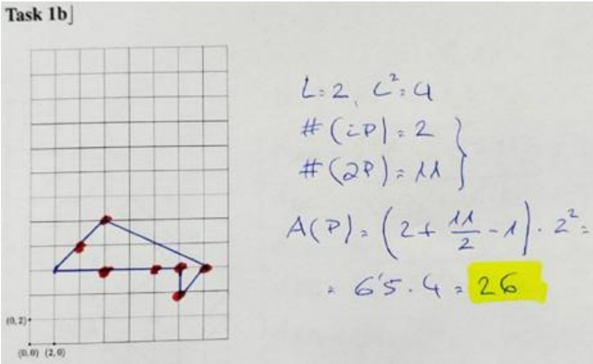


Figure 13. Task 1b. Simple pentagon, minimum area (PPB method) (Source: Student's paper-and-pencil work produced during the study, reproduced with the participants' informed consent (anonymised))

Table 2. Summary of strategies and support observed during the implementation of the three tasks

Task		Strategy observed (SO), Difficulties (D), Teacher interventions (TI)
Task 1	**SO**	· Perimeter estimation by counting unit squares · Direct application of the Pythagorean theorem · Direct application of Pick's theorem
	D	· Confusion between the length of sides and diagonals · Difficulty applying the distance formula · Misclassification of boundary points
	TI	· Use of Cartesian coordinates and triangles to measure distances · Clarification on the actual length of diagonals · Review of Pick's theorem using visual examples
Task 2	**SO**	· Trial with simple shapes (triangles, rectangles) · Avoiding blue points through trial and error · Manual comparison of different solutions
	D	· Unintentional inclusion of blue points · Difficulty recognising when no solution exists · Construction of valid but non-optimal polygons
	TI	· Discussion of the meaning of "no possible solution" · Guidance to identify key points that determine feasibility · Use of the software to calculate perimeter
Task 3	**SO**	· Manual construction of a pentagon enclosing all red points · Manual adjustment of vertices to reduce the covered area
	D	· Use of polygons with more than five sides · Errors when trying to include all points with the minimum possible area
	TI	· Use of the software to verify the problem solution · Guided discussion on how to include all red points while minimising the area

Task 2: Optimal Fencing in an Agricultural Setting

The second task addressed a contextualized problem in livestock management, requiring students to separate two types of animals represented by red and blue points. The objective was to design a fence that enclosed only the red-point group while minimizing the total perimeter. The activity was divided into two complementary subtasks.

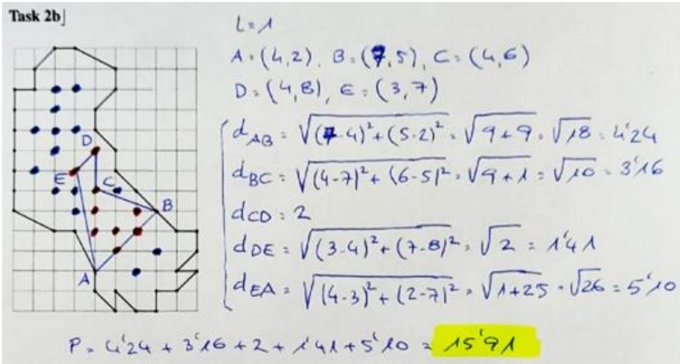


Figure 14. Task 2b. Simple pentagon, minimum perimeter (PPB method) (Source: Student's paper-and-pencil work produced during the study, reproduced with the participants' informed consent (anonymised))

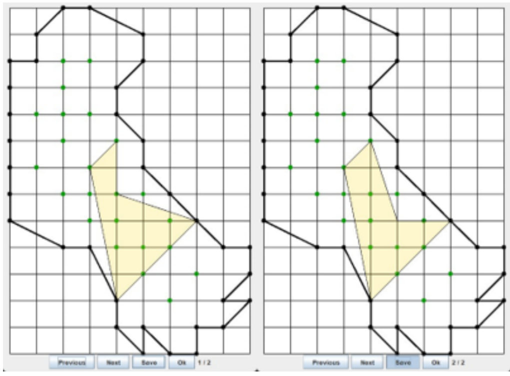


Figure 15. Task 2b. Simple pentagon, minimum perimeter (BichromaticSolver) (Source: Authors' own elaboration; output from BichromaticSolver (Molano, 2025))

Task 2a asked whether it was possible to enclose all the red points using basic geometric figures such as triangles, rectangles, or quadrilaterals, without including any blue points. During the manual phase, students explored different strategies on paper and then verified their hypotheses using the BichromaticSolver software.

The following utterances were selected because they recurred across multiple groups during the PPB phase of Task 2a and capture the progression from trial-and-error to explicit constraint articulation. Each is interpreted together with the associated paper sketches and the BichromaticSolver output for that group.

"I drew several triangles, but there was always a blue point inside."

"I thought a rectangle might work, but it always ended up including a blue point."

"I tried using a quadrilateral to see if that would work, but one of the sides crossed over a blue point."

"With BichromaticSolver, I saw there was no solution."

"I used to think any four-sided figure would work, but it didn't."

Taken together, these comments suggest that students moved from trying familiar shapes to articulating explicit constraints of separability, for instance identifying that a blue point remained inside a candidate polygon or that a side crossed a blue point. In the digital phase, the software output enabled them to revisit their paper sketches, confirm when no feasible solution existed for a given shape, and revise their initial assumptions accordingly.

Viewed through a van Hiele lens, Task 2a shows a progression from Level 2 (Analysis) to Level 3 (Ordering). Initially, students identified candidate shapes by their known properties and tested them by trial and error, a Level 2 approach. As the session advanced, they shifted towards Level 3 reasoning: rather than testing shapes one by one, they began to articulate why a configuration failed ("a blue point is always inside", "a side crosses a blue point"), relating geometric constraints to each other. BichromaticSolver's confirmation that no feasible solution existed reinforced this transition, prompting students to move from empirical testing to informal justification (van Hiele, 1986). It should be noted that Level 4 (Formal Deduction), which would require students to construct a formal proof of impossibility rather than verifying it empirically through the software, was not evidenced in this study. This is consistent with the exploratory and technology-mediated nature of the intervention.

In Task 2b, students were asked to calculate a simple pentagon that separates the red points from the blue ones while minimizing the perimeter. This subtask required not only identifying a valid solution but also achieving an optimal one. Comparing manual and digital approaches made visible CT-related practices, particularly evaluating efficiency (perimeter comparison), verifying feasibility, and refining a candidate polygon through successive adjustments.

Figures 14 and **15** show some of the configurations analyzed during the task, both on paper and through the digital interface of the software. **Figure 14** is used to document a representative manual attempt, whereas **Figure 15** provides the software-derived optimum used for comparison; together, they support the discussion of evaluation and refinement practices associated with the PPB-to-software transition.

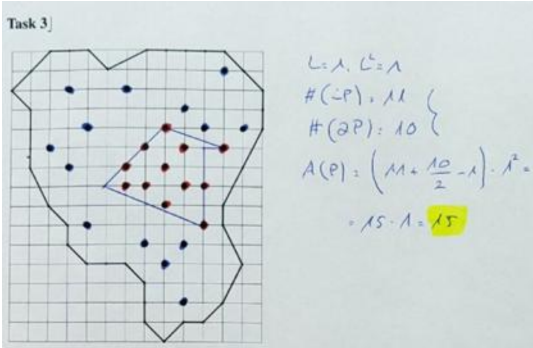


Figure 16. Task 3) Simple pentagon, minimum area (PPB method) (Solution 1) (Source: Student's paper-and-pencil work produced during the study, reproduced with the participants' informed consent (anonymised))

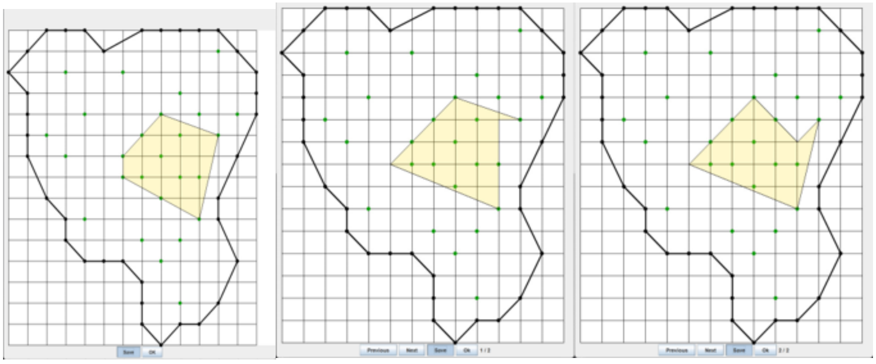


Figure 17. Task 3) Pentagon, minimum area; (a) Simple; (b) Convex (Source: Authors' own elaboration; output from BichromaticSolver (Molano, 2025))

Task 3: Tumor Tissue in Medical Imaging

Task 3 introduced a medical diagnosis context in which students were required to isolate a group of tumor cells (red points) from healthy tissue (blue points) within a bounded lattice area. The objective was to compute a pentagon of minimal area that enclosed only the red points.

During the manual phase, most students were able to exclude the blue points entirely and calculate the area of their solutions with reasonable precision. These challenges prompted discussions about how to position the polygon's sides to avoid including blue points while still minimizing the area.

In the digital phase, BichromaticSolver generated optimal solutions that surprised many students due to the unexpectedly small areas of the resulting polygons. This discrepancy between their expectations and the actual results sparked a discussion about the meaning of optimisation in geometry and the role of digital tools in facilitating the exploration of multiple alternatives more efficiently.

In response to this diversity of strategies, the teacher projected a student's manual pentagon alongside the BichromaticSolver output on the IDW and asked the class: “Both polygons separate the same points, why does the software find a smaller area?” Students identified that their vertices had been placed further from the interior points than necessary and tested this immediately in the software. This pattern of teacher-guided whole-class comparison followed by individual revision recurred across tasks and constituted the main mechanism through which software feedback became a reasoning tool rather than a mere answer checker.

This episode illustrates a transition from Level 2 (Analysis) to Level 3 (Ordering). Students who placed vertices at a safe visual distance were satisfying the inclusion/exclusion constraint property by property, without relating vertex position to the optimisation criterion, a Level 2 approach. The teacher's question prompted them to connect vertex placement explicitly to enclosed area, coordinating two geometric properties simultaneously, characteristic of Level 3 reasoning. Software testing immediately allowed them to verify this relationship (Battista, 2007; van Hiele, 1986). As in the preceding tasks, Level 4 (Formal Deduction) was not evidenced, consistent with the exploratory nature of the intervention and the absence of proof-writing activities.

Task 3 made explicit connections between geometry, a medical imaging context, and CT-related practices, particularly systematic testing, software-based verification, and revision of candidate solutions. **Figures 16** and **17** illustrate some of the constructions created by students during the activity. These figures are discussed to highlight how students positioned sides to exclude blue points while reducing area, and how the software output helped them recognise when a seemingly reasonable polygon was not optimal under the selected criterion.

Table 3. Key CT components and observable indicators across phases of task resolution

CT Component	Phase (PPB / software / both)	
Abstraction	**PPB**	· Focus on relevant points while ignoring obstacles · Use of simplified shapes to represent complex regions · Verbal reasoning to describe geometric constraints
Decomposition	**Software**	· Separate steps: selection, construction, evaluation · Isolation of red and blue point sets · Breakdown of problem into manageable subtasks
Pattern recognition	**PPB**	· Recognition of configurations that block solutions · Generalising from successful polygons in earlier tasks · Transfer of reasoning across different values of *k*
Evaluation	**Both**	· Comparison of polygon areas and perimeters · Use of software feedback to assess solution quality · Identification of suboptimal or invalid outputs
Debugging	**Software**	· Testing alternative values after failed attempts · Modification of point selection to correct errors · Re-evaluation after changing polygon constraints

Summary of Student Strategies and Teacher Support

To bring together the diversity of behaviours observed across the three tasks, **Table 2** presents an organised synthesis. For each activity, it highlights the most common strategies employed by students (SO), the main difficulties that arose (D), and the types of teacher intervention (TI) that responded to those situations.

Although the table does not aim to be exhaustive, it reflects the patterns that recurred throughout the sessions. The interplay between autonomous exploration and teacher guidance emerges as a key pedagogical axis in the development of these tasks.

Observations indicate that, during the paper-and-pencil phase, students often relied on visual intuition to approach the problems. However, as they began interacting with the software interface, they gradually refined both their constructions and their strategies. Teacher support proved particularly effective when it involved posing open-ended questions, encouraging comparisons between solutions, and assisting with the interpretation of feedback provided by the program. This observation aligns with research suggesting that effective digital integration depends on the alignment between technological affordances, mathematical goals, and teacher-guided classroom routines (Aldalalah et al., 2025).

COMPUTATIONAL THINKING IN ACTION

In addition to addressing mathematical content, the three tasks offered meaningful opportunities for the development of computational thinking (CT) skills. **Table 3** summarises how the main CT components emerged during classroom implementation, indicating the phase (paper-and-pencil or software) in which each component was most evident and providing illustrative examples from the sessions.

For clarity, we distinguish between (i) observed behaviours, such as students' written productions, utterances, parameter changes in the software, and explicit comparisons of results, and (ii) interpretive inferences, where these observations are mapped to CT components using the indicators specified in **Table 3**. When CT labels are used below, they refer to this explicit mapping and not to an independent measurement of CT.

For example, when students compared two feasible pentagons by perimeter and then modified vertices to reduce the value, the observed sequence (comparison, adjustment, re-check) was coded as evaluation and debugging in CT terms (**Table 3**).

Across the three sessions, classroom observations showed that CT was not introduced as a separate topic, but became visible through repeated cycles of proposing, testing, and revising polygons under explicit constraints and optimisation criteria. Students engaged in algorithmic reasoning by adjusting parameters, testing hypotheses, and interpreting the software's feedback in relation to their own reasoning processes.

This pattern resonates with, but also extends, recent work on CT integration in secondary mathematics. Chytas et al. (2024) reported CT-related practices in GeoGebra-supported calculus lessons, finding that digital tools prompted students to test and revise their mathematical conjectures. The present study extends this line of inquiry to computational geometry and separability tasks, and differs in that it explicitly traces CT practices across a paper-to-software transition rather than within a single digital environment.

This finding aligns with Wing's (2006) view of CT as a transversal competence that goes beyond programming, and with later studies that highlight its integration into different disciplinary contexts (Brennan & Resnick, 2012; Grover & Pea, 2013; Mumcu et al., 2023; Weintrop et al., 2016). In the case analysed here, the paper-and-pencil phase primarily fostered processes of abstraction and pattern recognition, whereas the use of software brought to the fore dimensions such as evaluation of results and debugging of strategies. This complementarity between analogue and digital methods reinforces the idea that CT can be cultivated within mathematics without the need to rely exclusively on programming environments.

Classroom observations further showed that teacher interventions were crucial in making sense of the activity. Open-ended questions, guided comparisons, and support in interpreting the program's feedback helped students perceive the software as a reasoning tool rather than as a mere solution checker. From a pedagogical perspective, this experience offers teachers a concrete example of how CT can be developed transversally within the geometry curriculum, enriching traditional content without radically altering school programmes.

Looking ahead, it is worth exploring whether this methodology can be transferred to other areas of mathematics, such as algebra or functions, and to examine to what extent repeated exposure to separability problems contributes to consolidating CT skills in the medium and long term. It would also be advisable to design teacher training initiatives that provide resources and confidence to integrate this type of activity into everyday practice.

Episodes were tagged only when the observed behaviour matched at least one positive indicator. For instance, a student drawing a polygon without stating any inclusion/exclusion criterion was not coded as Abstraction; a student accepting the software result without comparing it to a manual solution was not coded as Evaluation.

CONCLUSIONS

This paper has shown that separability problems, usually addressed from a theoretical perspective, can be successfully adapted to secondary education. The combination of paper-and-pencil phases and the use of BichromaticSolver allowed students to verify their results, compare strategies, and refine constructions, thereby supporting the refinement of geometric reasoning and helping students articulate constraints and optimisation goals more explicitly. With respect to the three research questions, the findings show that students progressed from visual trial-and-error to constraint-based reasoning (RQ1); that CT-related practices were more explicitly visible in the software phase than in the paper-and-pencil phase (RQ2); and that teacher-guided whole-class comparison was the key mechanism linking software feedback to student reasoning (RQ3).

The inclusion of real-world contexts, connected to domains such as agriculture or medicine, gave meaning to the activities and helped students perceive the usefulness of geometry in authentic situations. This practical orientation enhanced motivation and demonstrates the potential of such tasks to connect school mathematics with real-world problems. This interpretation is consistent with recent technology-enhanced mathematics education studies linking motivation to classroom designs that emphasise active work, discussion, and feedback (Jarrah et al., 2025).

The findings also indicate that these activities provide a suitable environment for the development of computational thinking (CT). While the manual phase encouraged abstraction and pattern recognition, working with the software highlighted processes of evaluation, comparison, and debugging. The complementarity between analogue and digital methods supports the view that CT may emerge within the mathematics classroom, provided that teacher interventions guide reflection and give purpose to the use of digital tools.

This positioning aligns with mathematics education accounts that treat CT as intertwined with core mathematical practices and classroom activity, rather than as a separate programming add-on (Gadanidis et al., 2017; Weintrop et al., 2016).

Limitations

The study is exploratory and context-specific (one classroom, three sessions) and does not include a pre/post assessment or a comparison group. Therefore, we avoid strong claims about learning gains and interpret the findings as evidence of strategy development and CT-related practices as they became visible through students' artefacts, discussions, and software-based verification.

A first limitation concerns sample size and transferability. The intervention involved a small group (N = 20, aged 13–14) from a single classroom, so the findings should be read as situated evidence of how students engaged with the tasks in this setting, rather than as results that generalise without replication.

A second limitation is the influence of teacher orchestration. Teacher prompts, whole-class validation, and support in interpreting software feedback were integral to the implementation (see **Table 2**), and they may have shaped both the strategies students adopted, and which CT-related practices became publicly visible. Future work should replicate the sequence across settings and include quantitative measures (e.g., repeated assessments) to examine learning trajectories more directly. In addition, it would be valuable to examine how teachers' attitudes towards mathematical modelling shape the enactment and sustainability of this type of classroom sequence across contexts (Wardat et al., 2025).

A third limitation concerns the ceiling of geometric reasoning evidenced. The van Hiele analysis reported in the *Results* section (Task 1–Task 3) suggests that students progressed up to Level 3 (Ordering), but Level 4 (Formal Deduction) was not reached. Future work could examine whether sustained exposure to separability tasks, combined with explicit proof-writing activities, supports progression towards Level 4 in secondary students.

Despite these limitations, the results open avenues for further research and classroom applications. It would be interesting to explore adaptations to other mathematical domains and to examine the longer-term impact of sustained exposure to this paper-and-pencil to software sequence. Overall, the experience suggests that separability is not only a mathematically rich challenge, but also a pedagogical resource with strong potential to connect school geometry, CT, and the demands of contemporary society.

Author contributions: RM: conceptualisation, methodology, formal analysis, software, investigation, writing – original draft; **MH:** software, visualisation; **MMÁ:** methodology, formal analysis, writing – review & editing; **PGR:** validation, writing – review & editing; **AC:** supervision, project administration, writing – review & editing. All authors have agreed with the results and conclusions.

Funding: No funding source is reported for this study.

Ethical statement: The authors stated that the study did not require formal approval by an ethics committee, in accordance with the internal guidelines of the University of Extremadura for educational research that does not involve sensitive data or clinical procedures. Nevertheless, the study was conducted in full compliance with international ethical standards for research involving human participants, including voluntary participation and data confidentiality. The authors further stated that Informed consent was obtained from all participants included in the study, and the entire process of requesting and validating consent was carried out in English. An email was sent to the students' parents or legal guardians outlining the purpose and objectives of the study. The consent form was included as an attachment. They were given two days to read the document carefully and, if they agree, to sign and return it. Only the data corresponding to students whose participation was formally authorized through the signed consent form were included in the study.

AI statement: The author stated that, during the preparation of this work the authors used AI-assisted technologies for language editing and text correction. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Declaration of interest: No conflict of interest is declared by the authors.

Data sharing statement: Data supporting the findings and conclusions are available upon request from the corresponding author.

Code availability: The BichromaticSolver application is publicly available through the Zenodo repository 1, which provides a permanent DOI and versioned archiving. The downloadable package includes runnable binaries (.jar and .exe) as well as several example configurations that demonstrate the core functionalities of the software. The program is compatible with Windows systems and requires Java SE Development Kit version 24.0.1 or higher. The authors assume full responsibility for the development and educational use of the software presented in this study.

REFERENCES

Acharyya, A., De, M., Nandy, S. C., & Pandit, S. (2020). Variations of largest rectangle recognition amidst a bichromatic point set. *Discrete Applied Mathematics, 286*, 35-50. https://doi.org/10.1016/j.dam.2019.05.012

Aldalalah, O. M. A., Wardat, Y., Al-Omari, A. A. H., & Khodair, R. M. (2025). The effectiveness of interactive digital content based on the TPACK model in developing the skills of educational aids production and improving cognitive achievement among early childhood university students. *Contemporary Educational Technology, 17*(2), Article ep572. https://doi.org/10.30935/cedtech/16046

Alkouri, Z., & Wardat, Y. (2025). Digital integration in preschool math: Assessing the efficacy of a structured curriculum in classroom settings. *International Electronic Journal of Mathematics Education, 20*(4), Article em0845. https://doi.org/10.29333/iejme/16662

Antiga, L., Ene-Iordache, B., & Remuzzi, A. (2003). Computational geometry for patient-specific reconstruction and meshing of blood vessels from MR and CT angiography. *IEEE Transactions on Medical Imaging, 22*(5), 674-684. https://doi.org/10.1109/TMI.2003.812261

Armaselu, B., & Daescu, O. (2019). Dynamic minimum bichromatic separating circle. *Theoretical Computer Science, 774*, 133-142. https://doi.org/10.1016/j.tcs.2016.11.036

Aronov, B., Garijo, D., Núñez-Rodríguez, Y., Rappaport, D., Seara, C., & Urrutia, J. (2012). Minimizing the error of linear separators on linearly inseparable data. *Discrete Applied Mathematics, 160*(10-11), 1441-1452. https://doi.org/10.1016/j.dam.2012.03.009

Augustine, J., Putnam, B., & Roy, S. (2008). *Largest empty circle centered on a query line*. arXiv preprint arXiv:0809.2651. https://doi.org/10.48550/arXiv.0809.2651

Bae, S. W., & Yoon, S. D. (2023). Empty squares in arbitrary orientation among points. *Algorithmica, 85*(1), 29-74. https://doi.org/10.1007/s00453-022-01002-1

Bandyapadhyay, S., & Banik, A. (2017). Polynomial time algorithms for bichromatic problems. In *Conference on Algorithms and Discrete Applied Mathematics* (pp. 12-23). https://doi.org/10.1007/978-3-319-53007-9_2

Battista, M. T. (2007). The development of geometric and spatial thinking. In F. K. Lester (Ed.), *Second handbook of research on mathematics teaching and learning* (pp. 843-908). Information Age Publishing.

Bitner, S., Cheung, Y., & Daescu, O. (2010). Minimum separating circle for bichromatic points in the plane. In *2010 International Symposium on Voronoi Diagrams in Science and Engineering* (pp. 50-55). https://doi.org/10.1109/ISVD.2010.14

Boyce, J. E., Dobkin, D. P., Drysdale III, R. L., & Guibas, L. J. (1982). Finding extremal polygons. In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing* (pp. 282-289). https://doi.org/10.1145/800070.802202

Braß, P. (2002). Combinatorial geometry problems in pattern recognition. *Discrete & Computational Geometry, 28*, 495-510. https://doi.org/10.1007/s00454-002-2884-3

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*, Vancouver, Canada.

Chytas, C., van Borkulo, S. P., Drijvers, P., Barendsen, E., & Tolboom, J. L. J. (2024). Computational thinking in secondary mathematics education with GeoGebra: Insights from an intervention in calculus lessons. *Digital Experiences in Mathematics Education, 10*(2), 228-259. https://doi.org/10.1007/s40751-024-00141-0

De Floriani, L., Magillo, P., & Puppo, E. (2000). Applications of computational geometry to geographic information systems. *Handbook of Computational Geometry, 7*, 333-388. https://doi.org/10.1016/B978-044482537-7/50008-5

De Vita, M., Verschaffel, L., & Elen, J. (2014). Interactive whiteboards in mathematics teaching: A literature review. *Education Research International, 2014*(1), 401315. https://doi.org/10.1155/2014/401315

Dobkin, D. P., Gunopulos, D., & Maass, W. (1996). Computing the maximum bichromatic discrepancy, with applications to computer graphics and machine learning. *Journal of Computer and System Sciences, 52*(3), 453-470. https://doi.org/10.1006/jcss.1996.0034

Duda, R. O., & Hart, P. E. (2000). *Pattern classification*. John Wiley & Sons.

Duval, R. (2006). A cognitive analysis of problems of comprehension in a learning of mathematics. *Educational Studies in Mathematics, 61*, 103-131. https://doi.org/10.1007/s10649-006-0400-z

Ebalunode, J. O., Ouyang, Z., Liang, J., Zheng, W. (2008). Novel approach to structure-based pharmacophore search using computational geometry and shape matching techniques. *Journal of Chemical Information and Modeling, 48*(4), 889-901. https://doi.org/10.1021/ci700368p

Eckstein, J., Hammer, P. L., Liu, Y., Nediak, M., & Simeone, B. (2002). The maximum box problem and its application to data analysis. *Computational Optimization and Applications, 23*(3), 285-298. https://doi.org/10.1023/A:1020546910706

Edelsbrunner, H., & Preparata, F. P. (1988). Minimum polygonal separation. *Information and Computation, 77*(3), 218-232. https://doi.org/10.1016/0890-5401(88)90049-1

Farin, G., Hoschek, J., & Kim, M. S. (Eds.). (2002). *Handbook of computer aided geometric design*. North Holland. https://doi.org/10.1016/B978-0-444-51104-1.X5000-X

Gadanidis, G., Cendros, R., Floyd, L., & Namukasa, I. (2017). Computational thinking in mathematics teacher education. *Contemporary Issues in Technology and Teacher Education, 17*(4), 458-477.

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher, 42*(1), 38-43. https://doi.org/10.3102/0013189X12463051

Holmes, K. (2009). Planning to teach with digital tools: Introducing the interactive whiteboard to pre-service secondary mathematics teachers. *Australasian Journal of Educational Technology, 25*(3), 351-365. https://doi.org/10.14742/ajet.1139

Hurtado, F., Mora, M., Ramos, P. A., & Seara, C. (2004). Separability by two lines and by nearly straight polygonal chains. *Discrete Applied Mathematics, 144*(1-2), 110-122. https://doi.org/10.1016/j.dam.2003.11.014

Jarrah, A. M., Wardat, Y., Fidalgo, P., & Ali, N. (2025). Gamifying mathematics education through Kahoot: Fostering motivation and achievement in the classroom. *Research and Practice in Technology Enhanced Learning, 20*, Article 10. https://doi.org/10.58459/rptel.2025.20010

Jin, K. (2015). *Maximal parallelograms in convex polygons*. arXiv preprint arXiv:1512.03897. https://doi.org/10.48550/arXiv.1512.03897

Jones, K. (2002). Issues in the teaching and learning of geometry. In *Aspects of teaching secondary mathematics* (pp. 137-155). Routledge.

Lee, I., Grover, S., Martin, F., Pillai, S., & Malyn-Smith, J. (2020). Computational thinking from a disciplinary perspective: Integrating computational thinking in K-12 science, technology, engineering, and mathematics education. *Journal of Science Education and Technology, 29*, 1-8. https://doi.org/10.1007/s10956-019-09803-w

Lee, S., Eom, T., & Ahn, H. K. (2021). Largest triangles in a polygon. *Computational Geometry, 98*, Article 101792. https://doi.org/10.1016/j.comgeo.2021.101792

Maharani, S., Agustina, Z. F., & Kholid, M. N. (2021). Exploring the prospective mathematics teachers computational thinking in solving pattern geometry problem. *Al-Ishlah: Jurnal Pendidikan, 13*(3), 1756-1767. https://doi.org/10.35445/alishlah.v13i3.1181

Molano, R. (2025). *BichromaticSolver: A classroom resource for computational geometry*. Zenodo. https://doi.org/10.5281/zenodo.16754364

Molano, R., Avila, M., Sancho, J. C., Rodríguez, P. G., & Caro, A. (2022). An algorithm to compute any simple k-gon of a maximum area or perimeter inscribed in a region of interest. *SIAM Journal on Imaging Sciences, 15*(4), 1808-1832. https://doi.org/10.1137/22M1482676

Mumcu, F., Kidiman, E., & Özdinç, F. (2023). Integrating computational thinking into mathematics education through an unplugged computer science activity. *Journal of Pedagogical Research, 7*(2), 72-92. https://doi.org/10.33902/JPR.202318528

Naamad, A., Lee, D., & Hsu, W. L. (1984). On the maximum empty rectangle problem. *Discrete Applied Mathematics, 8*(3), 267-277. https://doi.org/10.1016/0166-218X(84)90124-0

Panigrahi, N. (2014). *Computing in geographic information systems*. https://doi.org/10.1201/b17147

Papadopoulos, I., & Iatridou, M. (2010). Systematic approaches to experimentation: The case of Pick's theorem. *The Journal of Mathematical Behavior, 29*(4), 207-217. https://doi.org/10.1016/j.jmathb.2010.10.002

Peters, J. F. (2017). *Foundations of computer vision: computational geometry, visual image structures and object shape detection*. https://doi.org/10.1007/978-3-319-52483-2

Pick, G. (1899). Geometrisches zur zahlenlehre [Geometric principles for number theory]. *Sitzenber Lotos (Prague), 19*, 311-319.

Pu, J., Leader, J. K., Zheng, B., Knollman, F., Fuhrman, C., Sciurba, F. C., & Gur, D. (2008). A computational geometry approach to automated pulmonary fissure segmentation in CT examinations. *IEEE Transactions on Medical Imaging, 28*(5), 710-719. https://doi.org/10.1109/TMI.2008.2010441

Sheikhi, F., & Alipour, S. (2018). *On triangular separation of bichromatic point sets in polygonal environment*. arXiv preprint arXiv:1809.00116. https://doi.org/10.48550/arXiv.1809.00116

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review, 22*, 142-158. https://doi.org/10.1016/j.edurev.2017.09.003

Sinclair, N., Bartolini Bussi, M. G., de Villiers, M., Jones, K., Kortenkamp, U., Leung, A., & Owens, K. (2016). Recent research on geometry education: An ICME-13 survey team report. *ZDM Mathematics Education, 48*(5), 691-719. https://doi.org/10.1007/s11858-016-0796-6

Sinclair, N., & Bruce, C. D. (2015). New opportunities in geometry education at the primary school. *ZDM Mathematics Education, 47*(3), 319-329. https://doi.org/10.1007/s11858-015-0693-4

Szeliski, R. (2022). *Computer vision: algorithms and applications*. https://doi.org/10.1007/978-3-030-34372-9

Toussaint, G. T. (1982). Computational geometric problems in pattern recognition. In *Pattern Recognition Theory and Applications: Proceedings of the NATO Advanced Study Institute held at St. Anne's College, Oxford, March 29–April 10, 1981* (pp. 73-91). https://doi.org/10.1007/978-94-009-7772-3_7

van Hiele, P. M. (1986). *Structure and insight: A theory of mathematics education*. Academic Press.

Wardat, Y., Hidayat, R., Belbase, S., Al-Shraifin, A. A., & Alribdi, N. I. (2025). Cross-National Comparison of Mathematical Modelling Attitudes Among Educators. *European Journal of Education, 60*(4), Article e70288. https://doi.org/10.1111/ejed.70288

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*(1), 127-147. https://doi.org/10.1007/s10956-015-9581-5

Whitesides, S. H. (1985). Computational geometry and motion planning. *Machine Intelligence and Pattern Recognition, 2*, 377-427. https://doi.org/10.1016/B978-0-444-87806-9.50019-0

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35. https://doi.org/10.1145/1118178.1118215

Ye, H., Liang, B., Ng, O. L., & Chai, C. S. (2023). Integration of computational thinking in K-12 mathematics education: A systematic review on CT-based mathematics instruction and student learning. *International Journal of STEM Education, 10*(1), Article 3. https://doi.org/10.1186/s40594-023-00396-w