

# Graph Pattern Matching in Agriculture

## Literature Review

Chelsea Van Collier

Department of Computer Science

University of Cape Town

Cape Town, Western Cape, South  
Africa

vclche001@myuct.ac.za

## ABSTRACT

Food security is a pressing issue that is faced globally. The agricultural sector is forced to keep up with rising demands as the population increases. As a result, farmers need to ensure an improvement in yield as they are met with land constraints. Precision agriculture attempts to assist farmers by taking high resolution images of their crops via drones and analyzing these images to provide various insights into the best management practices. Planting patterns have been shown to effect crop yield and the problem that will be addressed in this paper is how to effectively identify various patterns within the crops. The method used to establish these patterns is graph pattern matching. The crops are converted into a graph by treating each center point of the tree polygon as the vertex/node within the graph and the edges will correspond to the Euclidean distance between the trees. Background is given on the topic as graph pattern matching is also referred to as subgraph isomorphism/ homomorphism. Graph pattern matching is split into two categories – exact and inexact pattern matching. Various algorithms are analyzed in each category. Exact pattern matching is further reduced to tree search methods and other techniques. The Ullmann algorithm is discussed at length as it forms the basis of tree search algorithms, followed by VF2, Valiente, Koch, Saltz, GraphPi, STwig. In other techniques Nauty and Imniger and Bunke are analyzed. The inexact pattern matching gives a brief overview of how it operates. Literature pertaining to graph pattern matching in relation to agriculture is analyzed revealing the minimal work in this area. The discussion highlights the proficiency of Saltz algorithm in relation to tree pattern matching and discusses how it would outperform the other algorithms. However, it notes that inexact pattern matching is an area of possible research as the data received may contain various inconsistencies among trees, thus an isomorphic mapping may be too stringent for the data.

## CCS CONCEPTS

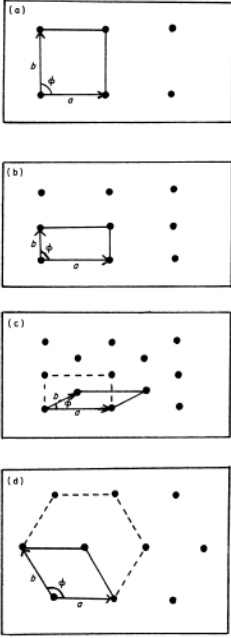
• Precision Agriculture • Optimization •

## KEYWORDS

Graph pattern matching, subgraph isomorphism/ homomorphism

## 1 Introduction

Food security is a large problem faced globally with many factors influencing it such as high population growth, global warming and poor agricultural systems [1]. As the population increases exponentially, the demand for food will follow this trend [2]. Farmers need to ensure that they can keep up with this demand, however they are faced with land constraint problems. Thus, they need to take the best approach that will improve their yield. Precision agriculture is the science of improving crop yields and increasing cost-effectiveness by assisting management decisions using high technology sensors [3]. Drones are used to analyze the land taking high resolution images, we can then process this data to establish findings on productivity of the crops. These findings can be used to evaluate yield, soil condition, plant health, fertilizer and pesticide effect and irrigation [3]. This project serves to analyze the most effective yield that can be produced by various planting patterns. Studies have been conducted over the years to determine the optimal structure and density for plants per unit area. There is a vast array of results as certain experiments yield different results and it is challenging to determine one specific pattern that is the best, holding all other factors constant [4][5][6]. However, we can use precision farming to detect these patterns in orchards and accurately determine the most suitable structures based on the real data. Our project will focus on identifying and classifying various planting patterns and recovering their parameters. Figure 1 demonstrates some of the patterns that will be identified. The input data will consist of images containing tree polygons in order to abstract this data for analysis, we will treat each center point of the tree as the vertex within the graph and the edges will correspond to the Euclidean distance between the trees. We will be looking at applying graph pattern matching in order to solve this problem.



**Figure 1** [6]: An example of various planting patterns

## 2 Background

Graph pattern matching is an extensively researched area and is based on the premise of finding various subgraph isomorphisms or homomorphisms in graphs. A graph  $G$  is defined as  $(V, E, L)$  where  $V$  consists of a set of vertices,  $E \subseteq (V \times V)$  the edges between them and  $L$  is the labeling function which maps a vertex or edge set of labels [7]. Then we assume that each graph contains subgraphs, we want to determine whether these subgraphs match the query graph.

We can further define this graph as either subgraph isomorphism or homomorphism.

**Definition:** An isomorphism between two graphs  $G$  and  $H$  is a bijective mapping  $f: G \rightarrow H$  with the property  $(x, y) \in E_G \Leftrightarrow (f(x), f(y)) \in E_H$  [8].

An homomorphism between two graphs  $G$  and  $H$  is a mapping  $f: G \rightarrow H$  with the property  $(x, y) \in E_G \Rightarrow (f(x), f(y)) \in E_H$  [8]. Basically, isomorphism and homomorphism are a mapping of graph nodes to the sub pattern. The definitions are fairly similar, except for the distinction that the isomorphism is a bijective mapping whereas homomorphism is a one-way mapping. This means the isomorphic definition is much stronger allowing for non-edges to be mapped, as well as edges. Homomorphism relaxes this somewhat as it is not concerned with non-edges. There is research in both areas as both concepts have relevance. Graph pattern matching is used in many areas of research to model various complex objects, such as chemical compounds, social networks, and biological structures [9]. We will classify existing algorithms into two categories (1) exact subgraph matching and (2) inexact subgraph matching [10]. Exact matching can be broken down into two subcategories, first being tree search (Ullman, VF2, Saltz, Valiente and Martinez, Koch, STwig, GraphPi). The two most

common and famous are Ullmann's and VF2. They both use a breadth first search approach on a tree graph; however they have different methods in which they make their algorithms more efficient. The next subcategory is other techniques (Nauty, Imniger and Bunke) and finally we will briefly look at inexact pattern matching.

## 3 Exact matching

The problem that is encountered when using graph pattern matching is that it is NP-complete [10] meaning that for every  $n$  nodes in our graph we need to map  $m$  nodes in our query graph (the pattern) testing whether they are homomorphisms, this results in the worst case  $O(m^n)$  [11]. However, many algorithms have been investigated in attempt to solve this problem.

### 3.1 Tree search

Most algorithms in exact matching use tree search and backtracking to solve the problem [10]. As Ullman's algorithm forms the basis for many subgraph isomorphic matching algorithms, it will be discussed in a fair amount of detail. Ullmann's algorithm attains efficiency by inferentially eliminating successor nodes in the tree search [12]. It also uses effective back tracking and a look ahead procedure to reduce the space [13]. The algorithm obtains feasible matches for each vertex by selecting all vertices in the data graph that are equal and adjacent to the query graph. In the next step, the search space is globally pruned through a simple iterative process to ensure that every candidate vertex in the data graph that matches vertex in the query graph of which all children match the vertices in the query graph [13]. Next query vertices are processed in order of increasing degree, this will allow more branches to be encountered and pruned earlier on. Finally, the algorithm works through the ordered query vertices by isolating the first item in the feasible matches set. It then takes one of its vertices and performs the pruning procedure and recursively calls this procedure on the remaining matches in the feasible set. When the search procedure reaches the depth of the vertices in the query graph each item in the query set should contain a single unique vertex, thus an isomorphic match has been found. The algorithm then backtracks until all paths are explored. VF2 takes a slightly different approach to Ullmann as it uses a "build up" approach as opposed to a "prune down" approach [14]. Thus, search paths are continuously eliminated during the search. It adds a matching pair to the partial matching set before the depth first search [14]. It has been shown to be of complexity  $O(N)$  in respect to the number of nodes, which is much more efficient [10]. It has also been shown to outperform Ullman's algorithm by 1000 times, however it does not achieve as higher performance when run on more complicated graphs, but still outperforms Ullmann [13]. Valiente and Martinez [11] present an algorithm that improves on Ullman's by decomposing the pattern matching. It focuses on a combinatorial search bounded by a pruning operator that has been seen in Ullmann's algorithm. This means that a graph pattern of order 100 is reduced to around 5 searches with the pattern of order 20. Many algorithms are produced in this paper, but it is noted that the C-variants all outperform all the non-c-variants. Koch [15] introduces a method

for finding subgraphs which is based on the transformation of the problem into the clique problem. Cliques are maximal complete subgraphs, wherein each unit in the subset is tied to all others [16]. The algorithm enumerates all cliques that represent connected maximal common subgraphs [15]. This algorithm has been shown to significantly reduce runtimes. Looking at a graph with 100 vertices runtimes around 0.07 are seen. Saltz [14] introduces an interesting algorithm that uses dual simulation in order to achieve a much-improved algorithm. The queries were generated in two methods, one performing breadth first search (BFS) until finding the desired vertices and the other generated a small random graph. This exposed the system to more unpredictable behavior. Saltz [14] goes on to compare their algorithm to Turbo ISO [17], an algorithm that has claimed to outperform many others using candidate region exploration and a combine and permute strategy. Turbo ISO will not be discussed, however it is a possible area of future research. Saltz's research shows that the dual simulation showed a runtime of about 100ms on average over 100 queries. There is another algorithm that also looks at dual simulation by Shi et al [18]. They proposed a high-performance distributed pattern matching system called GraphPi. This is achieved via a 2-cycle based automorphism elimination algorithm, this reduces automorphisms for arbitrary patterns thus improving efficiency. It does this through restriction generator which generates sets of restrictions on the input pattern. The key effectiveness of this algorithm arises from the dual simulation in pruning unnecessary vertices. GraphPi has been shown to outperform Ullmann's matching system by up to two orders of magnitude and can scale to 1,024 nodes [18]. STwig [19] looks at efficient matching on a billion nodes and achieves this in a time of one or two seconds, their approach scales sub linearly with graph size, thus the reason it is able to handle such large graphs. This is very useful as many graphs can reach extreme magnitudes and algorithms such as Ullmann would not be able to handle such capacity.

### 3.2 Other techniques

The most famous algorithm in this area is Nauty, developed by McKay [20]. It is based around group theory and looks to improve the basic canonical algorithm [21]. Two graphs are checked by verifying the equality matrices of their canonical forms. This verification is done in  $O(N^2)$ , but the construction of canonical labeling can result in the exponential case [10]. The average case of this algorithm performs fairly well, however certain algorithms such as VF2 have been shown to outperform it. It does not perform well when exploiting node and edge attribute, however it is highly effective when matching a single graph to a large database [10]. This database matching is seen in Imiger and Bunke's [22] paper whereby they use graph database filtering via utilizing decision trees. It shows that filtering significantly improves the database matching performance.

## 4 Inexact matching

There were various issues that arose in the area of subgraph isomorphic matching such as attempting to preserve structures whilst adhering to the nature of the graph, along with the high computational costs [23]. Also, to be considered is the various deformations that could occur in these graphs. The algorithms may not be able to detect these defects within the graph allowing for incorrect matching. As a result of all these factors, another concept was brought about namely inexact graph matching. This process ignores the identical structure of the subgraph and embarks on a process of penalizing when the edge preserving is not respected. These edges that do not match will be assigned a certain cost, these costs can all be computed to find the most optimal matches. Basically, this algorithm aims to compute a distance that measures the similarity between graphs. There are two parts to inexact pattern matching where optimal matching which aims to find the *global* minimal cost between graphs and approximate matching aims to find the *local* minimum [23].

## 5 Application in Agriculture

A paper by Lasso and Corrales (2016) [25] explores using graph pattern matching to assist in identifying crop disease. They used expert systems to analyse the various crop diseases amongst a large number of tools to analyze and solve the problem. The graph pattern matching in this case is composed of one or more algorithms such as VF2 and others, and is used to find the favorable conditions for the development of crop disease. This will assist farmers in the ability to detect the areas that are most prone and intervene earlier on. Lasso et al. (2017) [26] write another paper on the topic used for coffee rust detection. In this paper, they use the algorithm proposed by Saltz [14] as discussed previously this algorithm utilizes dual simulation. Lasso et al. propose an adaptation of Saltz's algorithm that matches the graph patterns for identifying favorable rust conditions. Other than Lasso's work there is fairly little application of graph pattern matching in agriculture. There is work by Cordella et al. [27] that uses farm land to identify various subgraph shapes. They do a comparison of Ullmann and the VF2 algorithm which provides insight that shows that the Ullmann algorithm scales very badly after it reaches a threshold of 100 nodes. The VF2 algorithm performs well as indicated previously. However, this is not directly applicable to agriculture. Madi [28] has also done work using 2D satellite images to identify Kites, which are ancient archeological builds made of a star like structure with two tails. A process was used to extract the data from the satellite images and transform them into graphs through a process of edge detection and LSD algorithm. A synthetic set of data is also produced to act as a benchmark for detecting the Kite and level of deformities. Although, this data again does not pertain to agriculture, it reveals similarities and looks at the notion of inexact graph pattern matching.

## 6 Discussion and critical comparison

Table 1: Critical Analysis Table

	Time Complexity	Time	Size
Ullman [12]	Best: $O(n^3)$ Worst: $O(N!N^2)$		Smaller graphs (<100 nodes)
VF2 [13]	Best: $O(N^2)$ Worst: $O(N!N)$		Large Graphs
Valiente [11]	$O(m^{\frac{n}{k}})$		
Koch (BKC algorithm) [15]		0.2s (100 nodes)	
Saltz [14]	Worst: $O( V  V ( V_q +3))$	>0.1s (500 000 nodes)	Large graphs
GraphPi [18]	$O(n + m)$	1.94-2.46s	Scale to 1,024 nodes
STwig [19]	$O(m^3)$	33s (for Facebook)	Billion node graph
Nauty [20]		0.21 (126 nodes)	Medium, large graphs
Imniger and Bunke [22]	Very Expensive		
Inexact Graph matching [23]	Polynomial		Large Graphs

We have reviewed many algorithms pertaining to graph pattern matching. When comparing the most efficient tree searching algorithm, we need to look at what best suits our data. We must note that the algorithms we are comparing have been utilized mostly in relation to big databases and in our case, we would need to translate these images into various nodes. A topic that is only briefly touched in this paper. We will compare the various algorithms that have been discussed. Ullmann [12], although having been an important influence in this area would not be applicable, as orchards can reach immense sizes and Ullmann does not have the capacity to handle this effectively. The paper itself is well structured and shows immense insight having been the first of its kind. VF2 [13] shows immense promise as it has been used extensively in industry for many years. However, its outdatedness has given rise to newer and more improved algorithms. Valiente

and Martinez's [11] algorithm shows promise through decomposition; however it does not improve upon Ullmann's algorithm to a high enough degree. Koch's algorithm shows impressive performance in relation to the previous algorithms and introduces an interesting concept through enumerating cliques. However, other algorithms have solved the problem with greater efficiency. The algorithm proposed by Saltz [14] seems to have shown impressive performance. Achieving times under 10ms for very large graphs, whilst maintaining a fairly good level of complexity. This is certainly a solution that we could use when identifying tree patterns. The paper is well written and shows extensive background knowledge on the topic of subgraph isomorphism. GraphPi [18] could also be a viable option, however it does not perform quite to the same degree as Saltz [14] and it does not scale to the capacity that we would need when analyzing an orchard, therefore we will not consider it as a candidate. STwig [19] is also an option that we could utilize as it performs exceptionally well on graphs of enormous magnitudes. However, Saltz[14] discussed this algorithm in their paper and concludes that their algorithm outperforms it by a number of magnitudes. Nauty[20] and Imniger & Bunke[22] use a different approach as opposed to the more common tree search. Nauty performs well and has been shown to outperform VF2, however we have already shown that VF2[13] can be improved upon. Imniger & Bunke [22] are not an effective solution as it is highly computationally expensive. Thus, Saltz's algorithm has been shown to be the most efficient algorithm applicable to exact pattern matching in agriculture. However, speed and efficiency are certainly aspects we strive for there is a certain degree of accuracy that we need to achieve in this experiment. As subgraph isomorphism implies a fairly strict sense of mapping there may be problems that arise when identifying various pattern, if trees are damaged or missing. This could lead to incorrect identification. The solution that could be proposed would be ensuring the tree polygons went through a quality assurance process, thus the with correct detections selected. Alternatively, one could look at using inexact pattern matching which could solve this problem, however it is met with higher runtimes and complexity, but can still process very large graphs effectively. As various inexact pattern matching algorithms are not discussed in this paper, outside of Madi's [28] work. This would be an area to look into for future reference. We have also identified a somewhat gap in literature in relation to graph pattern matching and agriculture. This provides an interesting area of research that can be delved into.

## CONCLUSION

This review has conducted an in-depth analysis on various literature surrounding graph pattern matching. It has given the basis of graph theory, assessing how graphs are formed within the algorithms. Elaborating particularly on tree search algorithms, it assesses various work done over the years looking at subgraph isomorphism and homomorphism. The discussion establishes that the Saltz algorithm is best suited to the problem at hand. However, there is also room to investigate further into inexact graph pattern matching in application to agriculture. There was shown to be a fair

shortage of literature to pertaining to graph pattern matching in the area of agriculture, however there is sufficient research that can be applied to the problem. This would be an interesting area of future research.

## REFERENCES

- [1] United Nations. Sustainable Development Goals. Retrieved May 25, 2021 from <https://www.un.org/sustainabledevelopment/hunger/>
- [2] Qin Zhang. 2016. Precision Agriculture Technology for Crop Farming. Taylor & Francis, Washington.
- [3] Prem Pandey, Prashant Srivastava, Heiko Balzter and Bimal Bhattacharya. 2020. Hyperspectral Remote Sensing: Theory and Applications. *Elsevier*.1(2020), 121-146. DOI: <https://doi.org/10.1016/B978-0-08-102894-0.00009-7>
- [4] M. K. Jagannath. 1978. A MODIFIED MODEL FOR PLANT COMPETITION UNDER VARYING PLANT GEOMETRY. *Current Science*, 47, 8 (April 1978), 251-254. DOI: <https://www.jstor.org/stable/24081256>
- [5] Raza, M.A., Bin Khalid, M.H., Zhang, X. et al. 2019. Effect of planting patterns on yield, nutrient accumulation and distribution in maize and soybean under relay intercropping systems. *Sci Rep* 9, 4947 (2019). DOI: <https://doi.org/10.1038/s41598-019-41364-1>
- [6] M. M. Pant. 1979. Dependence of Plant Yield on Density and Planting Pattern. *Annals of Botany*. 44, 4 (October 1979), 513-516. DOI: <https://www.jstor.org/stable/42756640>
- [7] Tiberio Caetano. Li Cheng. Quoc V. Le. Alex J. Smola. 2007. Learning Graph Matching, IEEE 11th International Conference on Computer Vision, (2007), 1-8, DOI: 10.1109/ICCV.2007.4408838.
- [8] Davis W. Shurbert. 2013. An Introduction to Graph Homomorphisms.
- [9] Jinsoo Lee, Wook-Shin Han, Romans Kasperovics, and Jeong-Hoon Lee. 2012. An in-depth comparison of subgraph isomorphism algorithms in graph databases. *Proc. VLDB Endow.* 6, 2 (December 2012), 133–144. DOI:<https://doi.org/10.14778/2535568.2448946>
- [10] Donatello Conte, Pasquale Foggia, Mario Vento, Carlo Sansone. 2004. Thirty Years Of Graph Matching In Pattern Recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, World Scientific Publishing, (2004), 18 (3), 265-298. DOI: 10.1142/S0218001404003228
- [11] Gabriel Valiente. Conrado Matinez. 1997. An algorithm for graph pattern-matching. Technical University of Catalonia. Department of SoftwareE-08034, Barcelona, Catalonia, Spain.
- [12] J.R.Ullman. 1976. Analgorithm for subgraph isomorphism. *J. Assoc. Comput.Mach.* 23(1976), 31-42. DOI: <https://doi.org/10.1145/321921.321925>
- [13] L. P. Cordella, P. Foggia, C. Sansone, M. Vento. 2001. *An Improved Algorithm for Matching Large Graphs*. Department of Information and Sytems. Università degli Studi di Napoli Via Claudio, 21 – 80125 Napoli, Italy.
- [14] Saltz, Matthew. 2013. *A Fast Algorithm for Subgraph Pattern Matching on Large Labeled Graphs*. Master's thesis. The University of Georgia, Athens, Georgia.
- [15] Ina Koch. 2001.Enumerating all connected maximal common subgraphs in two graphs.Theoretical Computer Science. 250, 1–2, (2001),1-30. DOI: [https://doi.org/10.1016/S0304-3975\(00\)00286-3](https://doi.org/10.1016/S0304-3975(00)00286-3).
- [16] Science Direct. 2005. Clique. Retrieved May 25 from <https://www.sciencedirect.com/topics/mathematics/clique>
- [17] Wook-Shin Han, Jinsoo Lee, and Jeong-Hoon Lee. 2013. Turboiso: towards ultrafast and robust subgraph isomorphism search in large graph databases. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data (SIGMOD '13)*. Association for Computing Machinery, New York, NY, USA, 337–348. DOI:<https://doi.org/10.1145/2463676.2465300>
- [18] Tianhui Shi, Mingshu Zhai, Yi Xu, and Jidong Zhai. 2020. GraphPi: high performance graph pattern matching through effective redundancy elimination. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '20)*. IEEE Press, Article 100, 1–14.
- [19] Zhao Sun, Hongzhi Wang, Haixun Wang, Bin Shao, and Jianzhong Li. 2012. Efficient subgraph matching on billion node graphs. *Proc. VLDB Endow.* 5, 9 (May 2012), 788–799. DOI:<https://doi.org/10.14778/2311906.2311907>
- [20] B.D.McKay. Practical graph isomorphism. *Congressus Numerantium* 30(1981)45-87.
- [21] Stephen G. Hartke and A. J. Radcliffe. 2013. McKay's Canonical Graph Labeling Algorithm. DOI: 10.1090/conm/479/09345
- [22] Christophe Irmige. Horst Bunke. 2004. Graph database filtering using decision trees. *Conference: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on Volume: 3*. DOI: 10.1109/ICPR.2004.1334547
- [23] Kamel Madi. Inexact graph matching : application to 2D and 3D Pattern Recognition. *Computer Vision and Pattern Recognition*. Université de Lyon, (2016). NNT : 2016LYSE1315
- [24] Tsai, W. and K. Fu. "Error-Correcting Isomorphisms of Attributed Relational Graphs for Pattern Analysis." *IEEE Transactions on Systems, Man, and Cybernetics* 9 (1979): 757-768. DOI: 10.1109/TSMC.1979.4310127
- [25] Emmanuel Lasso. Juan Corrales. 2016. Expert System for Crop Disease based on Graph Pattern Matching: A proposal. December 2016 *Revista Ingenierías Universidad de Medellín* 15(29):81-98. DOI: 10.22395/rium.v15n29a5
- [26] Emmanuel Lasso. Thiago Thamada. Carlos Meira. Juan Corrales. 2017. Expert system for coffee rust detection based on supervised learning and graph pattern matching. *Research Gate*. January 2017 *International Journal of Metadata Semantics and Ontologies* 12(1):19. DOI: 10.1504/IJMSO.2017.087641
- [27] Luigi Cordella. Pasquale Foggia. Carlo Sansone. Mario Vento. 2004. A (Sub)Graph Isomorphism Algorithm for Matching Large Graphs. November 2004 *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26(10):1367 – 1372. DOI: 10.1109/TPAMI.2004.75
- [28] Kamel Madi. 2016. Inexact graph matching : application to 2D and 3D Pattern Recognition. PhD Dissertation. Université de Lyon, 2016. NNT : 2016LYSE1315ff. ffilet-01493118f