DISTANCE PRESERVING GRAPHS

By

Emad Zahedi

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Mathematics – Doctor of Philosophy
Computer Science – Doctor of Philosophy

2017

ABSTRACT

DISTANCE PRESERVING GRAPHS

By

Emad Zahedi

The computational complexity of exploring distance properties of large graphs such as real-world social networks which consist of millions of nodes is extremely expensive. Re-computing distances in subgraphs of the original graph will add to the cost. One way to avoid this is to use subgraphs where the distance between any pair of vertices is the same as in the original graph. Such a subgraph is called *isometric*. A connected graph is *distance preserving*, for which we use the abbreviation dp, if it has an isometric subgraph of every order. In this framework we study dp graphs from both the structural and algorithmic perspectives. First, we study the structural nature of dp graphs. This involves classifying graphs based on the dp property and the relation between dp graphs to other graph classes. Second, we study the recognition problem of dp graphs. We intend to develop efficient algorithms for finding isometric subgraphs as well as deciding whether a graph is dp or not.

ACKNOWLEDGEMENTS

Before I start writing I would offer my great thanks to my adviser Dr. Bruce Eli Sagan. I came to Michigan because of you. You are not only an advisor for me, you did a lot for me and I am always thankful to you. As my father told me "you are my father in US", you changed my life in a good direction and I owe you for my whole life. And also I would thank my other advisor Dr. Abdol-Hossein Esfahanian. I cannot thank you enough for giving me a chance as your student. I miss talking to you about my problems and feeling much lighter when I have had a hard time. In hard times people need some somebody and thanks to god that I have had you.

I would also like to extend my sincere gratitude to the rest of my committee: Drs. Eric Torng, Robert W. Bell and Jonathan Hall. Your suggestions and help made the process of completing a dissertation be much more possible.

To my wife, Rui Zahedi, you are the one who did her best to be kind and helpful to me. You were worried about me while you were sick. You took a back seat to my needs. Thanks for being with me and making me happy in this journey. With you I found the power of love makes the impossible possible.

To my parents, Mohammad-Mehdi and Fatemah. You were the only ones with me every day, for every high and low, through my whole life. Your faith in me gave me energy to do all these things. You showed me how to make a good decision in a hard situation. Asheghetoonam!

To Elham, Iman, Ali my sister and brothers who supported me from afar. Your warm words make me strong. You guys always tell me "We miss you." You missed one but I missed all, Alireza, Edris and Ermia, my fiiiraaa, who stood by me in this way.

To my friends, you all did a great thing for me. I would not have gotten through it without you all, nor would my time at Michigan State have been as much fun. Thank you Amin Heydari, Kraig Ehm, Celeste Ehm, Rahan Pouri, Drs. Seyedmehdi Hosseini Nasr, Faramarz Vafaee, Vahid Zamani, Vahid Khademi, Jason Smith and Kaveh Kasebian.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

This is a global fight to get the right people in the right place and we're talking about people with PhDs in engineering, computer science, mathematics.

-Jerry Moran

1.1 Motivation and goals

Subgraph classification has been a challenging problem and many researchers have addressed it by applying different methods to large graph databases. Subgraphs whose properties are the same as the main graph are important since we can use them as smaller samples to analyze large graphs when the full graph analysis is computationally expensive.

Subgraphs in which distances between every pair of vertices are the same as their distance in the main graph are called *isometric*. Isometric subgraphs come into play, for example, in network clustering [25, 26]. A connected graph G is called *distance hereditary* if every connected induced subgraph of G is isometric. These graphs can be shown to be perfect [8, 12]. We relax this property by using a notion called distance preserving.

A connected graph is *distance preserving*, for which we use the abbreviation dp, if it has an isometric subgraph of every possible order. The definition of a distance-preserving

graph is similar to the one for distance-hereditary graphs, but is less restrictive. Because of this, distance-preserving graphs can have a more complex structure than distance-hereditary ones. The distance-preserving property has been applied to real world problems such as, route recommendation systems, logistics planning, and all kinds of shortest-path-related applications that run on resource-limited mobile devices [32].

From the theoretical standpoint, we would like to know

1. the structural nature of dp graphs,

2. how dp graphs are related to other graph classes, and

3. how to clasify graphs based on the dp property.

In regard to these problems, distance-hereditary graphs have been well studied. However, unlike the distance-hereditary graphs, in which the isometric property for the induced subgraphs is universal, in dp graphs this property is an existential property. Because of this, considering the dp property makes such problems more challenging compared to the distance-hereditary property.

From an algorithmic standpoint, we would like to know *i*) which complexity class the dp decision problem belongs to, *ii*) an algorithm to efficiently find an isometric subgraph of order k, for a given k and *iii*) an algorithm to efficiently check whether a graph is dp or not.

It is straightforward to see that the decision problem for dp graphs belongs to NP. Our initial results suggest that detecting the dp property is likely to be NP-Complete. If this is the case, we will want to develop heuristic methods for finding isometric subgraphs of arbitrary order within a graph.

1.2 Definitions and terminology

In this framework all graphs are finite with at least one vertex , undirected, simple, and connected, unless assumed otherwise. For ease of notation, we let $|G|$ be the number of vertices of G. A sequence of vertices $a_0, a_1, \ldots, a_l$ is a *walk of length* l if $a_{i-1}a_i \in E$ for $1 \leq i \leq l$. The walk is a *path* if the a_i are distinct. The *distance* between vertices a, b in G, $d_G(a, b)$, is the minimum length of a path connecting a and b. In the case of a disconnected graph G, we let $d_G(a, b) = \infty$ when there is no path between a and b in G. If the graph G is clear from context, we will use $d(a, b)$, instead of $d_G(a, b)$. A path P from a to b with length $d(a, b)$ is called an *a–b geodesic*. The maximal possible length of all geodesics in G is called the *diameter* of G and denoted by $\text{diam}(G)$. A *cycle* of a graph is a sequence of vertices $v_0, \ldots, v_k$ which are distinct, except for $v_0 = v_k$, and $v_iv_j \in E(G)$ if $|i - j| = 1 \pmod{k}$. The length of a cycle C is its number of edges. The *girth* of a graph G is the smallest length of a cycle in G and denoted by $\text{girth}(G)$.

If G is a graph and $A \subseteq V(G)$ then $G[A]$ denotes the subgraph induced by A. Given two graphs G and H, let $G - H$ be the graph induced by $V(G) \setminus V(H)$. For $A \subseteq V(G)$, we use the notation $G - A$ for the graph obtained after removing A and its incident edges from G. Write $H \subseteq G$ if H is a subgraph of G. For $v \in V(G)$, let $\mathcal{N}_G(v)$ denote the *open neighborhood* of v, that is, the set of vertices adjacent to v. We also define the *closed neighborhood* $\mathcal{N}_G[v] = \mathcal{N}_G(v) \cup \{v\}$. A vertex $v \in V(G)$ is called a *simplicial* vertex if $G[\mathcal{N}(v)]$ is a clique. A graph G is said to have a *simplicial elimination ordering* if there is an ordering $V(G) = \{v_1, \ldots, v_{|V(G)|}\}$ such that v_j is simplicial in $G[v_1, \cdots, v_j]$ for $1 \leq j \leq |V(G)|$.

A subgraph H of a graph G is called an *isometric* subgraph, denoted $H \leq G$, if $d_H(a, b) = d_G(a, b)$ for every pair of vertices $a, b \in V(H)$. A connected graph G with $|G| = n$ is called *distance preserving (dp)* if it has an i-vertex isometric subgraph for every $1 \leq i \leq n$. A connected graph G is called *sequentially distance preserving (sdp)* if there is

an ordering $a_1, \ldots, a_n$ of the vertices of G such that $G - \{a_i\}_{i=1}^{s} \leq G$ for $1 \leq s \leq n$. In this case we say that $a_1, \ldots, a_n$ is an *sdp sequence* for G.

The *lexicographic product* $G[H]$ of graphs G and H is the graph with vertex set $V(G) \times V(H)$ and edge set

$$E(G[H]) = \{(a, x)(b, y) \mid ab \in E(G), \text{ or } xy \in E(H) \text{ and } a = b\}.$$

The *Cartesian product* of G and H is the graph, denoted $G \,\square\, H$, on the vertex set $V(G) \times V(H)$ whose edge set is

$$E(G \,\square\, H) = \{(a, x)(b, y) \mid ab \in E(G) \text{ and } x = y, \text{ or } xy \in E(H) \text{ and } a = b\}.$$

The reader can consult the book of Imrich and Klavzar [18], for more details about products. The rest of our notation is mainly taken from Bondy and Murty's book [3].

1.3 Problem statement

Problem 1.3.1. *Let G be a connected graph, does G contain an isometric subgraph of order k, for some $k \leq |G|$?*

Finding isometric subgraphs is integral to recognizing dp graphs. Our foremost concern is answering the above question in a reasonable time. We believe this problem can not be solved in polynomial time complexity.

One way to determine whether a given subgraph is isometric would be computing the all-pairs shortest paths for the subgraph, and for G as well. The Floyd Warshalls algorithm, which has the time complexity of $O(n^3)$ [15], can be used to address this problem; however, choosing an appropriate subgraph to see whether it is isometric appears to be a complicated and time-consuming process.

The algorithms that we know are almost as bad as a brute-force algorithm and cannot decide the dp property of a graph in reasonable time. As a first attempt, we propose to consider genetic algorithms, with some promising results, but the proposed algorithm can not determine whether some graphs are dp. Thus we propose considering graph structures to refine this algorithm, and the goal/hope is to find an algorithm that would efficiently decide if a graph is dp.

We introduce the graph $G \,\square\, H$ where G and H are the graphs shown in Figures 5.1 and 5.2 respectively. The sparse graph $G \,\square\, H$ contains 80 nodes and for each order k, $k \leq |G|$, there is a limited number of isometric subgraphs in this graph. This graph is a dp graph whose factors are non-dp graphs. We ran the brute-force algorithm on $G \,\square\, H$ and after three days, it could only find around 20 isometric subgraphs. Since there is no other algorithm to compare to our proposed algorithms, recognizing the dp property of $G \,\square\, H$ in a reasonable time, around half a day, became the benchmark for all proposed algorithms. However, such algorithms obviously should not be designed to exploit properties of this particular graph.

Finding a classification of dp graphs based on graph products could be useful since in some cases there is a unique factorization of graphs into prime factors [18]. Analyzing the smaller set of prime dp graphs could make recognition easier. In addition, factors of a Cartesian product graph can be recognized in a linear time with respect to the number of vertices, see [31]. This shows the value of working in this context.

1.4 Overview

In this chapter we introduced isometric subgraphs and dp graphs, addressed our motivations for studying them, discussed an important problem related to dp graphs, and provided some definitions and background. In Chapter 2 we give a literature review for dp graphs, along with some other important related concepts. In Chapter 3 we investigate various ways to construct larger dp graphs from smaller ones. We prove that if a graph

does not contain any induced cycles of length 5 or greater, then it is sdp and thus dp. We also consider the distance preserving property on graphs with a cut vertex. In contrast, Chapter 4 focuses on non-dp graphs. We introduce two families of non-dp graphs based on girth and induced cycles. In Chapter 5 we prove certain results about products of dp graphs. In chapter 6 we generalize lexicographic product and consider the dp property in this context. in Chapter 7 we investigate finding isometric subgraphs, by proposing an evolutionary algorithm to analyze the dp property of weighted graphs. In Chapter 8 we propose an algorithm using the first algorithm, in which some local search strategies are amalgamated to improve convergence speed. In addition, a selection operator is proposed to prevent premature convergence. In Chapter 9 we discuss ongoing investigations.

Chapter 2

Literature review

Mathematicians are like managers, they want improvement without change.

-Edsger Dijkstra

Computing distance properties of large graphs such as real-world social networks which consist of millions of nodes is extremely expensive. Recomputing distances in subgraphs of the original network will be even more costly. A solution to remedy this issue would be to find isometric subgraphs.

One family of graphs which has been studied in the literature involving isometric subgraphs is the set of distance-hereditary graphs. A *distance-hereditary* graph is a connected graph in which every connected induced subgraph of G is isometric [16]. Distance-hereditary graphs have been considered in various papers [1, 7, 14] since they were first studied by Howorka [16]. This class of graphs was originally mentioned by Sachs [28] while working with perfect graphs.

Hammer and Maffrey proposed a linear time recognition algorithm for distance-hereditary graphs [14]. Damiand et al, showed this algorithm is incorrect. They provided their own linear time recognition algorithm by decomposing a graph into a sequence of pendant vertex and twin operations [7] as illustrated in Figure 2.1. Using distance hereditary graphs can simplify certain problems, for example, determining whether they are Hamiltonian can be done in linear time [17].

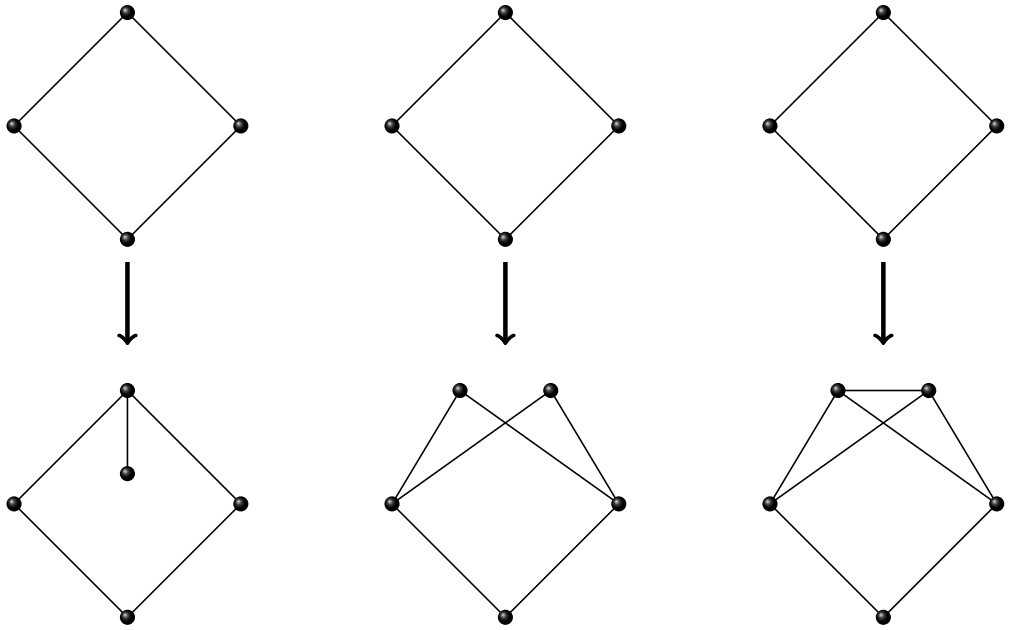


Figure 2.1 Pendant Vertex and Twin Operations

Distance preserving graphs been applied to real world problems such as route recommendation systems, logistics planning, and shortest-path-related applications that run on resource-limited mobile devices [32]. Esfahanian and Nussbaum explored various properties of dp graphs [25] and potential applications in clustering of social networks [26]. It is easy to see that trees are dp by removing leaves. Conditions under which adding a vertex to a dp graph preserves the dp property are given in [33] and discussed below in Chapter 3. By applying this construction recursively to K_1, one can construct various families of dp graphs such as chordal graphs (which include trees). In contract to the acyclic case, the presence of certain cycles can cause a graph not to be dp. In the same paper it is also shown that if G is a graph with girth$(G) \geq 5$ and every vertex is either a cut vertex or in a cycle, then G does not have any isometric subgraph of order $|V(G)| - 1$ and so is not dp. This is discussed in Chapter 4.

Esfahanian et al. [27], constructed regular distance-preserving graphs of all possible orders and degrees of regularity. By modifying the Havel-Hakimi algorithm, they also construct distance preserving graphs for certain other degree sequences.

Khalifeh et al. [19] gave a necessary and sufficient condition for the lexicographic product of two graphs to be dp. This condition implies that if G is dp then the lexicographic product of G and any graph H is dp. Moreover, they characterized all isometric subgraphs of the lexicographic product of two arbitrary graphs, and also they proved that the Cartesian product of two graphs is sdp if and only if its factors are. These results will be found in Chapter 5.

Chapter 3

Distance-preserving graphs

A mathematician is a device for turning coffee into theorems.
-Paul Erdos

It is easy to see that trees are dp by removing leaves. In this chapter we will find various other families of dp graphs.

3.1 Chordal graphs

A *chordal* graph is a graph in which any cycle of length four or more has a chord, that is, an edge of the ambient graph connecting two vertices not adjacent along the cycle. A graph G is chordal if and only if G has a simplicial elimination order [9]. The following lemma will permit us to prove that a chordal graphs are sdp.

Lemma 3.1.1. *Let v be a simplicial vertex in G. If $G - v$ is a dp graph then G is dp.*

Proof. Let $G' = G - v$ and $n = |V|$. We claim it suffices to show that G' is an isometric subgraph of G. Indeed, G' will be an isometric subgraph of G of order $n - 1$. And for $k < n - 1$, the fact that G' is dp implies that there is an isometric subgraph H of G'. But then H is also isometric in G because being an isometric subgraph is a transitive relation.

To show G' is isometric in G, consider $x, y \in V(G')$. Since v is simplicial in G, $G[\mathcal{N}_G(v)]$ is an induced complete subgraph of G. We claim that in G, any x-y geodesic can

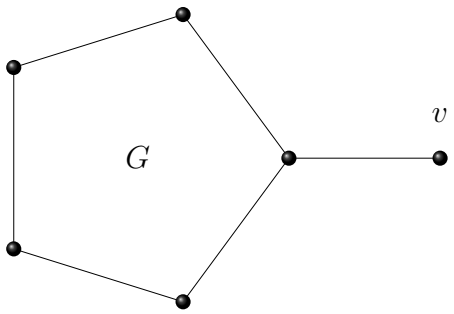


Figure 3.1 A counterexample graph G to the converse of Lemma 3.1.1.

not contain v. This implies that $d_G(x, y) = d_{G'}(x, y)$, and thus G' wi be isometric as desired. Suppose, towards a contradiction, that $P : x, \ldots, u, v, w, \ldots, y$ is an x-y geodesic in G then u, w lie in $\mathcal{N}_G(v)$. But $G[\mathcal{N}_G(v)]$ is complete, so $uw \in E(G)$ and $\hat{P} : x, u_1, \ldots, u, w, \ldots, y$ is another path from x to y in G which is shorter than P. This contradicts the fact that P is an x-y geodesic and finishes the proof. □

The converse of the Lemma 3.1.1 is not true. In Figure 3.1, it is easy to check that G is a dp graph and $G[\mathcal{N}_G(v)]$ is complete. But $G - v = C_5$ is not dp since removal of any vertex of the cycle results in a subgraph which is a path and not isometric in C_5.

The next Lemma can be proven in a way similar to the Lemma 3.1.1.

Lemma 3.1.2. *Let v be a simplicial vertex in G. If $G - v$ is a sdp graph then G is sdp.*

The next corollary generalizes the fact mentioned previously that trees are dp.

Corollary 3.1.3. *Chordal graphs are sdp.*

Proof. Our proof is by induction on $n = |V|$. The result is clear when $n = 1$. Given a chordal graph G, let $v_1, v_2, ..., v_n$ be a simplicial elimination order for its vertices. By induction $G' = G[v_1, v_2, .., v_{n-1}]$ is sdp and v_n is simplicial in G. Using Lemma 3.1.2 implies that G is sdp. □

We can relax the condition in Lemma 3.1.1 as follows.

Theorem 3.1.4. *Let G contain a vertex v such that every pair of non-adjacent $u, w \in \mathcal{N}(v)$ are in a 4-cycle in G. If $G - v$ is dp then so is G.*

Proof. As in the proof of Lemma 3.1.1, it suffices to show that $G' = G - v$ is an isometric subgraph of G. So take $x, y \in V(G')$ and an x-y geodesic P in G. It suffices to show that there is an x-y geodesic in G' of the same length. If P does not contain v, then it is a geodesic in G' and we are done. If P contains v, say $P : x, \ldots, u, v, w, \ldots, y$. If uw is an edge of G then we derive a contradiction as in the proof of Lemma 3.1.1. If u and w are not adjacent then, by the cycle hypothesis, there must be a vertex $z \neq v$ with $uz, zw \in E(G)$. So, by the choice of z, the path $\hat{P} : x, \ldots, u, z, w, \ldots, y$ is an x-y geodesic in G' with the same length as P. □

3.2 All 4-chordal graphs are distance preserving

We say a graph is *k-chordal* if the largest induced cycle is of length k. So the 3-chordal graphs are just the chordal graphs previously defined. It was shown in Section 3.1 that 3-chordal graphs are sdp. This is shown using the well known property that all chordal graphs have a simplicial ordering. This property is generalized to k-chordal graphs in [20], using the notion of a k-simplicial ordering.

Definition 3.2.1. *A vertex v of a graph G is* weakly k-simplicial *if $\mathcal{N}_G(v)$ induces a clique in $(G - v)^{k-2}$. Furthermore, v is* k-simplicial *if it is weakly k-simplicial and for each non-adjacent pair x, y in $\mathcal{N}_G(v)$, every chordless x, y-path whose interior is entirely in $G - \mathcal{N}_G[v]$ has at most $k - 2$ edges. A vertex ordering $v_1, \ldots, v_n$ of G is a* (weakly) k-simplicial ordering *if v_i is (weakly) k-simplicial in $G[v_i, \ldots, v_n]$.*

We use this generalized simplicial ordering to prove the conjecture in [25] that all 4-chordal graphs are distance preserving. In order to do this we need the main result from [20], which we present next. Note that there is a third equivalent statement in the original theorem which we omit here as we do not require it for our results.

Theorem 3.2.2. *[20, Theorem 1] Consider a graph G and integer $k \geq 3$. The graph G is k-chordal if and only if G has a k-simplicial ordering.*

Before proving the main result of this section we present the following lemma, which is a generalization of Lemma 3.1.1.

Lemma 3.2.3. *Consider a graph G and vertex $v \in V(G)$. The graph $G - v$ is isometric if and only if v is weakly 4-simplicial.*

Proof. Suppose v is weakly 4-simplicial. This implies that $\mathcal{N}_G(v)$ induces a clique in $(G-v)^2$, that is, any pair $x, y \in \mathcal{N}_G(v)$ have a distance of at most 2 in $G - v$. Consider any path P which contains v in its interior. There must be a subpath $x - v - y$ of P, where $x, y \in \mathcal{N}_G(v)$. Because v is 4-simplicial we know that x and y are either neighbours or have a common neighbour $z \neq v$. Therefore, we can either remove v or replace it with z to get a path that is at least as short as P lying in $G - v$. It follows that $G - v$ is isometric.

Suppose $G - v$ is isometric. Consider any pair $u, w \in \mathcal{N}_G(v)$, then we know that $d_G(u, w) \leq 2$ which implies $d_{G-v}(u, w) \leq 2$. Therefore, $\mathcal{N}_G(v)$ induces a clique in $(G-v)^2$, so v is weakly 4-simplicial. □

The following proposition follows immediately from Lemma 3.2.3.

Proposition 3.2.4. *A graph is sdp if and only if it admits a weakly 4-simplicial ordering.*

Now we have all we need to prove Conjecture 5.2 of [25]:

Theorem 3.2.5. *Any 4-chordal graph is sdp, and thus dp.*

Proof. Applying Theorem 3.2.2 with $k = 4$ shows that for any 4-chordal graph there is a 4-simplicial ordering of the vertices. Moreover, Proposition 3.2.4 implies this ordering is an sdp ordering. □

The converse of Theorem 3.2.5 is not true. The graph in Figure 3.2 is not 4-chordal, because it contains an induced 5-cycle, so by Theorem 3.2.2 the graph cannot have a 4-simplicial ordering. However, the ordering given by the vertex labels is a weakly 4-simplicial

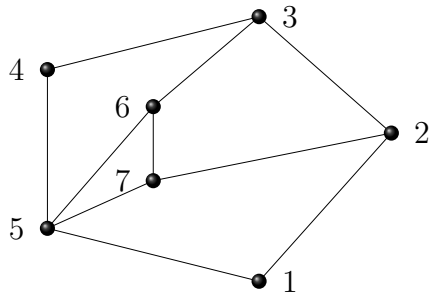


Figure 3.2 A non-4-chordal graph that is sdp. The vertex labels give an sdp ordering.

ordering, so the graph is sdp. To see the ordering is not 4-simplicial, note that the vertex labelled 1 is not 4-simplicial because the path $2-3-4-5$ violates the 4-simplicial condition. Combining Theorem 3.2.5 with Corollary 3.1.3 gives the following corollary:

Corollary 3.2.6. *Any dp graph that is not sdp must contain an induced cycle of length* $k \geq 5$.

3.3 Separable graphs

A connected graph is said to be *separable* if it can be disconnected by removing a vertex, which we call a *cut vertex*. In this section we consider the distance preserving property in separable graphs. A separable graph can be represented in the following way:

Definition 3.3.1. *Consider two graphs* G *and* H*, with a single common vertex* x*. Let* $G +_x H$ *be the union of* G *and* H.

So $G +_x H$ is a separable graph with a cut vertex x. We characterise the isometric subgraphs of $G +_x H$. To do this we introduce the following lemma.

Lemma 3.3.2. *Consider a graph* $G +_x H$ *and two induced subgraphs* $H' \subseteq H$, $G' \subseteq G$, *with* $x \in V(G') \cap V(H')$, *then:*

$$G' +_x H' \leq G +_x H \quad \text{if and only if} \quad H' \leq H \text{ and } G' \leq G.$$

Proof. The result follows easily from the observation that if $u, v \in G +_x H$ then a path P from u to v is a geodesic if and only if

1. $u, v \in G$ and P is a geodesic in G (similarly for $u, v \in H$), or
2. $u \in G$, $v \in H$ and the u–x and x–v portions of P are geodesics in G and H, respectively (similarly for $u \in H$ and $v \in G$).

This completes the proof. □

Now we state the main result of this section which is a direct corollary of the previous lemma.

Theorem 3.3.3. *A graph $G +_x H$ is dp if and only if for every k, $1 \leq k \leq |G +_x H|$, we have*

1. *G or H contains an isometric subgraph of order k, or*
2. *there are $G' \leq G$ and $H' \leq H$ both containing x with $|G'| + |H'| - 1 = k$.*

3.4 Chapter summary

In this chapter, we investigated conditions under which adding a vertex to a dp graph preserves the dp property. By applying this construction recursively to K_1, one can construct various families of dp graphs. We used this method to prove that chordal graphs (which include trees) are dp. Next we gave an equivalent condition to sequentially distance preserving based upon simplicial orderings. Using this condition, we proved that if a graph does not contain any induced cycles of length 5 or greater, then it is sdp and thus dp. Finally, we considered the distance preserving property on graphs with a cut vertex.

Chapter 4

Non-distance preserving graphs

Mathematics is a game played according to certain simple rules with meaningless marks on paper.

-David Hilbert

It is conjectured in [25] that almost all graphs are non-dp. So understanding this class is a logical step towards a full classification of the class of dp graphs. As we have noted, trees are dp. But the presence of certain cycles can cause a graph not to be dp. As also previously mentioned, the 5-cycle C_5 is not dp. In this section two families of non-dp graphs will be considered.

4.1 Girth

We now give a condition on the girth of a graph G which implies that it is not dp. Note the contrast with the cycle condition in Theorem 3.1.4.

Theorem 4.1.1. *Let G be a graph such that* $\text{girth}(G) \geq 5$ *and such that every vertex is either a cut vertex or in a cycle. Then G is not dp.*

Proof. Assume that G is dp so we can delete a vertex, say v, to obtain an isometric subgraph of order $|V(G)| - 1$ in G. Now v can not be a cut vertex since a disconnected subgraph of G can not be isometric. Therefore v belongs to a cycle C and there exist two vertices u, w in C such that uvw is a path in G. By assumption $\text{girth}(G) \geq 5$ and so G does not contain

a 3-cycle. Thus $uw \notin E(G)$ and $d_G(u,w) = 2$. Since $G - v$ is isometric, $d_{G-v}(u,w) = 2$ and consequently there is a vertex $\hat{v} \in V(G-v)$ so that $u, \hat{v}, w$ is a path in $G - v$. This implies $u, v, w, \hat{v}, u$ is a 4-cycle in G which contradicts girth$(G) \geq 5$. Since the vertex v was arbitrary, G has no isometric subgraph of order $|V(G)| - 1$ and so is not dp. □

The fact that girth$(G) \leq 4$ does not imply G is dp and containing a k-cycle, for $k \geq 5$, is not a necessary condition for a graph to be non dp, as we see in Figure 4.1.

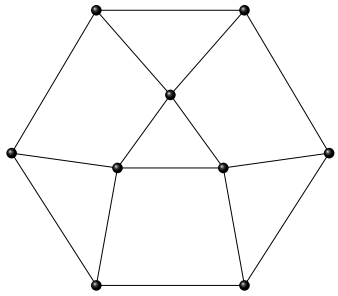


Figure 4.1 A non-dp graph G with girth$(G) \leq 4$.

Note that $G +_x H$ can be either dp or not for dp graphs G and H. For example, let $G = K_m$ and $H = K_n$, then G and H are clearly dp, and it is easy to check $G +_x H$ is a dp graph. On the other hand if G and H are 5-cycles with a pendant edge, and we let x be the vertex of degree one in both graphs, then one can see G and H are dp, but by Theorem 4.1.1, $G +_x H$ is not a dp graph.

4.2 Maintaining the non-dp property

By Corollary 3.1.3 and Theorem 3.2.5, we know that a non-dp graph must contain a cycle of length $k > 4$. The simplest non-dp graphs are the cycle graphs C_k, for all $k \geq 5$. We investigate how to add vertices to cycle graphs whilst maintaining the non-dp property. To this end, a family of non-dp graphs is defined.

Consider the cycle C_k and a set of vertices A, with $|A| = \ell$, such that $A \cap V(C_k) = \emptyset$. For each $a \in A$, select three consecutive vertices of C_k and join a to at least one of the

three selected vertices. Let $\mathcal{C}_{k,\ell}$ denote the family of graphs that can be constructed in this way. Given a graph $G \in \mathcal{C}_{k,\ell}$, let $C(G)$ be the original cycle vertices of G and $A(G)$ the added vertices. Let $C(H) = V(H) \cap C(G)$ for $H \subseteq G$. Note that the addition of the vertices to the cycle graph cannot change the distance between any pair of vertices in $C(G)$, so $C_k \leq\ G$.

Recall that we label the vertices of C_k as $v_1, \ldots, v_k$, and let $v_{k+1} := v_1$ and $v_0 := v_k$. So there is an edge between two vertices v_i and v_j if and only if $i = j \pm 1$.

Theorem 4.2.1. *If $k > 2(\ell + 2)$, then any graph in $\mathcal{C}_{k,\ell}$ is non-dp.*

Proof. Consider a graph $G \in \mathcal{C}_{k,\ell}$. If an added vertex a is connected to two cycle vertices v_{i-1} and v_{i+1}, then the removal of either a or v_i results in isomorphic subgraphs. Therefore, when constructing an isometric subgraph of G, by removing a set of vertices of G, we can assume that a is always removed before v_i. Also recall that the added vertices do not alter the distance between any of the cycle vertices. Combining these two points implies that given a graph $H \leq G$ there is a geodesic path in H between any two elements of $C(H)$ that is entirely contained in $H[C(H)]$. Therefore, if $H \leq C_{k,\ell}$, then $H[C(H)] \leq C_k$.

We show that there is no isometric subgraph of G with order $\lfloor \frac{k}{2} \rfloor + 2$. Suppose for a contradiction that such a subgraph does exist, we denote it H. We know that $\ell < \frac{k}{2} - 2$, so to obtain H we must remove a set of s cycle vertices, where $\lceil \frac{k}{2} \rceil - 2 > s > 0$. However, this implies that $C(H)$ has t vertices, where $k > t > \lfloor \frac{k}{2} \rfloor + 2$, and it is straightforward to see that there is no isometric subgraph of C_k with t vertices. Therefore, H is not isometric, so G is non-dp. □

Note that the converse of Theorem 4.2.1 is not true. For example in Figure 4.2, the graph G is not dp but $k = 10 \not> 10 = 2(\ell + 2)$. An interesting question, which we leave open, is:

Open problem 4.2.2. *How can we add vertices to more general non-dp graphs to preserve the non-dp property?*

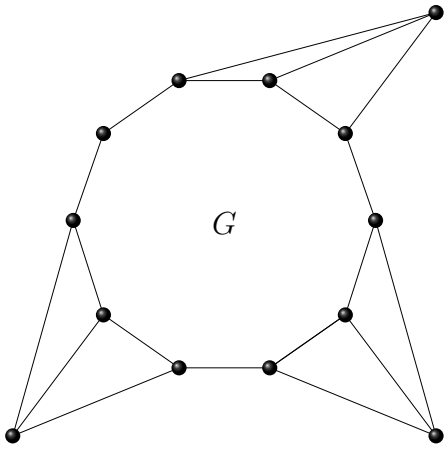


Figure 4.2 A counterexample to the converse of the Theorem 4.2.1

4.3 Chapter summary

The presence of certain cycles can cause a graph not to be dp. In this chapter, we showed that if G is a graph with girth$(G) \geq 5$ and every vertex is either a cut vertex or in a cycle, then G does not have any isometric subgraph of order $|V(G)| - 1$ and so is not dp. We also defined a family of graphs, namely $C_{k,l}$, and proposed a condition under which $C_{k,l}$ is not dp. We end up with an open problem about building non-dp graphs.

Chapter 5

Distance-preserving graphs and graph products

To me, mathematics, computer science, and the arts are insanely related. They're all creative expressions.

-Sebastian Thrun

The purpose of this chapter is to investigate what happens to the dp and sdp properties when taking products of graphs.

5.1 Lexicographic products of graphs

In this section we derive a necessary and sufficient condition for a connected graph $G[H]$ to be distance preserving. Furthermore we will find all the isometric subgraphs of $G[H]$.

We first need a lemma about the distance function in $G[H]$.

Lemma 5.1.1. *Suppose G is a graph with $|G| \geq 2$ and H is an arbitrary graph.*

(a) Let G be connected. For distinct vertices (a,x) and (b,y) in $G[H]$,

$$d_{G[H]}\big((a,x),(b,y)\big)=\begin{cases} d_G(a,b) & if\ a\neq b,\\ 2 & if\ a=b,\ xy\notin E(H),\\ 1 & if\ a=b,\ xy\in E(H).\end{cases}$$

(b) The graph $G[H]$ is connected if and only if G is connected.

Proof. To see part (a), note in the case $a \neq b$ that, by the definition of lexicographic product, $a = a_0, a_1, \cdots, a_l = b$ is a geodesic in G if and only if $(a_0, x), (a_1, x_1), \cdots, (a_{l-1}, x_{l-1})$, (a_l, y) is a geodesic in $G[H]$ where x_i is any vertex of H for $0 < i < l$. Thus $d_G(a,b) = d_{G[H]}\big((a,x),(b,y)\big)$. In the second case, since G is connected and $|G| > 1$ we have $ac \in E(G)$ for some c, by definition of lexicographic product we have a path $(a,x),(c,z),(a,y)$ where z is any vertex of H. Moreover $(a,x)(b,y) \notin E(G[H])$ so the distance must be 2. For the third case, $(a,x)(b,y) \in E(G[H])$.

To see part (b), If G is connected with $|G| \geq 2$ then all the distances are finite in G. By part (a) this happens if and only if all the distances are finite in $G[H]$ which means $G[H]$ is connected. For the converse, suppose $a, b \in V(G)$. Since $G[H]$ is connected there is a walk $(a,x_0)(a_1,x_1),\ldots,(a_{l-1},x_{l-1}),(b,x_l)$ in $G[H]$ where x_i is any vertex of H for $0 \leq i \leq l$. It follows that $a, a_1, \ldots, a_{l-1}, b$ will be a walk in G once subsequences of adjacent equal vertices have been replaced by a single copy of the repeated vertex. Thus G is connected. □

In order to state the main theorem of this section, we need some notation. Let

$$\mathrm{dp}(G) = \{k \mid G \text{ has an isometric subgraph with } k \text{ vertices}\}.$$

If a, b are integers with $a < b$, then let $[a, b] = \{a, a+1, a+2, \ldots, b\}$. So a graph G is dp if and only if $\mathrm{dp}(G) = [1, |G|]$. Two elements $a, b \in \mathrm{dp}(G)$ *bound a non-dp interval* if the set of integers c with $a < c < b$ is nonempty and consists only of elements not in $\mathrm{dp}(G)$.

Finally, the *projection* of a subgraph K of $G[H]$, denoted $\pi(K)$, is the induced subgraph of G whose vertex set is

$$\mathrm{V}\big(\pi(K)\big) = \{a \mid (a, x) \text{ is a vertex of } K\}.$$

Theorem 5.1.2. *Let G be a connected graph with $|G| \geq 2$ and H be an arbitrary graph with $|H| = n$. Then*

$$G[H] \text{ is dp if and only if } b \leq an + 1$$

for every pair $a, b \in \mathrm{dp}(G)$ bounding a non-dp interval.

Proof. We claim, for an induced subgraph K of $G[H]$ with $\pi(K)$ having at least two vertices,

$$\pi(K) \leq G \text{ if and only if } K \leq G[H]. \tag{5.1}$$

To prove the forward direction of the claim, assume that $\pi(K) \leq G$ and consider distinct vertices $(a, x), (b, y) \in V(K)$. If $a \neq b$ then, using the same ideas as in the proof of the first case in Lemma 5.1.1(a), we see that $d_{\pi(K)}(a, b) = d_K\big((a, x), (b, y)\big)$. Using $\pi(K) \leq G$ and the lemma itself gives

$$d_K\big((a, x), (b, y)\big) = d_{\pi(K)}(a, b) = d_G(a, b) = d_{G[H]}\big((a, x), (b, y)\big)$$

as desired. If $a = b$ and $xy \notin E(H)$, then a similar proof shows $d_K\big((a, x), (b, y)\big) = 2 = d_{G[H]}\big((a, x), (b, y)\big)$. Finally, if $a = b$ and $xy \in E(H)$, then since K is induced we have $d_K\big((a, x), (b, y)\big) = 1 = d_{G[H]}\big((a, x), (b, y)\big)$.

Conversely, if $K \leq G[H]$, then we must show

$$d_{\pi(K)}(a,b) = d_G(a,b)$$

for any two distinct vertices a, b in $\pi(K)$. Again using the ideas in the proof of the first case in Lemma 5.1.1(a), we see that $d_{\pi(K)}(a,b) = d_K\big((a,x),(b,y)\big)$ for any $x, y \in V(H)$. Using $K \leq G[H]$ and the lemma itself, we have

$$d_{\pi(K)}(a,b) = d_K\big((a,x),(b,y)\big) = d_{G[H]}\big((a,x),(b,y)\big) = d_G(a,b).$$

To prove the theorem suppose that $|\pi(K)| = c$, $|G| = m$ and $|H| = n$ so that $|G[H]| = mn$. By definition of projection $c \leq |K| \leq cn$. Also every connected graph with at least two vertices has isometric subgraphs with one vertex and with two vertices. So by equation (5.1), $G[H]$ will be dp if and only if

$$\bigcup_{c \in \mathrm{dp}(G)} [c, cn] = [1, mn].$$

Since $1, 2, m \in \mathrm{dp}(G)$, the last equality is equivalent to $[a, an] \cup [b, bn]$ being an interval for every pair $a, b \in \mathrm{dp}(G)$ bounding a non-dp interval. But this is equivalent to $b \leq an+1$. $\square$

We see from the theorem that $G[H]$ being dp only depends on the size of H and the non-dp intervals in G. The next result is an immediate corollary of the previous theorem.

Corollary 5.1.3. *If G is dp with $|G| \geq 2$ then so is $G[H]$ for any graph H.*

Similarly, the next result follows easily from Lemma 5.1.1 and equation (5.1).

Corollary 5.1.4. *For a connected graph G with $|G| \geq 2$ and an induced subgraph K of $G[H]$,*

$$K \leq G[H] \text{ if and only if } \begin{cases} \pi(K) \leq G & \text{if } |\pi(K)| \geq 2, \\ \operatorname{diam}(K) \leq 2 & \text{if } |\pi(K)| = 1. \end{cases}$$

5.2 Cartesian product graphs

We now turn to Cartesian products and the sdp property. We first need some notation and a few well-known results. A *removal set* in G is a set of vertices of G whose removal gives an isometric subgraph, let

$$\mathrm{DP}'(G) = \big\{ A \subseteq V(G) \,\big|\, G - A \leq G \big\} \quad \text{and} \quad \mathrm{dp}'(G) = \big\{ |A| \,\big|\, A \in \mathrm{DP}'(G) \big\}.$$

Proposition 5.2.1. *[4] Suppose G and H are graphs,*

(a) If (a, x) and (b, y) are vertices of a Cartesian product $G \,\square\, H$ then

$$d_{G \,\square\, H}\big((a, x), (b, y)\big) = d_G(a, b) + d_H(x, y).$$

(b) A path $(a_0, x_0) \dots (a_l, x_l)$ is geodesic in $G \,\square\, H$ if and only if $a_0 \dots a_l$ is a geodesic in G after removal of repeated vertices and similarly for $x_0 \dots x_l$ in H.

Next we consider isometric Cartesian product subgraphs of a Cartesian product graph.

Lemma 5.2.2. *Suppose G' and H' are nonempty subgraphs of G and H respectively, then $G' \,\square\, H' \leq G \,\square\, H$ if and only if $G' \leq G$ and $H' \leq H$.*

Proof. For the forward direction using the assumption and proposition 5.2.1(a) we have

$$d_{G'}(a, b) + d_{H'}(x, y) = d_{G' \,\square\, H'}\big((a, x), (b, y)\big) = d_{G \,\square\, H}\big((a, x), (b, y)\big) = d_G(a, b) + d_H(x, y),$$

for every pair of vertices $(a,x),(b,y) \in V(G' \,\square\, H')$. As any distance in a subgraph is greater than or equal to the corresponding distance in the original graph, we get $d_{G'}(a,b) = d_G(a,b)$ and $d_{H'}(x,y) = d_H(x,y)$.

Conversely, suppose G' and H' are isometric subgraphs, by proposition 5.2.1(a) we have

$$d_{G' \,\square\, H'}\big((a,x),(b,y)\big) = d_{G'}(a,b) + d_{H'}(x,y) = d_G(a,b) + d_H(x,y) = d_{G \,\square\, H}\big((a,x),(b,y)\big),$$

for each pair of vertices $(a,x),(b,y) \in V(G' \,\square\, H')$. This complete the proof. $\square$

We now prove a lemma about removal sets of vertices.

Lemma 5.2.3. *For nonempty subsets A and B in the vertex set of graphs G and H respectively, $A \times B \in \mathrm{DP}'(G \,\square\, H)$ if and only if $A \in \mathrm{DP}'(G)$ and $B \in \mathrm{DP}'(H)$.*

Proof. To prove the forward direction, we show $A \in \mathrm{DP}'(G)$ as $B \in \mathrm{DP}'(H)$ is similar. Let $a,b \in V(G-A)$ and $x \in B$. By Proposition 5.2.1(b), the (a,x)–(b,x) geodesics in $(G-A) \,\square\, B$ are the same as the geodesics in $(G \,\square\, H) - (A \times B)$. Now using this fact, Proposition 5.2.1(a), and the assumption in this direction

$$\begin{aligned} d_{G-A}(a,b) &= d_{(G-A) \,\square\, B}\big((a,x),(b,x)\big) \\ &= d_{(G \,\square\, H)-(A\times B)}\big((a,x),(b,x)\big) \\ &= d_{G \,\square\, H}\big((a,x),(b,x)\big). \end{aligned}$$

Finally, applying Proposition 5.2.1(a) again shows that the last distance equals $d_G(a,b)$ as desired.

To see the backward direction, first note that $(G \,\square\, H) - (A \times B) = ((G-A) \,\square\, H) \cup (G \,\square\, (H-B))$. So it suffices to show that

$$d_{(G \,\square\, H)-(A\times B)}\big((a,x),(b,y)\big) = d_{G \,\square\, H}\big((a,x),(b,y)\big)$$

for any (a,x) in $(G-A)\,\square\,H$ and (b,y) in $G\,\square\,(H-B)$ since Lemma 5.2.2 takes care of the other possibilities. Clearly there is a path $(a,x),\ldots,(a,y)$ with length $d_H(x,y)$ in $(G-A)\,\square\,H$, and also $(a,y),\ldots,(b,y)$ with length $d_G(a,b)$ in $G\,\square\,(H-B)$. The concatenation of these paths is a path from (a,x) to (b,y) in $(G\,\square\,H)-(A\times B)$ of length $d_G(a,b)+d_H(x,y)=d_{G\,\square\,H}\big((a,x),(b,y)\big)$ and so must be a geodesic. This concludes the proof. □

We are now in a position to prove the main theorem of this section.

Theorem 5.2.4. *The product $G\,\square\,H$ is sdp if and only if G and H are sdp.*

Proof. For the forward direction, we will prove that G is sdp, the proof for H being similar. Take an sdp sequence of vertices for $G\,\square\,H$. Fix $x\in H$ and consider the subsequence $(a_1,x),(a_2,x),\ldots,(a_n,x)$ where $n=|G|$. We claim that $a_1,a_2,\ldots,a_n$ is an sdp sequence for G. Indeed, let $G'=G-\{a_i\}_{i=1}^{s}$ and let K' be $G\,\square\,H$ with the vertices through (a_s,x) removed so that $G'\,\square\,\{x\}\subseteq K'$. Now if $b,c\in V(G')$ then, by Proposition 5.2.1(b), P is a b–c geodesic in G' if and only if $P\,\square\,\{x\}$ is a (b,x)–(c,x) geodesic in K'. From this fact, the sdp property of the original sequence, and Proposition 5.2.1(a) we obtain

$$d_{G'}(b,c)=d_{K'}\big((b,x),(c,x)\big)=d_{G\,\square\,H}\big((b,x),(c,x)\big)=d_G(b,c)$$

as desired.

For the converse, suppose that if $a_1,\ldots,a_n$ and $b_1,\ldots,b_m$ are sdp sequences for G and H, respectively. Then it follows easily from Lemma 5.2.3 and the transitivity of the isometric subgraph relation that

$$(a_1,b_1),\ldots,(a_n,b_1),(a_1,b_2),\ldots,(a_n,b_2),\ldots,(a_1,b_m),\ldots,(a_n,b_m)$$

is an sdp sequence for $G\,\square\,H$. □

Now we prove a result which follows a similar argument to that of Theorem 5.2.4.

Proposition 5.2.5. *If G is sdp and H is dp, then $G \,\square\, H$ is dp.*

Proof. Since G is sdp there is an ordering $v_1, \ldots, v_{|G|}$ of $V(G)$ such that $\{v_i\}_{i=1}^{s} \in \mathrm{DP}'(G)$, for every $1 \leq s \leq |G|$. Moreover, since H is distance preserving there is a set $A_j \in \mathrm{DP}'(H)$ with $|A_j| = j$, for every $1 \leq j \leq |H|$. By Lemma 5.2.3 we know that $v_1 \times A_j \in \mathrm{DP}'(G \,\square\, H)$, for all $1 \leq j \leq |H|$. Furthermore, Lemma 5.2.3 implies that $v_2 \times A_j \in \mathrm{DP}'((G - \{v_1\}) \,\square\, H)$, for all $1 \leq j \leq |H|$, so by the transitivity of the isometric property we get $(v_1 \times H) \cup (v_2 \times A_j) \in \mathrm{DP}'(G \,\square\, H)$. Applying this argument inductively we get $(\{v_i\}_{i=1}^{s-1} \times H) \cup (v_s \times A_j) \in \mathrm{DP}'(G \,\square\, H)$, for all $1 \leq s \leq |G|$ and $1 \leq j \leq |H|$. Therefore, $[0, |G| \times |H|] \subseteq \mathrm{dp}'(G \,\square\, H)$, so $G \,\square\, H$ is dp. □

The relationship between Cartesian product and the dp property seems more delicate. In particular, we note that $G \,\square H$ can be dp even though G or H may not be. As an example suppose a graph G consists of the cycle C_7 with a pendant edge and H is the path P_2. It is easy to see that G does not have any isometric subgraph of order 5, and using Lemma 5.2.3 one can prove that $G \,\square\, H$ is dp. Computations suggest the following conjecture.

Conjecture 5.2.6. *If G and H are dp then so is $G \,\square\, H$.*

Furthermore let G and H be the shown graphs in the Figures 5.1 and 5.2 respectively, since $2 \notin \mathrm{dp}'(G)$ and $3 \notin \mathrm{dp}'(H)$ we have that G and H are not dp, while using the Lemma 5.2.3 and transitivity of isometric property we will see $G \,\square\, H$ is dp.

5.3 A Benchmark graph

In this section we investigate a sparse dp graph of order 80 whose factors are non-dp. We used a brute force algorithm and found all non-dp, unlabeled graphs of order 1 up to the order 9. The results are partially listed in the following table.

Table 1 presents number of non-dp graphs of oreder 5 up to 9. Column one shows the order being considered. Column two gives the number of non-dp graphs of each order.

Table 1	
Graph order	#of non-dp graphs
5	1
6	1
7	4
8	19
9	183

Table 5.1 The number of non-dp graphs of certain orders

Proposition 5.3.1. *The C_5 and C_6 are the only non-dp graphs of order less than 7.*

Proof. By table 1 we have only two non-dp graphs of order 5 and 6. Using Theorem 4.1.1, they are C_5 and C_6. □

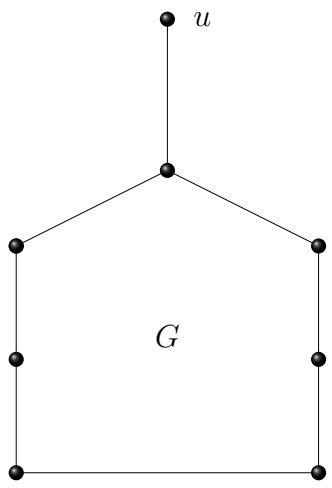


Figure 5.1 A non-dp graph G with order 8 such that $1 \in \mathrm{dp}'(G)$ and $2 \notin \mathrm{dp}'(G)$.

Theorem 5.3.2. *Let G and H be the graphs in Figure 5.1 and 5.2 respectively. Then $G \,\square\, H$ is dp.*

Proof. The result has been verified by computer. □

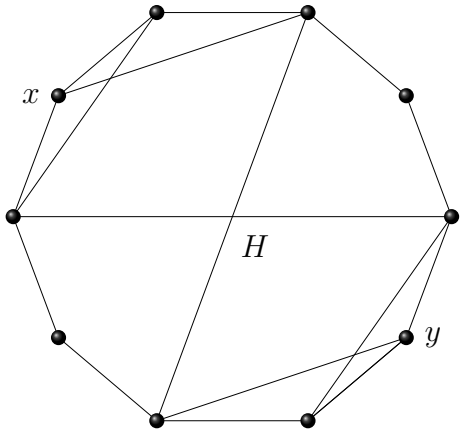


Figure 5.2 A non-dp graph H of order 10 such that $1 \in \mathrm{dp}'(H)$ and $2 \in \mathrm{dp}'(H)$. It is easy to see that $\{x\} \in \mathrm{DP}'(H)$ and $\{x, y\} \in \mathrm{DP}'(H)$, but $3 \notin \mathrm{dp}'(H)$.

5.4 Chapter summary

In this chapter, We gave a necessary and sufficient condition for the lexicographic product of two graphs to be dp. This condition implies that if G is dp then the lexicographic product of G and any graph H is dp. Moreover, all isometric subgraphs of the lexicographic product of two arbitrary graphs were characterized. We also showed that the Cartesian product of two graphs is sdp if and only if its factors are, further the Cartesian product of sdp and dp graphs are dp. In closing, we investigated a distance preserving graph whose product factors are not distance preserving. This graph has 80 vertices, and can be used as benchmark for algorithms in this concept.

Chapter 6

Distance-preserving graphs and modular decomposition

Pure mathematics is, in its way, the poetry of logical ideas.
-Albert Einstein

Many problems in graph theory can be tackled by decomposing a graph into smaller pieces and then studying the problem on these parts individually. There are many different ways to decompose a graph that have been applied to a variety of problems. In this chapter we use modular decompositions of graphs to study the distance preserving property. Modular decomposition has been used to solve many problems, see [10, 22, 23, 29].

6.1 Introduction

The *lexicographic product* $G[H]$ replaces every vertex of the graph G with the graph H. We introduce the *generalized lexicographic product* $G[\mathcal{H}]$ which replaces each vertex v of the graph G with a graph $H_v \in \mathcal{H}$, where $\mathcal{H}$ is a set of graphs indexed by the vertices of G, see Figure 6.1. This can be viewed as a generalization of the traditional lexicographic product because setting $H_v = H$, for all vertices v of G results in the lexicographic product $G[H]$. Moreover, we see that any graph M can be represented using the generalized lexicographic product, that is, M is isomorphic to $G[\mathcal{H}]$ for some G and $\mathcal{H}$.

The generalized lexicographic product has appeared in various forms in the literature. This operation is equivalent to applying a substitution, as first defined in [6], to every vertex in the graph. One example of the implicit use of the generalized lexicographic product is Lovász's proof of the perfect graph theorem [21] which uses the multiplication of vertices of a graph G, which is equivalent to the generalized lexicographic product $G[\mathcal{H}]$ with $H_v = \overline{K}_{h_v}$ for every vertex v of $V(G)$, where $h_v \geq 1$ and $\overline{K}_{h_v}$ is the empty graph with h_v vertices.

A *module* in a graph M is an induced subgraph H whose vertices share the same neighbourhood outside of H. A *modular decomposition* of a graph M is a collection of modules of M, where every vertex of M appears in exactly one module. The neighborhood condition forces empty or complete bipartite graphs between modules. There are various polynomial time algorithms for computing the modular decomposition of a graph, see [13]. Given a modular decomposition $\mathcal{H}$ of M we define the *quotient graph* of M with respect to $\mathcal{H}$, denoted $M/\mathcal{H}$, as the graph obtained by replacing each module of $\mathcal{H}$ with a single vertex, where there is an edge between two vertices of $M/\mathcal{H}$ if and only if there are edges between the vertices of the corresponding modules in M. The generalized lexicographic product can be consider as the inverse of the modular decomposition operation, thus M is isomorphic to $(M/\mathcal{H})[\mathcal{H}]$.

A *split decomposition* of a graph is a modular decomposition into two modules both with order greater than 1. Split decompositions have been used to study distance hereditary graphs, that is, graphs in which every induced subgraph is isometric. It was shown in [1] that the distance hereditary property is equivalent to a graph being totally decomposable using split decompositions. See [11] for a definition of totally decomposable and a general overview of split decompositions.

6.2 Generalized lexicographic product

In this section we define two graph operations, the generalized lexicographic product and modular decomposition more precisely. We note that these two operations are the inverses of each other. First we introduce the generalized lexicographic product.

Definition 6.2.1. *Let G be a graph and $\mathcal{H} = \{H_v\}_{v \in V(G)}$ be a set of graphs. Define the* generalized lexicographic product $G[\mathcal{H}]$ *as the graph with vertex and edge sets*

$$V(G[\mathcal{H}]) = \bigcup_{v \in V(G)} (\{v\} \times V(H_v)),$$

$$E(G[\mathcal{H}]) = \{(u,x)(v,y) \mid uv \in E(G)\} \cup \bigcup_{v \in V(G)} \{(v,x)(v,y) \mid xy \in E(H_v)\}.$$

In other words $G[\mathcal{H}]$ is constructed by replacing every vertex $v \in V(G)$ with the graph H_v, and the edges between H_u and H_v form a complete bipartite graph or the empty graph depending on whether $uv \in E(G)$ or $uv \notin E(G)$, respectively. To clarify the notation Figure 6.1 is given as an example.

Note that if $H_v = H$, for all $v \in V(G)$, then $G[\mathcal{H}]$ is the lexicographic product graph $G[H]$.

The inverse of this operation has been well studied and is known as the modular decomposition of a graph, see [13] for an overview. The *neighbourhood* of a vertex $v \in V(G)$, denoted $N_G(v)$, is the set of all vertices in G joined by an edge to v. Moreover, given a subgraph A of G let $N_G(A) = \cup_{v \in V(A)} N_G(v) \setminus V(A)$.

Definition 6.2.2. *Let H be a subgraph of a graph M. We call H a* module *of M if $N_M(u) \setminus V(H) = N_M(H)$, for all $u \in V(H)$. The module H is* maximal *if there is no module H' of M such that $H \subsetneqq H' \subsetneqq M$. A module of M is* trivial *if it is a single vertex or the whole graph. A* modular partition $\mathcal{H}$ *of M is a set of disjoint modules of M such that*

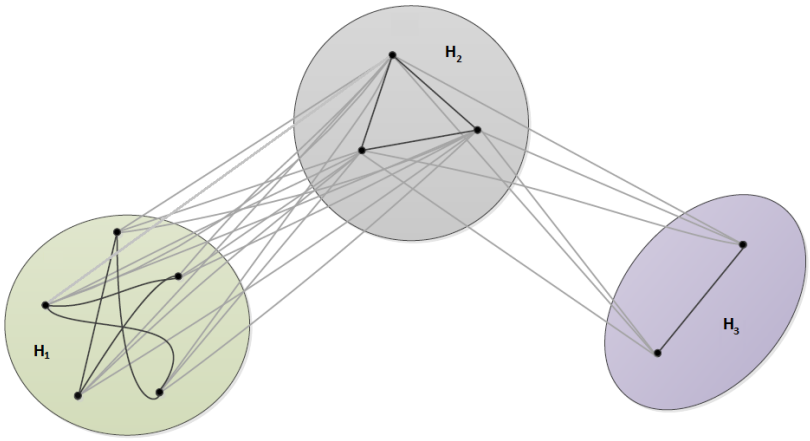


Figure 6.1 A graph $G[\{H_{a_1}, H_{a_2}, H_{a_3}\}]$, where G is the 2-path $a_1a_2a_3$ and $H_{a_1} = C_5$, $H_{a_2} = K_3$ and $H_{a_3} = K_2$

$V(M) = \bigcup_{H\in\mathcal{H}} V(H)$. *Two modules H and H' of a partition are said to be* adjacent *if $(u, v) \in E(M)$ for every $(u, v) \in V(H) \times V(H')$. A trivial or maximal decomposition of a graph is the modular decomposition where every module is trivial or maximal, respectively.*

For example, $H_{a_1} \cup H_{a_3}$ and H_{a_2} are some of the modules in Figure 6.1. Moreover, deleting any vertex from H_{a_2} gives a maximal module and the modules H_{a_1} and H_{a_2} are adjacent, but the modules H_{a_1} and H_{a_3} are not adjacent.

Definition 6.2.3. *Let M be a graph with a modular partition $\mathcal{H}$. The* quotient graph $M/\mathcal{H}$ *is the graph with a single vertex v_H for each $H \in \mathcal{H}$ and an edge between v_H and $v_{H'}$ if and only if H and H' are adjacent in M. We say that $M/\mathcal{H}$ is a* minimal quotient graph *of M if $M/\mathcal{H}$ contains no non-trivial modules.*

Note that the quotient operation and generalized lexicographic product are inverses of each other up to isomorphism, that is, $M \cong (M/\mathcal{H})[\mathcal{H}]$ and $G \cong (G[\mathcal{H}])/\mathcal{H}$, where $\cong$ denotes that two graphs are isomorphic. We say that a graph M can be *represented* by a graph G and set $\mathcal{H}$ if $M \cong G[\mathcal{H}]$. For example in Figure 6.1, the graph can be represented as $K_2[\{H_{a_1} \cup H_{a_3}, H_{a_2}\}]$.

Next we present a useful lemma on the union of modules and then present the main result of this section.

Lemma 6.2.4. *If H and K are both modules of M with $V(H) \cap V(K) \neq \emptyset$, then $H \cup K$ is also a module of M.*

Proof. Any vertex $b \in M - (H \cup K)$ is either a neighbour of all or none of $V(H)$. If b is a neighbour of all of $V(H)$, then it is a neighbour of all of $V(H) \cap V(K)$, so it is a neighbour of all of $V(K)$. Similarly if b is a neighbour of none of $V(H)$, then it is a neighbour of none of $V(K)$. Therefore, every element of $V(H) \cup V(K)$ has the same neighbours in $M - (H \cup K)$. □

Theorem 6.2.5. *Consider a graph G with at least three vertices. The graph G is a minimal quotient graph of $M = G[\mathcal{H}]$ if and only if $\mathcal{H}$ is a maximal modular decomposition.*

Proof. By definition a graph G is a non-minimal quotient graph of M if and only if G contains a non-trivial module K. We first prove the backwards implication of the theorem by contradiction. If G contains a non-trivial module K, then H_v is a non-maximal module in M for any $v \in K$. We also prove the forwards direction in the contrapositive form. Suppose H_v is a non-maximal module in M, so there is a maximal module $H' \supset H_v$. Furthermore, there is some other module H_u with $H' \cap H_u \neq \emptyset$, so $H' \cup H_u$ is also a module by Lemma 6.2.4. Since H' is maximal, either $H_u \subset H'$ or $H' \cup H_u = M$. If the former case is true for all possible u then H' is a union of two or more modules of M and so can be used to get a smaller quotient than G. So consider the case $H' \cup H_u = M$. If H_u is the only other module in $\mathcal{H}$ then G has only two vertices which is a contradiction. If there are $k > 2$ modules then $\mathcal{H}$ contains $k - 1 > 1$ modules and as these modules are all contained in one larger module the corresponding vertices in G must form a module in G, so G is not minimal. □

In the proof of Theorem 6.2.5 the requirement that G has at least three vertices is only needed for the backwards direction, so we get the following corollary:

Corollary 6.2.6. *If $\mathcal{H}$ is a maximal modular decomposition of M, then $M/\mathcal{H}$ is a minimal quotient graph.*

However it is necessary that the graph has at least three vertices for the forwards direction. To see this consider K_4 and the modular decomposition $\mathcal{H}$ partitioning K_4 into two modules of three vertices and one vertex. This is not a maximal modular partition but $K_4/\mathcal{H}$ equals K_2 which is the minimal quotient graph of K_4.

Theorem 6.2.5 is similar to Theorem 2 in [13], but gives an equivalence statement rather than just a necessary condition. Moreover, the condition in Theorem 2 of [13] states that the graph $M = G[\mathcal{H}]$ must have a connected complement graph. The complement graph $\bar{M}$ of M is the graph with the same vertices as M and xy is an edge in $\bar{M}$ if and only if xy is not an edge in M. In fact the condition that $M = G[\mathcal{H}]$ must have a connected complement is equivalent to our condition that $|G| \geq 3$, as can be seen by the following result:

Lemma 6.2.7. *The graph M has a disconnected complement graph if and only if K_2 is a quotient graph of M.*

Proof. The graph $\bar{M}$ is disconnected with components A and B if and only if there is no edge in $\bar{M}$ between any vertices $a \in V(A)$ and $b \in V(B)$, which is equivalent to the graphs induced by $V(A)$ and $V(B)$ in M being a modular decomposition $\mathcal{H}$ with $M/\mathcal{H} = K_2$. □

We can also determine when a graph has a unique maximal modular decomposition and unique minimal quotient graph.

Lemma 6.2.8. *If K_2 is not a quotient graph of M, then M has a unique maximal modular decomposition.*

Proof. Suppose that M has two different maximal modular decompositions $\mathcal{H}$ and $\mathcal{H}'$. There must exist a pair $H \in \mathcal{H}$ and $H' \in \mathcal{H}'$ with $H \cap H' \neq \emptyset$ and $H \neq H'$. As both H and H' are modules $H \cup H'$ is also a module by Lemma 6.2.4. So the only way that H

and H' are maximal is if $H \cup H' = M$. However $M - H$ is also a module. To see this first note that every element of H' has the same neighbours in H, because there is at least one element $h \in H$ with $h \notin H'$, so h is either a neighbour of all or none of H'. Moreover, since every element of H has the same neighbours either all elements of H are neighbours of all elements of H' or none are. Furthermore, $M - H \subseteq H'$ so every element of $M - H$ has the same neighbours in H, thus $M - H$ is a module. Therefore, H and $M - H$ form a modular decomposition of M, so K_2 is a quotient graph of M. □

Corollary 6.2.9. *Every graph M has a unique minimal quotient graph.*

Proof. If K_2 is a quotient graph of M, then K_2 is the unique minimal quotient graph. Otherwise, M has a unique maximal modular decomposition $\mathcal{H}$ by Lemma 6.2.8, so $M/\mathcal{H}$ is the unique minimal quotient graph. □

Note that if a graph G has a modular decomposition with the quotient graph K_2, where both modules are non-trivial, then this is a split decomposition.

6.3 Distance preserving characterization

In this section we investigate some conditions under which $G[\mathcal{H}]$ is distance preserving. We begin by considering the relationship between the geodesic paths in G and the geodesic paths in $G[\mathcal{H}]$. The following lemmas and corollary are easy modifications of parallel results from the lemma 5.1.1, hence proofs are omitted.

Lemma 6.3.1. *Consider a path $g_1, g_2, \ldots, g_\ell$ in G. A path $(g_1, h_1), (g_2, h_2), \ldots, (g_\ell, h_\ell)$ is (g_1, h_1)–(g_ℓ, h_ℓ) geodesic in $G[\mathcal{H}]$ if and only if $g_1, g_2, \ldots, g_\ell$ is g_1–g_ℓ geodesic in G.*

Lemma 6.3.2. *Consider a connected graph G with $|G| \geq 2$ and a set of graphs $\mathcal{H} = \{H_v\}_{v \in V(G)}$.*

(a) If $x \in V(H_u)$ and $y \in V(H_v)$ are distinct vertices, with $u, v \in V(G)$, then:

$$d_{G[\mathcal{H}]}((u,x),(v,y)) = \begin{cases} d_G(u,v), & \text{if } u \neq v, \\ 2, & \text{if } u = v \text{ and } xy \notin E(H_u), \\ 1, & \text{otherwise.} \end{cases}$$

(b) If u, v are distinct vertices of G, then $d_G(u,v) = d_{G[\mathcal{H}]}((u,x),(v,y))$, for any $x \in H_u$ and $y \in H_v$.

Corollary 6.3.3. *The graph $G[\mathcal{H}]$ is connected if and only if G is connected.*

In the remainder of this section we generalize some more results of Section 5.1. In order to state the main theorem of this section we need some notation. Let

$$\text{ndp}(G) = \{k \mid \text{G has no isometric subgraph with k vertices}\},$$

so a graph G is dp if and only if $\text{ndp}(G) = \emptyset$. If a and b are integers with $a < b$, then let $[a,b] = \{a, a+1, a+2, \ldots, b\}$. Given a subgraph M of $G[\mathcal{H}]$, let $\pi(M)$ be the induced subgraph of G with the vertex set:

$$V(\pi(M)) = \{a \in V(G) \mid (a,x) \in M \text{ for some } x \in H_a\}.$$

Theorem 6.3.4. *Let G be a connected graph with $|G| \geq 2$. Any generalized lexicographic product graph $G[\mathcal{H}]$ is dp if and only if for any $k \in \text{ndp}(G)$, there is a subgraph $L \leq G$ with $|L| < k \leq \sum_{u \in L} |H_u|$.*

Proof. We claim that for an induced subgraph M of $G[\mathcal{H}]$, with $\pi(M)$ having at least two vertices,

$$M \leq G[\mathcal{H}] \text{ if and only if } \pi(M) \leq G. \tag{6.1}$$

The proof is easy modification of parallel proof from the claim 5.1 in the Theorem 5.1.2.

Now we prove the theorem. By the definition of the quotient graph we know that $|\pi(M)| \leq |M| \leq \sum_{u \in \pi(M)} |H_u|$. Statement (6.1) implies that $G[\mathcal{H}]$ is dp if and only if

$$\bigcup_{L \leq G} \big[|L|, \sum_{u \in L} |H_u|\big] = \big[1, \sum_{u \in G} |H_u|\big]. \tag{6.2}$$

Since $1, 2, |G|$ are never in $\mathrm{ndp}(G)$, Equality (6.2) is equivalent to: for any $k \in \mathrm{ndp}(G)$, there is an $L \leq G$ with $|L| < k \leq \sum_{u \in L} |H_u|$. □

Theorem 6.3.4 generalizes Theorem 3.2 in [19].

Corollary 6.3.5. *[19, Theorem 3.2] Let G be a connected graph with $|G| \geq 2$ and H be an arbitrary graph with $|H| = n$. Then $G[H]$ is dp if and only if $b \leq an + 1$ for every pair $a, b \in \mathrm{ndp}(G)$ bounding a non-dp interval.*

Proof. When $H_u = H$ for all vertices u in G, Equality (6.2) in the proof of Theorem 6.3.4 is equivalent to $[a, an] \cup [b, bn]$ being an interval for every pair a, b bounding a non-dp interval, which is equivalent to $b \leq an + 1$. □

The next result is an immediate corollary of Theorem 6.3.4.

Corollary 6.3.6. *If G is dp, with $|G| \geq 2$, then $G[\mathcal{H}]$ is dp for any set of graphs $\mathcal{H}$.*

Since any tree is dp, Corollary 6.3.6 implies that the graph in Figure 6.1 is dp. The graph $G[\mathcal{H}]$ being dp does not necessarily imply that G is dp. This can be seen in Figure 6.2 which shows the graph $C_5[\mathcal{H}]$, where C_5 is the 5-cycle and $\mathcal{H}$ substitutes K_2 for

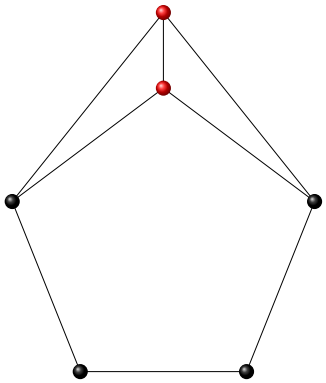


Figure 6.2 The graph $C_5[\mathcal{H}]$, where $\mathcal{H}$ substitutes K_2 for one vertex and K_1 for all others one vertex and K_1 for all others. It is straightforward to verify that $C_5[\mathcal{H}]$ is dp, however C_5 is non-dp.

The next result follows easily from Lemma 6.3.2 and Statement (6.1) in the proof of Theorem 6.3.4.

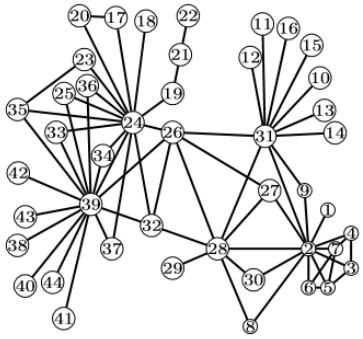
Corollary 6.3.7. *For a connected graph G with $|G| \geq 2$ and an induced subgraph M of $G[\mathcal{H}]$,*

$$M \leq G[\mathcal{H}] \text{ if and only if } \begin{cases} \pi(M) \leq G & \text{when } |\pi(M)| \geq 2, \\ \operatorname{diam}(M) \leq 2 & \text{when } |\pi(M)| = 1. \end{cases}$$

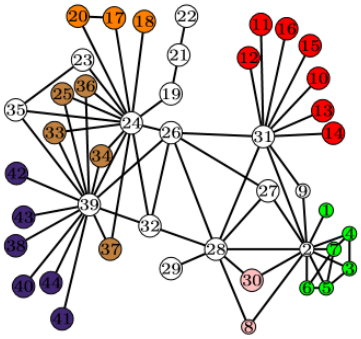
Corollary 6.3.7 implies the following result on sdp graphs:

Corollary 6.3.8. *If G is sdp, with $|G| \geq 2$, then $G[\mathcal{H}]$ is sdp for any set of graphs $\mathcal{H}$.*

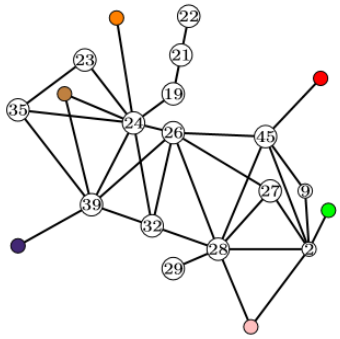
An illustrative example is shown in Figure 6.3. This figure depicts a graph M formed of a social network of friendships between 44 members of a community of international students, along with a modular decomposition of M and a minimal quotient graph of M. By the results of Section 6.3, to show that M is distance preserving it is sufficient to show that the quotient graph is distance preserving. Note that the quotient graph does not contain any induced cycles of length greater than 4, so the quotient graph is dp by Theorem 3.5 of [30]. Therefore, Corollary 6.3.6 implies that M is distance preserving.



(a) A Graph M



(b) The Maximal Modular Decomposition of M



(c) The Minimal Quotient Graph of M

Figure 6.3 An illustrative example for a distance preserving social network.

6.4 Chapter summary

In this chapter, We formally defined the generalized lexicographic product and modular decomposition. We presented a result that a quotient of a graph is minimal if and only if its corresponding modules are maximal, provided the quotient has at least three vertices. This strengthens some of the existing results in this area, see [13].

This generalization is the inverse of the modular decomposition of graphs, which divides the graph into disjoint clusters called modules. Using these operations, we gave a necessary and sufficient condition for graphs to be dp. This condition implies that if G is dp then $G[\mathcal{H}]$ is dp. Moreover, all isometric subgraphs of $G[\mathcal{H}]$ were characterized.

Chapter 7

Distance-preserving graphs and evolutionary algorithms

It is time to create new social science departments that reflect the breadth and complexity of the problems we face as well as the novelty of 21st-century science. These would include departments of biosocial science, network science, neuroeconomics, behavioral genetics and computational social science.

-Nicholas A. Christakis

Finding isometric subgraphs is integral to recognizing dp graphs, but may not be useful to find an efficient algorithm for this recognition. Our computational techniques used brute-forced methods for this recognition, which was slow even for very small graphs.

The first proposed algorithm presented in this chapter is an evolutionary algorithm to check the dp property of a graph.

7.1 Biologically inspired algorithms

Biologically inspired algorithms are part of a branch of computer science that generally uses and applies subfields related to the topics of connectionism, social behaviour and emergence to solve computationally challenging problems that usually do no have closed form optimal solution. On one hand, such algorithms massively depend on the fields of biology, computer science and mathematics since they apply ideas and models that have

been inspired by examining living organisms to solve unsolved sophisticated problems (usually NP hard) in computer science. On the other hand, such algorithms can help scientist to have a better understanding of complicated behavior of living organisms by proposing simple computational models (in comparison to complex biological dynamics) and test the validity of proposed hypotheses on fast computational models.

From their earliest days, bio-inspired algorithms were not only applied to calculating missile trajectories and deciphering military codes but also to modeling the brain functionality, imitating complex behavior of social animals, trying to model learning skill in human being, and simulating biological evolution. Bio-inspired algorithms have their ups and down over the years, but since the early 1980s they have all undergone a resurgence in the computation research community because of rapid increase in computation speed. After this re-birth, bio-inspired algorithms were divided into three branches. The first field is artificial neural networks (cutting-edge deep learning methodology is the off-spring of this sub-family), the second field was machine learning, and the third branch nowadays is called "evolutionary computation," of which genetic algorithms are the most prominent member of this category.

Genetic Algorithms (GAs) were first introduced by Prof. John Holland in the late 60s and later developed by Holland and his students and colleagues at the University of Michigan in early 70s. In comparison to other evolutionary computation frameworks and programming, Holland's primary goal was to propose a general purpose algorithms to solve not just a specific problems, but rather by formally studying the process of evolution, natural selection, and adaptation as it occurs in biological populations to develop ways in which the mechanisms of natural adaptation might be imported into computer systems. His 1975 book "Adaptation in Natural and Artificial Systems" demonstrated this algorithm as an abstraction of biological evolution and gave a theoretical framework for natural selection, evolutionary operators, and adaptation under the GA. Holland's GA is a framework for generating better solutions from ancestral populations, which are represented by strings of

"bits" as "chromosomes" by applying an evolutionary force similar to "natural selection" together with the geneticsinspired operators of crossover, mutation, and inversion. The selection operator gives higher chance of reproduction to those chromosomes in the ancestral population. Crossover operator combines subparts of two selected chromosomes, roughly mimicking biological recombination between two singlechromosome or "haploid" organisms; mutation operator randomly changes the allele binary values of some locations in the chromosome; and finally inversion reverses the order of a contiguous section of the chromosome, thus rearranging the order in which genes are arrayed. Since then, Holland's GA and other evolutionary computation frameworks have been successfully applied to different NP hard problems and have had superb performance.

In this research, we introduce an evolutionary algorithm that generates population of graphs in which the distance property is preserved. Here, the distance property of each graph G with vertex set $V(G)$ and edge set $E(G)$ is analyzed to decide which graph to be used as a member of ancestral population to generate distance preserved descendants over the course of evolution. In this algorithm, a mating event is performed as follows: Pick a vertex v of the graph uniformly at random. A neighbor of v is chosen for mating with a fitness bias. Crossover and probabilistic mutation are used to produce a single new individual which may or may not be used to replace the individual on vertex v. The details of how the neighbor is picked for mating and how to decide if the new individual replaces the individual on vertex v are called the local mating rule of the graph based genetic algorithm [5].

7.2 Methodology

7.2.1 Problem Definition

Problem 7.2.1. *Let G be a connected graph. Does G contain an isometric subgraph of order k, for some $k \leq |G|$?*

Answering this question helps us to determine whether G is a dp graph or not by considering all k for $k = 1, 2, \ldots, n$. Note that when G is a tree, every connected induced subgraph is obtained by subsequently removing some leaves; thus, the problem is challenging when the graph contains cycles. In other words when there are two different paths between a pair of vertices in a graph then the graph contains a cycle and removing a node from the shorter path may result in a non-isometric subgraph.

7.2.2 Algorithm Outline

In this section, an evolutionary search algorithm for solving distance-preserving graph problem is proposed. The flowchart of the proposed algorithm is demonstrated in Figure 7.1 in which each block will be illustrated in the following subsections. The population size N is a user defined parameter.

Generating the initial population

Every connected induced subgraph with diameter 2 is an isometric subgraph; hence the evolutionary algorithm begins it's search for isometric subgraphs of order greater than or equal to three. For the initial population, we can randomly choose N connected subgraphs of all $\binom{|G|}{3}$ induced subgraphs with order three of G. But choosing these N subgraphs for the initial population needs to know all subgraphs of G with order three that is computationally expensive when the order of the graph is large. One way to remedy this issue would be randomly generating N connected subgraphs of G with order three in which, we randomly pick a vertex, a, and then randomly select a neighbor of a, namely b, and then choose a random vertex c adjacent to $\{a, b\}$ in G. We consider the induced subgraph on vertex set $\{a, b, c\}$ as an element of the initial population. In fact, the randomness of this way is the same, but the time complexity is much less.

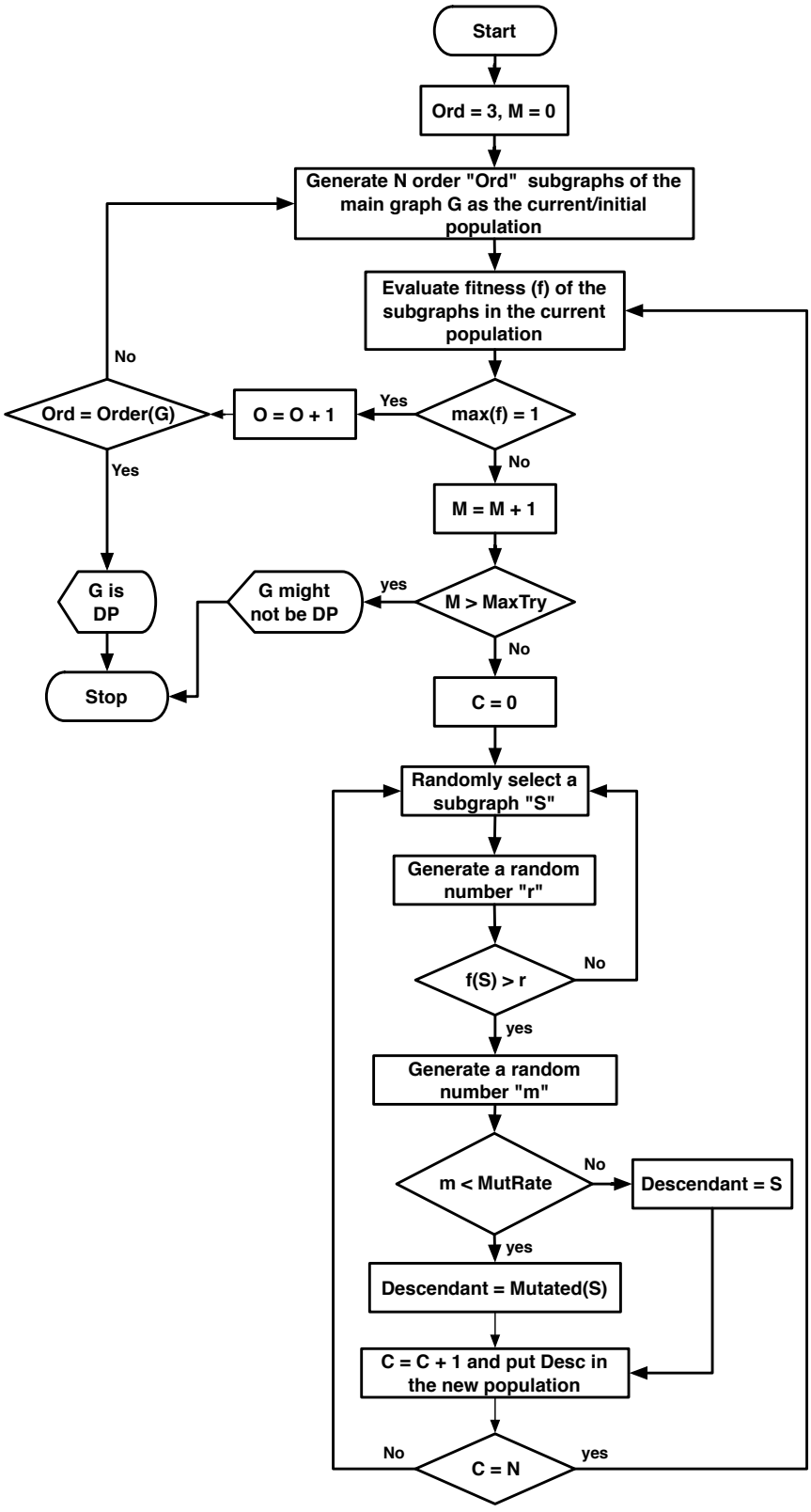


Figure 7.1 The proposed genetic-based search algorithm for distance preserving graphs.

Fitness function

Distance similarity between induced subgraphs and main graph is used to define the fitness function of the evolutionary algorithm. Among all sugraphs of a graph G, the maximally distance preserved subgraphs have the highest fitness, and consequently these subgraphs have higher chances to get selected and reproduce for the next generation. By above aforementioned criterion, we define the fitness function as follows:

$$f(H) = 1 - \left| \frac{\sum_{x,y \in V(H)} d_H(x,y) - \sum_{x,y \in V(H)} d_G(x,y)}{\binom{H}{2} \times \mathrm{diam}(H)} \right|,$$

where the first summand explains the summation of all distances in H and the second one is the summation of all distances between all pair of vertices of H in the original graph G. Clearly if H is a subgraph of G then

$$d_H(x,y) \geq d_G(x,y) \geq 0$$

for every pair of vertices $x, y \in V(H)$. Therefore, if H is isometric in G then $d_H(x,y) = d_G(x,y)$ that gives:

$$f(H) = 1.$$

In general, a subgraph H maximally preserves the distances in G if the phrase inside the absolute value of $f(H)$ is minimal. Having said that, two summations are close to each other and $f(H)$ is closed to 1. Moreover, the denominator of the fraction normalized inside of the absolute value makes the fraction to be a number between zero and one. Note that almost all graphs have diameter 2 [24]. And also using results in [2], one can find a diameter in random graphs according to the chosen model of randomness. Having a fix number for diameter reduces the time complexity of the algorithm. Generally speaking, by proposing this fitness function, we cover all real world networks.

Selection operator

Pick an arbitrary graph H from the population and an arbitrary real number, r, within the interval $[0, 1]$. If $f(H) \geq r$ then H is selected for the next generation. By using this operator, those subgraphs whose fitness values are high have more chance to get selected unlike those subgraphs whose fitness values are low.

Mutation

Suppose an element of the population, more precisely a subgraph H of G, is selected for the next generation. We mutate the subgraph in the following order of steps. Pick a number k uniformly at random in the interval [0,1]. Let the mutation rate be μ, if $k \leq \mu$ then remove a vertex v from H and randomly choose a neighbor u of $H - \{v\}$ in G, $u \in \mathcal{N}(H-\{v\})$, the vertex u is then added to $H-\{v\}$ to create a subgraph $\big(H-\{v\}\big)\cup\{u\}$ for the next generation and if $k > \mu$, the subgraph will not be mutated. The mutation operator is operational when all subgraphs in the population have fitness smaller than one. That means in the population there is no isometric subgraph and the algorithm could not find any isometric subgraph of order "O". Therefore, by using mutation operator, we give this chance to the subgraphs with high fitness value to evolve in the direction of isometric subgraphs. However, in distance-preserving graphs, existence of at least one isometric subgraph for each possible order is a necessary and sufficient condition.

Increasing order criterion

If there is a subgraph with fitness 1 in the population, the search algorithm has successfully found the isometric subgraph with order "O". Now, we can increase the order to "$O + 1$" and repeat the searching procedure for this order. To do so, we use the same selection operator to generate the initial population for order "$O + 1$". Moreover, we use a similar mutation operator to increase the order of all selected subgraphs. Let H be one of the selected subgraphs. To create a new graph using H, we randomly choose a neighbor

w of $\mathcal{N}(H)$ in G and add w to the graph H. Now $H \cup w$ is the desired subgraph. We use $H \cup w$ as an element of the new population in which all subgraphs have order $|H| + 1$.

As we discussed earlier, maximally distance preserved subgraphs are selected to reproduce next generation. The reason of this selection is that if an induced subgraph H of a graph G is not isometric, then there is a pair of vertices u, v in H such that $d_H(u, v) > d_G(u, v)$. If we randomly select a vertex w in $\mathcal{N}(H)$ then w less likely have this chance to be on a u-v geodesic in G.

Termination criterion

If an element of the population has order equal to $|G|$, or the population evolves for more than a user defined threshold of maximum generation, then the algorithm terminates and G is dp. If number of generation for one order exceeds the generation threshold (which is a user defined parameter), G might not be dp.

Algorithm 1 pseudocode of the evolutionary based search algorithm

Procedure:
Step 0. Set order "O" of subgraphs to 3.
Step 1. Generate N subgraphs of order "O".
Step 2. At generation "k" do the following:

a. Evaluate the fitness of subgraphs.
b. **If** the maximum fitness is 1.
If "O" equals to the order of the main graph G:
G is dp (Stop)
Else:
Increase "O" by 1 and go to step 1.

Else:
Increase k by 1.
If k exceeds the "*maximum generation*":
G might not be dp (stop).
Else:
Generate k generation by using the recombination and mutation operators.
The fittest subgraph should have higher chance to reproduce.
Go to step *a*.

The pseudo-code of this method is given in Algorithm 1, and the details of this algorithm are presented in the Figure 7.1.

7.3 Experimental results

Here we present the results of our algorithm on graphs randomly generated of order 10 to 100. And also for each given order, we generated 10 graphs with different edge probabilities. The population size varies from 10 to 400. Experiments run under the Linux operating system on an Intel 10 1536 cluster of 192 dual core Intel 2200+t processors (High Performance Computing Laboratory, Michigan State University http:// hpcc.msu.edu). The mutation rates for our experiments would be $.001, .002, .005, .01$ and $.05$. We have 100 replicates for each experimental set up. Since there is no search algorithm other than brute force in the literature; therefore, we compare the performance of our algorithm with brute force algorithm.

7.3.1 Overall performance of the evolutionary algorithm.

In this section, we demonstrate how many times out of 100 runs, this algorithm terminates correctly as graphs, with different orders, are dp. Here, we fix the population size to 10, mutation rate to 0.01, and finally the probability of existence of an edge in graphs is 0.03. As we can see in Figure 7.2, the evolutionary algorithm terminates more than 95 percent correctly for graphs of order up to 100. Since this search problem is very time consuming the brute force algorithm failed to give a result in a limited time.

Moreover when the order of graphs are increased, the running time would be increased too, see Figure 7.3.

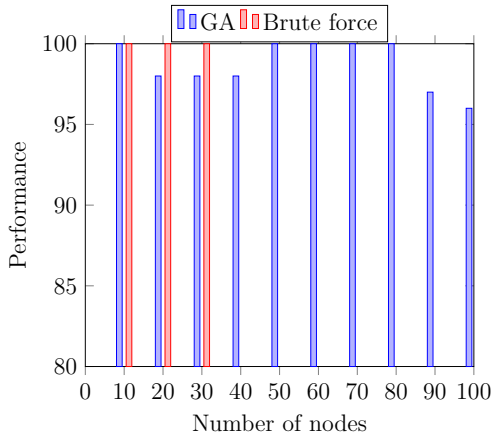


Figure 7.2 Performance of the genetic and brute-force algorithms in finding dp property of randomly generated graphs. Note that the brute-force algorithm does not terminate for graphs of order greater than 30 in our time limit which is 450 seconds.

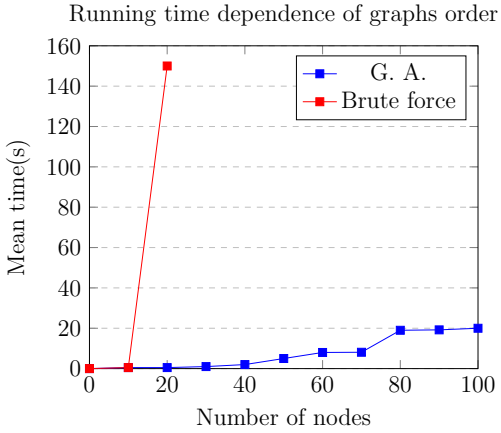


Figure 7.3 Running time of the genetic and brute force algorithms. Brute-force algorithm does not terminate in our time limit which is 450 seconds for graphs of order greater than 30.

7.3.2 Effect of population size.

In this experiment, we check the effect of population size on the performance of the algorithm on a given graph of order 30. Here the mutation rate is fixed to 0.05, and the probability of existence of an edge in the generated graph is 0.4. Our choice of picking 0.4 as the edge probability makes the problem more challenging. The results are shown in Figures 7.4 and 7.5.

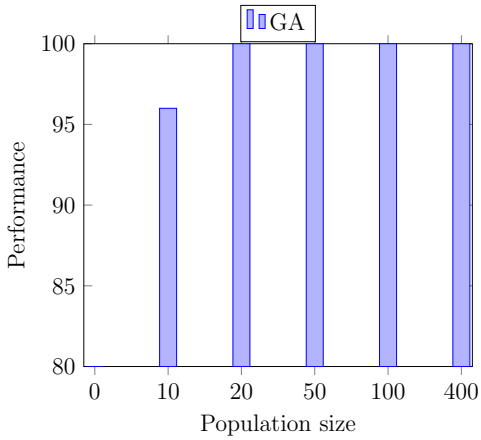


Figure 7.4 Effect of population size on performance of genetic algorithm in finding dp property. Genetic algorithm successfully finds dp graphs when population size is greater than 20.

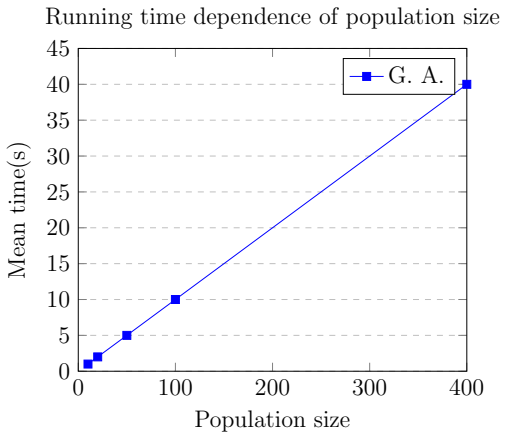


Figure 7.5 Running times of the genetic algorithm for different population sizes.

7.3.3 Effect of mutation rate

In this experiment, we consider the effect of mutation rate on the performance of the algorithm. Here, we fix the graph's order to 90, population size to 10, and the probability of existence of an edge in the graph is 0.4. As we see in Figures 7.6 and 7.7, the optimal value for mutation rate is 0.005.

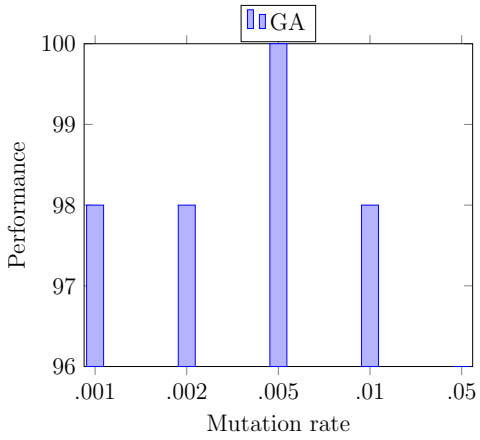


Figure 7.6 Effect of mutation rate on performance of genetic algorithm in finding dp property. This figure demonstrates that the optimal mutation rate is 0.005.

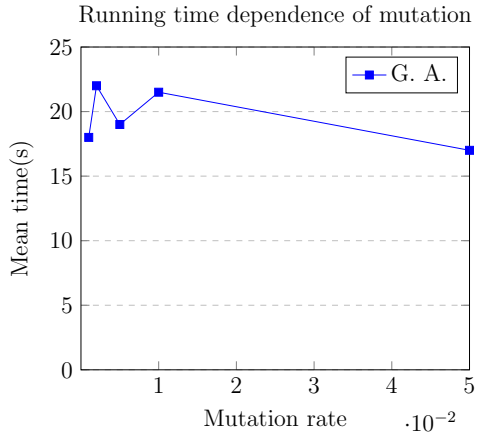


Figure 7.7 Running time of the genetic algorithm for different mutation rates.

7.3.4 Effect of edge probability

In this experiment, we check how the probability of existence of edges effects on the performance and running time of the algorithm. We set the order of graph to 100, population size to 10, and mutation rate to 0.001. As we see in Figure 7.8 when the edge probability goes up the performance is getting better. The selected graphs are dp, hence the brute force algorithm always gives result correctly.

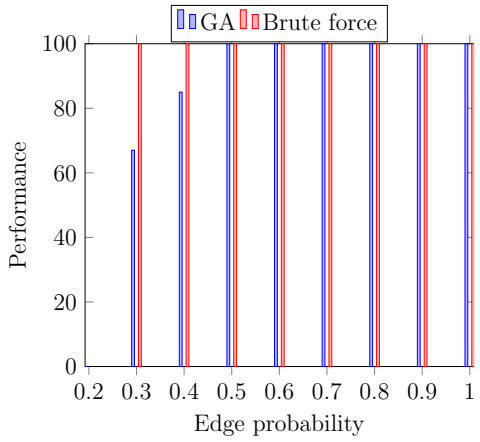


Figure 7.8 Performances of genetic algorithm in comparison with the performance of brute-force algorithm in checking dp property. As the edge probability goes up we have more accurate results for the genetic algorithm.

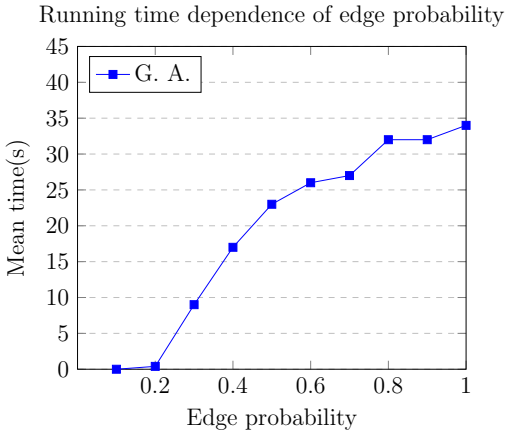


Figure 7.9 Running time of the genetic algorithm for randomly generated graph of order 100 with different edge probabilities.

7.4 Concluding remarks

The problem of finding an isometric subgraph of order k for a given graph G is challenging, even if the graph has an isometric subgraph of order k. We conjecture this problem cannot be solved in polynomial time; thus, in this paper we apply the heuristic algorithms to address this problem. Among these algorithms, we utilized biologically inspired search algorithm to search for the dp property of a given graph. Currently, brute force algorithm

is the only available algorithm for solving this problem. Therefore, brute force algorithm is used to check the performance of the proposed genetic algorithm. While the accuracy of proposed algorithm is more than %96, its running time is reasonably faster than brute force approach. Subgraphs that maximally preserve the distances in the original graphs have higher chances to get selected and reproduce for the next generation. Although there is always a slight chance of failure when the input graph is sparse, by increasing the population size and increasing the mutation rate we can avoid such failures by losing the speed of the proposed algorithm. The presence of certain cycles can cause a graph not to be dp. In the existence of such cycles in the graph, we can attach these cycles to the offspring of selected ancestral parents to have isometric subgraphs containing cycles whose existence makes problem in finding isometric property. This algorithm can be easily extended to determine if a graph is sequentially distance preserving by having a harsh selection operator (r equals to 1).

7.5 Issues and future work

Most of the time the fitness fucntion 7.2.2 gives a chance to graphs to get selected for the next generation. Indeed it rarely happens that a non-isometric subgraph is not chosen for the next generation. Moreover, subgraphs that maximally preserve the distances in the original graphs, have higher chances to get selected and reproduce for the next generation. This does not guarantee that the algorithm terminates correctly when the input graph is sparse.

If a graph G is not prime then we look for factors in a linear time [31], and we use the results in Chapter 5 to see whether the graph is dp. Let the graph G be prime and dense enough, using the genetic algorithm, the dp property of G can be recognized with a high probability as we saw in the section 7.3. So the problem would be challenging when the graph G is sparse and prime, in this case we propose to find a proper multiobjective fitness function for the future work.

7.6 Chapter summary

In this chapter, we proposed a biologically-inspired search algorithm to address the problem of finding an isometric subgraph of any order to determine if a given graph is distance preserving. In this algorithm, by using a well defined fitness function, the selection operator selects the almost isometric subgraphs to generate the offsprings for the next generation. There is also a tradeoff between the population size and searching speed. On one hand, the larger the population size is, the slower the search algorithm would be. On the other hand, the more we increase the population size, the higher is the chance to find an existing isometric subgraph. Experimental results depicted the performance of the proposed algorithm in finding isometric subgraphs even for challenging problems. Interestingly, by these results, one can conjecture that "almost" all graphs are distance preserving.

Chapter 8

Fast algorithm for detecting dp graphs

Make everything as simple as possible, but not simpler.
-Albert Einstein

The Algorithm 1 in Chapter 7 is too expensive for finding isometric subgraphs of a graph with more than a couple thousand nodes, but it has the potential to find isometric subgraphs. In this chapter, we propose an algorithm using the genetic Algorithm 1 in which some local search strategies are amalgamated to improve convergence speed. A selection operator is proposed to prevent premature convergence.

8.1 Methodology

8.1.1 Problem definition

Problem 8.1.1. *Let G be a connected graph. Does G contain an isometric subgraph of order k, for some $k \leq |G|$?*

8.1.2 Algorithm outline

The outline of the algorithm is listed in the following subsections. Suppose a graph G is an input of the algorithm.

Parallel search strategies

As we saw, to check if a graph is sequentially distance preserving using the algorithm 1, we just need to set the value of r to 1 in the selection operator 7.2.2. In other words, all fitness values for all subgraphs in the population should be one, and also we assume the mutation rate is zero. This makes the Algorithm 1 to a random algorithm which searches for N sequentially distance preserving subgraphs in the original graph at a time.

For the input graph G, suppose the Algorithm 1, when we set r to one, stops in order "$O+1$" that is, it could find isometric subgraphs of order 1 up to "O" but not "$O+1$". Let the algorithm found $H_1, H_2, \ldots, H_k$ as the first k smallest subgraphs whose fitness value is not 1, which simply means they are not isometric. We put these subgraphs in a queue while the algorithm is running. Our goal is to find isometric subgraphs of order 1 up to $|G|$, if they exist, or $\mathrm{dp}(G) = [0, |G|]$. Up until now, we found ["O"] $\subseteq \mathrm{dp}(G)$, and the problem is how to find an isometric subgraph of order "$O+1$". To this end, we use the subgraphs in the queue.

Attaching cycles

An induced subgraph is not isometric because of cycles in the original graph. Now in the queue we have k non-isometric subgraphs of order smaller than or equal to "O", so they all share some vertices with some cycles in G which is a problem. So if we attach cycles to these subgraphs, they will be changed to isometric subgraphs. After remedying the cycles

issue, constructed subgraphs in the queue can be considered for the next generation in section 8.1.2.

Since the algorithm stops in the order "O+1", we can add some vertices to subgraphs in the queue to make sure the orders of constructed graphs do not exceed "$O+1$". To see how many vertices is needed to attach cycles, we need the following proposition.

Proposition 8.1.2. *The* $diam(C_k) = \lfloor \frac{k}{2} \rfloor$ *and* $d(v_r, v_s) = \begin{cases} |r-s| & \text{if } |r-s| \leq \lfloor \frac{k}{2} \rfloor, \\ k - |r-s| & \text{if } |r-s| \geq \lceil \frac{k}{2} \rceil. \end{cases}$

The following corollary is an immediate result for the Proposition 8.1.2.

Corollary 8.1.3. *The graph* C_n $(n \geq 5)$ *does not have any isometric subgraphs of order* $\lfloor \frac{n}{2} \rfloor + 2$ *up to* $n-1$.

This implies that each subgraph in the queue shares more than $\lfloor \frac{k}{2} \rfloor + 2$, for some $k \geq 5$, vertices with an induced cycle C_k, in the original graph G. We now investigate how to attach cycles to the subgraphs in the queue by the following concept.

Average Clustering

In hierarchical cluster analysis the average linkage clustering is an agglomerative method to merge two clusters, namely X and Y, in order to minimize linkage function, where the linkage function is computed as the average distance of the elements in the first cluster and the elements in the second cluster. In other words, the average between all pairs of nodes (x, y) for x in the first cluster and y in the second one. Mathematically, linkage function can be described as follows.

$$D(X,Y) = \frac{1}{N_X \times N_Y} \sum_{i=1}^{N_X} \sum_{j=1}^{N_Y} d(x_i, y_j),$$

for $x_i \in X$ and $y_j \in Y$, where $d(x_i, y_j)$ is the distance between x_i and y_j, and N_X, N_Y are the number of elements in clusters X, Y respectively.

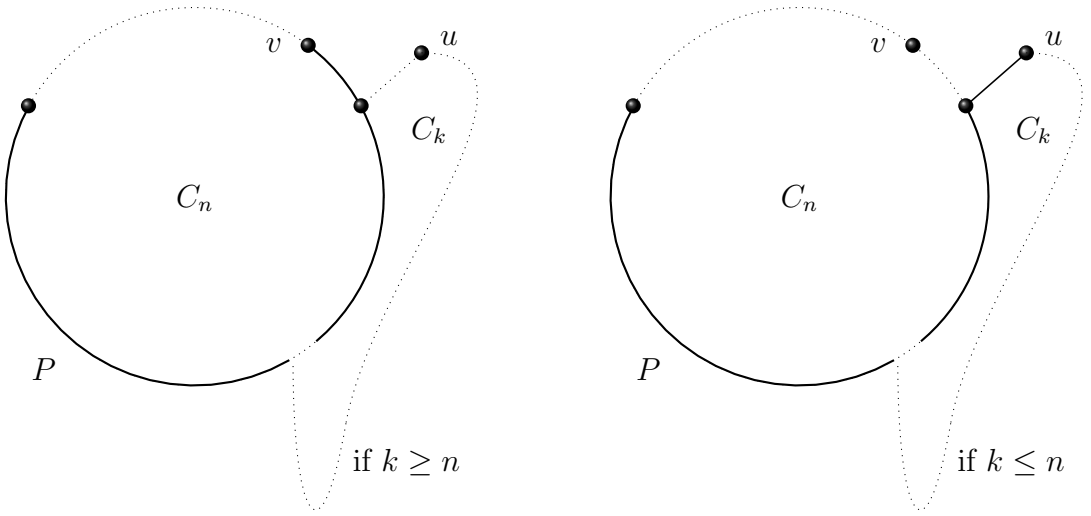


Figure 8.1 The local decision for the average linkage clustering method.

The linkage function has the minimum value one. When two clusters merge together so that it forms a join graph, then the linkage function valuates one. Average linkage uses the average pairwise of nodes in clusters, so clusters tend to be dense and far apart. Typically, based on initial condition each node can belong to different clusters. Two vertices are more likely to join if they have more clusters in common. This implies that if we let a node be a cluster, it will merge to a cluster which has a higher average density of edges, extensively two clusters merge together if they have higher average density.

Let us locally consider the average linkage clustering method by taking a subpath P of an induced cycle C_n in the original graph as a cluster where $|P| > \lfloor \frac{n}{2} \rfloor + 1$, $n \geq 5$, and other clusters all are single vertices of $G - P$. We are interested in joining a single vertex as a cluster and P, using the average linkage method such that the linkage function receives the minimum value. It suffices to consider two cases based on whether the vertex belongs to C_n or not. The desired model is shown in the Figure 8.1.

In the Figure 8.1, using the average linkage clustering method to join P with a single vertex u or v. If $D(P, u) < D(P, v)$, then u is joined to P and the algorithm tends

to attach another induced cycle C_k for some $k < n$ containing u, and at least two vertices of C_n, otherwise it tends to attach C_n.

Therefore we can attach cycles to the subgraphs in the queue by the average linkage method that solves the cycle's problem, if "O" is large enough.

Next we use the constructed subgraphs in the queue as the initial population for the first part of this algorithm, mentioned in the section 8.1.2, to continue finding isometric subgraphs until it terminates. The pseudo-code of this method is given in the Algorithm 2.

Algorithm 2 pseudocode of the locally based search algorithm

Procedure:
Step 1. Run the parallel search algorithm obtained by the Algorithm 1.
Step 2. Put the first k non-isometric subgraphs in the queue, and do the following.

If "O" equals to the order of the main graph G.
G is dp (Stop)

Else:
for H in the queue.

If $|H| \leq$ "$O + 1$".
Increase the order of H by 1, using the average linkage method.
Else:
Use H as an element of a new population.

for all H in new population.

If H is not isometric.
G might not be dp (stop).
Else:
Go to setp 1.

8.2 Chapter summary

In this chapter, we proposed an algorithm using the genetic algorithm 1 in chapter 7, in which some local search strategies are amalgamated to improve convergence speed by

setting $r = 1$ and $\mu = 0$. In addition, a selection operator, using the average linkage clustering method to attach cycles, is proposed to prevent premature convergence.

Chapter 9

Conclusions

The strongest arguments prove nothing so long as the conclusions are not verified by experience. Experimental science is the queen of sciences and the goal of all speculation.

-Roger Bacon

In this work we defined dp graphs and used a several theoretic approaches to study them. We proposed a number of conjectures to the understanding of dp graphs. The results which were shown in this present are : considering how to add a vertex to a dp graph so that the result is a dp graph. This condition implies that chordal graphs are dp, next we gave an equivalent condition to sequentially distance preserving based upon simplicial orderings. Using this condition, we proved that if a graph does not contain any induced cycles of length 5 or greater, then it is sdp and thus dp. we also considered the distance preserving property on graphs with a cut vertex.

We showed that if the minimum degree of any vertex is greater than half the number of vertices then the graph is distance preserving. And also we saw that if the minimum degree is greater than 2/3 of the number of vertices then the graph is sequentially distance preserving.

The presence of certain cycles can cause a graph not to be dp. we showed that if G is a graph with girth$(G) \geq 5$ and every vertex is either a cut vertex or in a cycle, then G does not have any isometric subgraph of order $|V(G)| - 1$ and so is not dp. We also define a family of graphs, namely $C_{k,l}$, and propose a condition under which $C_{k,l}$ is not dp.

A necessary and sufficient condition for the lexicographic product of two graphs to be a dp graph. Moreover, all isometric subgraphs of the lexicographic product of two arbitrary graphs will be characterized. We also showed that the Cartesian product of two graphs is sdp if and only if its factors are.

We proposed a biologically-inspired search algorithm to address the problem of finding an isometric subgraph of any order to determine if a given graph is distance preserving. In this algorithm, by using a well defined fitness function, the selection operator selects the almost isometric subgraphs to generate the offsprings for the next generation. There is also a tradeoff between the population size and searching speed. On one hand, the larger the population size is, the slower the search algorithm would be. On the other hand, the more we increase the population size, the higher is the chance to find an existing isometric subgraph.

A fast algorithm is introduced by using average linkage clustering notion in data mining and is compared with the other algorithms in the literature.

These all results together has application in fixed routs and also dynamic shuttle service and autonomous vehicles.

REFERENCES

REFERENCES

[1] Hans-Jürgen Bandelt and Henry Martyn Mulder. Distance-hereditary graphs. *Journal of Combinatorial Theory, Series B*, 41(2):182–208, 1986.

[2] Belá Bollobás. The diameter of random graphs. *Transactions of the American Mathematical Society*, 267(1):41–52, 1981.

[3] John Adrian Bondy and Uppaluri Siva Ramachandra Murty. *Graph theory with applications*, volume 6. Macmillan London, 1976.

[4] Boštjan Brešar, Sandi Klavžar, and Aleksandra Tepeh Horvat. On the geodetic number and related metric sets in cartesian product graphs. *Discrete Mathematics*, 308(23):5555–5561, 2008.

[5] Kenneth Mark Bryden, Daniel A Ashlock, Steven Corns, and Stephen J Willson. Graph-based evolutionary algorithms. *Evolutionary Computation, IEEE Transactions on*, 10(5):550–567, 2006.

[6] Václav Chvátal. On certain polytopes associated with graphs. *Journal of Combinatorial Theory, Series B*, 18(2):138–154, 1975.

[7] Guillaume Damiand, Michel Habib, and Christophe Paul. A simple paradigm for graph recognition: application to cographs and distance hereditary graphs. *Theoretical Computer Science*, 263(1):99–111, 2001.

[8] Alessandro D'Atri and Marina Moscarini. Distance-hereditary graphs, steiner trees, and connected domination. *SIAM Journal on Computing*, 17(3):521–538, 1988.

[9] Delbert R Fulkerson, Oliver A Gross, et al. Incidence matrices and interval graphs. *Pacific J. Math*, 15(3):835–855, 1965.

[10] Tibor Gallai. Transitiv orientierbare graphen. *Acta Mathematica Hungarica*, 18(1-2):25–66, 1967.

[11] Emeric Gioan and Christophe Paul. Split decomposition and graph-labelled trees: characterizations and fully dynamic algorithms for totally decomposable graphs. *Discrete Applied Mathematics*, 160(6):708–733, 2012.

[12] Martin Charles Golumbic and Udi Rotics. On the clique-width of some perfect graph classes. *International Journal of Foundations of Computer Science*, 11(03):423–443, 2000.

[13] Michel Habib and Christophe Paul. A survey of the algorithmic aspects of modular decomposition. *Computer Science Review*, 4(1):41–59, 2010.

[14] Peter L Hammer and Frédéric Maffray. Completely separable graphs. *Discrete applied mathematics*, 27(1):85–99, 1990.

[15] Pawan Harish and PJ Narayanan. Accelerating large graph algorithms on the gpu using cuda. In *High performance computing–HiPC 2007*, pages 197–208. Springer, 2007.

[16] Edward Howorka. A characterization of distance-hereditary graphs. *The quarterly journal of mathematics*, 28(4):417–420, 1977.

[17] Ruo-Wei Hung and Maw-Shang Chang. Linear-time algorithms for the hamiltonian problems on distance-hereditary graphs. *Theoretical Computer Science*, 341(1):411–440, 2005.

[18] Wilfried Imrich and Sandi Klavzar. *Product graphs*. Wiley, 2000.

[19] MH Khalifeh, Bruce E Sagan, and Emad Zahedi. Distance preserving graphs and graph products. *arXiv preprint arXiv:1507.04800*, 2015.

[20] R Krithika, Rogers Mathew, NS Narayanaswamy, and N Sadagopan. A Dirac-type characterization of k-chordal graphs. *Discrete Mathematics*, 313(24):2865–2867, 2013.

[21] László Lovász. A characterization of perfect graphs. *Journal of Combinatorial Theory, Series B*, 13(2):95–98, 1972.

[22] Rolf H Möhring. Algorithmic aspects of the substitution decomposition in optimization over relations, set systems and boolean functions. *Annals of Operations Research*, 4(1):195–225, 1985.

[23] Rolf H Möhring and Franz J Radermacher. Substitution decomposition for discrete structures and connections with combinatorial optimization. *North-Holland Mathematics Studies*, 95:257–355, 1984.

[24] John W Moon and Leo Moser. *Almost all (0, 1) matrices are primitive*. springer, 1965.

[25] R. Nussbaum and A-H. Esfahanian. Preliminary results on distance-preserving graphs. *Congressus Numerantium*, 211:141–149, 2012.

[26] Ronald Nussbaum, Abdol-Hossein Esfahanian, and Pang-Ning Tan. Clustering social networks using distance-preserving subgraphs. In *The Influence of Technology on Social Network Analysis and Mining*, pages 331–349. Springer, 2013.

[27] Dennis Ross, Bruce Sagan, Ronald Nussbaum, and Abdol-Hossein Esfahanian. On constructing regular distance-preserving graphs. *Congressus Numerantium, to appear*, 219:129–138, 2014.

[28] Horst Sachs. On the berge conjecture concerning perfect graphs. *Combinatorial Structures and their Applications, Gordon and Beach, New York*, 37:384, 1970.

[29] Jeffrey B Sidney and George Steiner. Optimal sequencing by modular decomposition: polynomial algorithms. *Operations Research*, 34(4):606–612, 1986.

[30] Jason P. Smith and Emad Zahedi. On distance preserving and sequentially distance preserving graphs. *arXiv preprint arXiv:1701.06404*, 2017.

[31] Wilfried and Iztok Peterin. Recognizing cartesian products in linear time. *Discrete mathematics*, 307(3):472–483, 2007.

[32] Da Yan, James Cheng, Wilfred Ng, and Steven Liu. Finding distance-preserving subgraphs in large road networks. In *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*, pages 625–636. IEEE, 2013.

[33] Emad Zahedi. Distance preserving graphs. *arXiv preprint arXiv:1507.03615*, 2015.