

Powers of Hamiltonian Paths in Interval Graphs

Garth Isaak*

DEPARTMENT OF MATHEMATICS

LEHIGH UNIVERSITY

BETHLEHEM, PA 18015

E-mail: gi02@lehigh.edu

Received February 26, 1996; revised October 14, 1997

Abstract: We give a simple proof that the obvious necessary conditions for a graph to contain the k^{th} power of a Hamiltonian path are sufficient for the class of interval graphs. The proof is based on showing that a greedy algorithm tests for the existence of Hamiltonian path powers in interval graphs. We will also discuss covers by powers of paths and analogues of the Hamiltonian completion number.

© 1998 John Wiley & Sons, Inc. J Graph Theory 27: 31–38, 1998

Keywords: *toughness, Hamiltonian, linear arrangement*

1. INTRODUCTION

The k^{th} power of a graph $G = (V, E)$ is the graph G^k with the same vertex set and $\{x, y\} \in E(G^k)$ if and only if the distance between x and y in G is at most k . We will say that $G = (V, E)$ contains the k^{th} power of a Hamiltonian path if $|V| = n$ and if G has a subgraph isomorphic to P_n^k where P_n is the path on n vertices. Similarly, G contains the k^{th} power of a Hamiltonian cycle if it has a subgraph isomorphic to C_n^k where C_n is the cycle on n vertices. We will say that G contains an HC^k and HP^k in these cases.

*Partially supported by ONR Grant N0014-94-1-0922.

Determining the largest k such that a graph G contains an HP^k can be viewed as a labeling problem which is “dual” to the bandwidth problem. For bandwidth we seek a labeling of the vertices which minimizes the maximum distance between adjacent vertices. To find the largest k such that G contains an HP^k we seek a labeling of the vertices which maximizes the minimum distance between nonadjacent vertices. This is equivalent to finding the separation number in the complement of G .

Let $c(S)$ denote the number of components in the subgraph of G induced by $V - S$.

Definition 1. G is t -tough if

$$\text{for all } S \subset V \text{ either } c(S) = 1 \text{ or } |S| \geq tc(S). \quad (1)$$

Definition 2. G is t -path tough if

$$\text{for all } S \subset V |S| \geq t(c(S) - 1). \quad (2)$$

Toughness was introduced by Chvatal [5]. It is well known that 1-toughness is a necessary condition for G to contain a Hamiltonian cycle and 1-path toughness is a necessary condition for G to contain a Hamiltonian path. It is easy to see that similar conditions hold for powers of Hamiltonian paths and cycles. For future reference we will write this as a lemma.

Lemma 1. (i) *If G contains the k^{th} power of a Hamiltonian cycle then G is k -tough.*

(ii) *If G contains the k^{th} power of a Hamiltonian path then G is k -path tough.*

In general these conditions are not sufficient. Indeed, there exist triangle-free graphs with arbitrarily large toughness (and hence also large path toughness) [2]. So, in contrast to Chvatal's conjecture for Hamiltonian cycles, for $k \geq 2$, there is no t such that t -tough implies the existence of the k^{th} power of a Hamiltonian cycle (and similarly for paths).

A (finite) interval graph is a graph that can be represented as the intersection graph of a set of real intervals. That is, there exist intervals $I_x = [l_x, r_x]$ such that $\{x, y\} \in E(G)$ if and only if $I_x \cap I_y \neq \emptyset$. (See for example Golumbic [10] for more about these graphs.) We will prove that for the class of interval graphs the path conditions are sufficient.

Theorem 1. *Let G be an interval graph. If G is k -path tough then G contains the k^{th} power of a Hamiltonian path.*

In fact, what we will show is that a greedy algorithm based on constructing a power of a path using vertices whose intervals have smallest endpoints “first” will correctly test for the power of a Hamiltonian path or produce a set S violating the necessary toughness conditions.

Hamiltonian paths and cycles ($k = 1$) in interval graphs have been examined in [3], [11], [12], [13], [1], [6], and [4] among others. Only in [13] and [4] is

the toughness condition discussed and used in proofs. Our algorithm will be a generalized version of those presented in [1] and [12] (see also [13]) but we have shorter proofs.

For Hamiltonian cycles in interval graphs, there is also a greedy algorithm based on endpoints of an interval representation [4], [12]. We can show that extending this algorithm in the “most obvious” manner to test for powers of Hamiltonian cycles fails to produce an algorithm that correctly tests for powers of Hamiltonian cycles. It is possible that algorithms based on clique representations [11], [6] will extend to the powers cases. There are also algorithms for Hamiltonian paths and cycles in the broader class of co-comparability graphs (see [7–9]).

We will assume that interval graph G has a representation with intervals denoted by $I_x = [l_x, r_x]$ and that the vertices are labeled such that $r_0 < r_1 < \dots < r_{n-1}$. So we will say $x < y$ when $r_x < r_y$. We can assume that the endpoints of the intervals are distinct (see for example Golumbic [10]) and that $0 < \epsilon < 1$ is a number less than the smallest gap between any pair of endpoints. We will denote the neighborhood of a vertex x by $N(x)$. There are efficient algorithms to construct interval representations of interval graphs.

2. THE ALGORITHM

Our aim is to show that the following simple greedy algorithm works to test for powers of Hamiltonian paths in interval graphs. We assume that we are given an interval graph G with a fixed interval representation. The vertices are labeled $0, 1, 2, 3, \dots, n-1$ so that $r_0 < r_1 < \dots < r_{n-1}$.

GREEDY ALGORITHM FOR HP^k :

- Start with $x_0 = 0$
- If the path power $P^k = 0, x_1, x_2, \dots, x_i$ is currently constructed and every vertex is in P^k , then P^k is Hamiltonian.
- If some vertex is not in P^k , let x_{i+1} be the vertex not already in P^k with smallest label (right endpoint in the interval representation) that is adjacent to the last k vertices $\{x_{i-k+1}, x_{i-k+2}, \dots, x_i\}$ (or $\{x_1, \dots, x_i\}$ if $i - k + 1 < 1$). If there is no such vertex then there is no HP^k .

From properties of the interval representation, when selecting x_{i+1} we only need to check adjacency with the minimum entry in $\{x_{i-k+1}, x_{i-k+2}, \dots, x_i\}$ (or $\{x_1, \dots, x_i\}$), which would allow for a more efficient implementation.

The idea of the proof is to delete 0 from G , appropriately modify the intervals for the remaining vertices to get a new graph G' for which the greedy algorithm produces the same order on G' as it does for these vertices in G , then show that an HP_{n-1}^k in G' extends to HP_n^k in G or if G' is not k -path tough then G is not k -path tough.

Theorem 2. *Either the greedy algorithm detects a k^{th} power of a Hamiltonian path or G is not k -path tough.*

Proof. If $|V| \leq k+1$ then it is easy to check that either G is the complete graph and the algorithm succeeds (and HP_n^k is complete in this case) or the algorithm fails and that $S = V - \{x, y\}$ for $\{x, y\} \notin E$ violates (2). So assume $|V| > k+1$. We can also assume that G is connected or else $S = \emptyset$ violates (2).

If $|N(0)| < k$ then vertex 0 is isolated in $G(V - N(0))$. So this graph has at least two components and $|N(0)| < k \leq k(2-1) \leq k(c(N(0)-1))$. This violates (2). Similarly if $|N(x_1)| < k$ then $\{0, x_1\}$ is a component of $G(V - N(x_1))$ and again (2) is violated. So assume that at least $k+2$ vertices are selected by the algorithm.

Let $0, x_1, x_2, \dots, x_k, x_{k+1}$ be the first $k+2$ vertices selected by the algorithm. Then $x_1 < x_2 < \dots < x_k$ and all these are adjacent to 0. Also, x_1 is adjacent to x_{k+1} . Let x_j, x_{j+1}, \dots, x_k be those vertices (if any) among $\{x_1, \dots, x_k\}$ which have larger right endpoint than x_{k+1} . Let R denote the right endpoint of interval x_{k+1} . Construct G' by deleting 0 and setting $r'_{x_j} = R - \epsilon, r'_{x_{j+1}} = R - \epsilon^2, \dots, r'_{x_k} = R - \epsilon^{j-k+1}$. That is, we shift the right endpoints of x_j, \dots, x_k to be just smaller than the right endpoint of x_{k+1} while maintaining their same relative order. Note that $G' \neq G - \{0\}$ since some of the adjacencies may have changed. However, all adjacencies to x_{k+1} have not been changed.

Let $y \neq 0$ have the second smallest label in G . If y is adjacent to 0 then $y = x_1$. If y is not adjacent to 0, then $I_y \subset I_{x_1}$ and so y is adjacent to x_1 . In this case $y = x_{k+1}$ and by construction x_1 has smallest label in G' . So in either case x_1 has smallest label in G' . Again by the construction and the choice of $x_{k+1}; x_2, x_3, \dots, x_{k+1}$ are selected (in order) after x_1 by the algorithm applied to G' .

When selecting x_{k+i} for $i = 2, 3, \dots, k$, x_{k+i} must be adjacent to $x_{k+i-k}, \dots, x_{k+i-1}$. It is straightforward to check that x_{k+i} has all these adjacencies in G' if and only if it has them in G . Hence, except for the initial vertex 0 in G the algorithm selects vertices in the same order for both G' and G .

If x_1, x_2, \dots, x_{n-1} is an HP_{n-1}^k detected in G' by the algorithm then $0, x_1, \dots, x_{n-1}$ is an HP_n^k in G since 0 is adjacent to x_1, \dots, x_k .

Otherwise, by induction there is an S' violating (2) in G' . That is, such that $|S'| < k(c_{G'}(S') - 1)$.

If $\{x_j, \dots, x_k\} \subseteq S'$ then all modified intervals are deleted. Then since $N_G(0) \subseteq N_G(y)$ for $\{0, y\} \in E$, $c_{G'}(S') \leq c_G(S')$ and S' violates (2) in G . Note that this includes the case that no intervals are modified (i.e., $x_{k+1} > x_k$). So we may assume that $x_{k+1} < x_k$ and that some vertex in $\{x_j, \dots, x_k\}$ is in $V(G') - S'$.

If there exists a vertex $y \in V(G') - S' - \{x_1, x_2, \dots, x_k\}$ adjacent to 0 then $x_y > x_k$ and $N_{G'}(y) \cup \{0\} = N_G(y) \supseteq N_G(x_i) \supseteq N_{G'}(x_i) \cup \{0\}$ for $i = 1, 2, \dots, k$. Also, $N_G(y) \supseteq N_G(0)$. Hence $c_{G'}(S') \leq c_G(S')$ and S' violates (2) in G .

Assume $y \in V(G') - S' - \{x_1, x_2, \dots, x_k\}$ implies $\{y, 0\} \notin E(G)$ and that $\{x_1, \dots, x_k\} \cap (V' - S')$ form a component C of $G'(V(G') - S')$. Some vertex $x \in \{x_j, x_{j+1}, \dots, x_k\}$ is in $V(G') - S'$. Then $I_{x_{k+1}} \subset I_x$ and so there is no vertex

in $V(G') - S' - \{x_1, x_2, \dots, x_k\}$ adjacent to x_{k+1} since such a vertex would be in C . Let $S = (S' - x_{k+1}) \cup \{x_1, \dots, x_k\}$. Then, in $G(V - S)$, 0 and x_{k+1} are isolated, C no longer appears and the remaining components are the same as in $G'(V' - S')$. So $c_{G'}(S') + 1 = c_G(S)$ and $|S| < |S'| + k < k(c_{G'}(S') - 1) + k = kc_{G'}(S') = k(c_G(S) - 1)$ violating (2).

Finally, assume $y \in V(G') - S' - \{x_1, x_2, \dots, x_k\}$ implies $\{y, 0\} \notin E(G)$ and that $\{x_1, \dots, x_k\} \cap (V' - S')$ is properly contained in a component C of $G'(V' - S')$. Let $S = S' \cup \{x_1, \dots, x_k\}$. Then 0 is isolated in $G(V - S)$, C breaks into at least one component in $G(V - S)$ and the remaining components are the same in $G(V - S)$. So $c_{G'}(S') \leq c_G(S) - 1$ and $|S| \leq |S'| + k < k(c_{G'}(S') - 1) + k = kc_{G'}(S') \leq k(c_G(S) - 1)$ violating (2). ■

Note that running the algorithm recursively, as described in the proof, will allow construction of S violating (2).

3. HAMILTONIAN COMPLETION PROBLEMS

We will conclude by briefly discussing some related problems related to the k^{th} power version of Hamiltonian completion.

For general graphs G , the path partition number is the minimum number of paths needed to cover the vertices and the Hamiltonian (path) completion number is the minimum number of edges needed to add to make G have a Hamiltonian path. Denoting these by $pp(G)$ and $hc(G)$ it is easy to see (and well known) that $pp(G) - 1 = hc(G)$. The same is not true for the k^{th} power analogues. That is, $pp_k(G)$ is not necessarily equal to $hc_k(G)$. We will examine a third version of these parameters which has a natural extension in the k^{th} power case.

Let the join of two graphs $G \vee H$ on disjoint vertex sets have $V = V(G) \cup V(H)$ and $E = E(G) \cup E(H) \cup \{xy | x \in V(G), y \in V(H)\}$. Define the Hamiltonian vertex completion number $vc(G)$ to be the minimum t such that $G \vee K_t$ contains a Hamiltonian path (where K_t is the complete graph on t vertices). For Hamiltonian paths it is easy to see that vertex completion and (edge) completion are the same, i.e., $vc(G) = hc(G)$. As above the same is not true for the k^{th} power analogues, $vc_k(G)$ and $hc_k(G)$.

If there is an efficient algorithm for detecting an HP^k , then there is one for determining $vc_k(G)$ for a class closed under adding vertices adjacent to all vertices (such as co-comparability graphs). Simply add vertices one at a time. For interval graphs this can be done more efficiently. Given G , run the algorithm on $G \vee K_t$ where $t \geq k|V(G)|$. Observe that $G \vee K_t$ contains an HP^k . Use an interval representation for G along with intervals for K_t all of which contain every $V(G)$ interval (and hence have larger right endpoints than any vertex of G). Then simply count the number of K_t vertices added to the Hamiltonian path power before the last vertex of $V(G)$. This is $vc_k(G)$. (Of course the extra vertices do not need to be 'added' until they are needed.)

The following elementary result also indicates that $vc_k(G)$ is a natural version of completion/path packing to examine. It would also be used in a more formal proof that the interval graph algorithm of the previous paragraph works.

Theorem 3. *Let \mathcal{C} be a class of graphs satisfying the following*

- (i) *If $G \in \mathcal{C}$ then $G \vee K_t \in \mathcal{C}$. (\mathcal{C} is closed under adding a new vertex adjacent to all current vertices.)*
- (ii) *For $G \in \mathcal{C}$, G contains an HP^k if and only if G is k -path tough.*

Then for $G \in \mathcal{C}$,

$$vc_k(G) = \max_{S \subseteq V} \{k(c(S) - 1) - |S|\}.$$

Proof. We first show that for general graphs $vc_k(G) \geq \max_{S \subseteq V} \{k(c(S) - 1) - |S|\}$. $G' = G \vee K_{vc_k}$ has an HP^k by the definition of $vc_k(G)$ and thus G' is k -path tough. Consider $S \subseteq V(G)$. Let $S' = S \cup V(K_{vc_k})$. Note that $c_G(S) = c_{G'}(S')$. By the k -path toughness of G' , $|S| + vc_k = |S'| \geq k(c_{G'}(S') - 1) = k(c_G(S) - 1)$.

Conversely, let $t = \max_{S \subseteq V(G)} \{k(c(S) - 1) - |S|\}$. If $G' = G \vee K_t$ does not contain an HP^k then by (i) and (ii), G' is not k -path tough. So there exists $S' \subseteq V(G')$ violating (2). As above, $S' = S \cup V(K_t)$ and $c_{G'}(S') = c_G(S)$. Then $|S| + t = |S'| < k(c_{G'}(S') - 1) = k(c_G(S) - 1)$ contradicting the choice of t . ■

Observe that a similar proof easily shows that for a class of graphs satisfying condition (i) of Theorem 3 if k -toughness is sufficient for G to contain the k^{th} power of a Hamiltonian cycle then k -path toughness is sufficient for G to contain the k^{th} power of a Hamiltonian path.

There are a number of bounds on the parameters pp_k , hc_k and vc_k . We will briefly mention some of these and indicate intuition behind them. More formal proofs are easily obtained.

As in Theorem 3 we have

$$vc_k(G) \geq \max_{S \subseteq V} \{k(c(S) - 1) - |S|\}.$$

This bound is not tight. The triangle free graphs with arbitrarily large toughness constructed in [2] have the max equal to 0 and $vc_k \geq |V|/2 - 1$ and $pp_k \geq |V|/2$ since at most two vertices can be in each part of a path partition.

If $G(V - S)$ has $c(S)$ components, then ‘‘connecting’’ two in a path power requires at least k vertices from S . So

$$pp_k(G) \geq \max_{S \subseteq V} \left\{ c(S) - \left\lfloor \frac{|S|}{k} \right\rfloor \right\}.$$

This bound is not tight even for interval graphs. Consider for example the path power P_{kn+1}^{k-1} , which has $pp_k = n + 1$ and the maximum in the equation equal to $(n + 1) - \lfloor n(k - 1)/k \rfloor$. So the gap can be arbitrarily large.

“Connecting” two parts of a minimum size path power partition into a single path power requires at least one extra vertex and at most k extra vertices. Then

$$k(pp_k(G) - 1) \geq vc_k(G) \geq pp_k(G) - 1.$$

“Connecting” two parts of a minimum size path power partition into a single path power requires at least one new edge and at most $k(k + 1)/2$ new edges. Then

$$\frac{k(k + 1)}{2}(pp_k(G) - 1) \geq hc_k(G) \geq pp_k(G) - 1.$$

Each of the possible values for vc_k and hc_k in the previous two inequalities can be obtained. Start with a graph consisting of the disjoint union of pp_k k^{th} powers of paths each with length $j \geq 2k$. Say the vertices of the i^{th} path power ($i = 1, 2, \dots, pp_k$) are $x_{i1}, x_{i2}, \dots, x_{ij}$ with x_{ij} adjacent to $x_{ij'}$ if and only if $|j - j'| \leq k$.

For the vertex inequalities add new vertices y such that for some i , y is adjacent to $x_{i(j-k+1)}, x_{i(j-k+2)}, \dots, x_{ij}$ and to $x_{(i+1)1}, x_{(i+1)2}, \dots, x_{(i+1)k}$ taking care to add at most $k - 1$ such y 's “connecting” the i^{th} and $(i + 1)^{st}$ parts.

For the edge inequalities, add an appropriate subset of edges of the form $x_{ip}x_{(i+1)q}$ where $j - p + q \leq k$ omitting the edge $x_{ij}x_{(i+1)1}$ for each i . Note that if only the $pp_k - 1$ edges $x_{ij}x_{(i+1)1}$ are omitted then the resulting graph has $hc_k = pp_k - 1$ and $vc_k = k(pp_k - 1)$.

Although each of the possible values for vc_k and hc_k can be obtained as above, there is some relation between these parameters. Given an HP^k in $G \vee K_{vc_k}$ deleting the new vertices yields a path power partition with $q \geq pp_k(G)$ parts. Let z_i denote the number of new vertices “connecting” the i^{th} and $(i + 1)^{st}$ parts in the partition. Each of these connections could also be made using at most $z_i(z_i + 1)/2$ new edges. Hence $hc_k(G) \leq \sum_{i=1}^q z_i(z_i + 1)/2$.

It is not clear if it is possible to use a variant on the greedy algorithm to determine pp_k or hc_k in interval graphs. The “obvious” approach for determining pp_k by running the algorithm until it fails, then starting over with the remaining vertices fails. Consider the graph represented by $I_1 = I_2 = [0, 1], I_3 = I_4 = [0, 6], I_5 = [2, 3], I_6 = [4, 5]$. If we try to modify the algorithm to “reuse” end vertices so that the algorithm works for the previous graph, then difficulties are encountered with the algorithm on simple paths.

References

- [1] S. R. Arikati and C. P. Rangan, Linear algorithm for optimal path cover problem on interval graphs, *Inform. Process. Lett.* **35** (1990), 149–153.
- [2] D. Bauer, J. van den Heuval, and E. Schmeichel, Toughness and triangle free graphs, *J. Comb. Theory B* **65** (1995), 208–221.
- [3] M. A. Bonuccelli and D. P. Bovet, Minimum node disjoint path covering for circular arc-graphs, *Inform. Process. Lett.* **8** (1979), 159–161.

- [4] M.-S. Chang, S.-L. Peng, and J.-L. Liaw, Deferred query—an efficient approach for problems on interval and circular-arc graphs, in F. Dehne et al. (ed.), *Algorithms and data structures*, Lecture Notes in CS 709, Springer-Verlag, New York (1993), 222–233.
- [5] V. Chvátal, Tough graphs and Hamiltonian circuits, *Discrete Math.* **5** (1973), 215–228.
- [6] P. Damaschke, Paths in interval graphs and circular arc graphs, *Discrete Math.* **112** (1993), 49–64.
- [7] P. Damaschke, J. S. Deogun, D. Kratsch, and G. Steiner, Finding Hamiltonian paths in cocomparability graphs using the bump number algorithm, *Order* **8** (1992), 383–391.
- [8] J. S. Deogun, D. Kratsch, and G. Steiner, 1-Tough cocomparability graphs are Hamiltonian, McMaster Univ. Dept. of Comp. Sci. technical report 94-05 (1994).
- [9] J. S. Deogun and G. Steiner, Polynomial algorithms for Hamiltonian cycle in cocomparability graphs, *SIAM J. Comput.* **23** (1994), 520–552.
- [10] M. C. Golumbic, *Algorithmic graph theory and perfect graphs*, Academic Press, New York (1980).
- [11] J. M. Keil, Finding Hamiltonian circuits in interval graphs, *Inform. Process. Lett.* **20** (1985), 201–206.
- [12] G. K. Manacher, T. A. Mankus, and C. J. Smith, An optimum $\Theta(n \log n)$ algorithm for finding a canonical Hamiltonian path and a canonical Hamiltonian circuit in a set of intervals, *Inform. Process. Lett.* **35** (1990), 205–211.
- [13] W.-K. Shih, T. C. Chern, and W.-L. Hsu, An $O(n^2 \log n)$ algorithm for the Hamiltonian cycle problem on circular arc graphs, *SIAM J. Comput.* **21** (1992), 1026–1046.