Hamiltonian Cycles in Permutation Graphs

Jitender S. Deogun and Charles Riedesel
Department of Computer Science and Engineering
University of Nebraska – Lincoln
Lincoln, NE 68588-0115, U.S.A.

Abstract

In this paper we employ the structures of a permutation graph as exhibited in the Euclidean representation to solve the existence and construction problems of Hamiltonian cycles on permutation graphs. We define and prove the existence of a *layered* Hamiltonian cycle in a Hamiltonian permutation graph. A linear (in size) time and (in order) space algorithm for construction of a layered Hamiltonian cycle on a permutation graph is presented and its correctness proven.

1 Introduction

The *Hamiltonian Cycle* (HC) problem is one of the classical problems in graph theory. The HC problem involves the existence and construction of a cycle on a given graph such that each vertex is visited exactly once. This problem is substantially more difficult than the superficially similar Eulerian problem in which each edge is to be traversed exactly once. For graphs in general the HC problem has been proven to be NP-complete [13]. Although the HC problem remains NP-complete for the class of perfect graphs, it does become tractable for several important subclasses such as interval graphs [14] and cocomparability graphs [4].

The HC problem for permutation graphs has been a well known open problem since 1979 [13]. An $O(n^3)$ time algorithm for the HC problem on permutation graphs follows from the superclass of cocomparability graphs [4]. That result is based on exploiting the relationship between the existence of a Hamiltonian cycle in a cocomparability graph and the bump number of a corresponding partial order [12]. In this paper we present an $O(n^2)$ time algorithm for the HC problem in permutation graphs. Our approach is based on exploiting the Euclidean representation for permutation graphs.

Three main representations have been developed which reveal some fundamental structure of permutation graphs. These are the *permutation* or *matching diagram*, the *interval containment diagram*, and the *Euclidean representation*.

JCMCC 27 (1998), pp. 161-200

Among these, the permutation diagram has been the tool of choice for investigating problems on permutation graphs [8, 11, 19]. The interval containment diagram and the Euclidean representation have been used in only a few studies [9, 20]. However, we have found that the Euclidean representation is useful for studying a number of problems on permutation graphs, including toughness, traceability, clustering, and Hamiltonicity [18, 6]. Of particular significance for this work on the HC problem, the Euclidean representation readily reveals a two-dimensional layering structure which is not apparent in either the permutation diagram or the interval containment diagram. This spatial perspective can be exploited in the solution of the HC problem.

We present the structural properties of the Euclidean representation that are relevant to the HC problem. We show that in a Euclidean representation the vertices can be partitioned into *layers* that induce complete subgraphs. The organization of the layers facilitates identification of *interlayer edges* for use as a *framework* for a particular sort of Hamiltonian cycle that we call a LAYERED HAMILTONIAN CYCLE. A set of greedy rules for selecting optimal interlayer edges for the framework is presented. We call our algorithm the *Greedy Layered Hamiltonian Cycle (GLHC) algorithm.* We prove that the GLHC algorithm constructs a LAYERED HAMILTONIAN CYCLE in $O(n^2)$ time and $O(n)$ space given a defining permutation for the graph. (A defining permutation can be computed in $O(n^2)$ time [3].)

The remainder of this paper is organized as follows. Section 2 provides a concise overview of relevant terminology and introduces the concepts of comparability graphs and permutation graphs. An overview of the Euclidean representation of permutation graphs is presented in Section 3. For more complete discussion the reader is referred to [18] and [5]. Subsections 3.1 and 3.2 define and discuss the characteristics of layers and their relevance to the existence and construction of a LAYERED HAMILTONIAN CYCLE respectively. The structural properties of layers with respect to interlayer edges are presented in Section 4. A proof of the existence of a LAYERED HAMILTONIAN CYCLE in a Hamiltonian permutation graph is given in Section 5. In Section 6 is coverage of the GLHC algorithm. We prove correctness of the GLHC algorithm in Section 7. An example of the algorithm is traced in Section 8. A summary concludes the paper in Section 9. The GLHC algorithm for permutation graphs is presented in pseudocode in an appendix.

2 Preliminaries and Definitions

For standard graph theoretic terminology used in this paper the reader is referred to sources such as [2] or [11].

Perfect graphs are an important class of graphs introduced by Claude Berge in 1961 [1]. Comparability graphs and cocomparability graphs are two well known subclasses of perfect graphs. A graph is called a *comparability graph* if its edges

admit an anti-symmetric, transitive orientation. Such an orientation imposes a partial ordering on the vertices; thus the justification of a partial order theoretic approach to its study. A graph G is called a *cocomparability graph* if its complement graph G^C is a comparability graph. Of these two, only the cocomparability graphs admit efficient Hamiltonian cycle algorithms [4]. A *permutation graph* is a graph that is both a comparability graph and a cocomparability graph [16].

Permutation Graphs. Let π denote a permutation on the first n natural numbers. Then π^{-1} denotes the inverse permutation which represents positions in π. For example, if $\pi = [5,1,3,4,2]$, then $\pi^{-1} = [2,5,3,4,1]$ so that $\pi_5 = 2$ and the position of 2 in π is given by $\pi_2^{-1} = 5$. Denote by π^R the reverse permutation of π. In the example $\pi^R = [2,4,3,1,5]$.

An undirected graph $G(\pi) = (V, E)$ can be defined as follows [8, 11]:

$$V = \{1, 2, \ldots, n\} \tag{1}$$

and

$$ij \in E \Leftrightarrow (i-j)(\pi_i^{-1} - \pi_j^{-1}) < 0,\ i,j \in V. \tag{2}$$

It may be noted that the graphs for different permutations can be isomorphic. For example, permutations $[4,3,1,2]$, $[4,2,3,1]$ and $[3,4,2,1]$ are each associated with a 4-cycle containing a chord.

Definition. A graph G is a permutation graph if there exists a permutation π (called a *defining permutation* of G) such that $G \cong G(\pi)$.

An immediate implication of this definition is that the vertices V of a permutation graph admit a labeling $\mathcal{L}(V)$ or simply $\mathcal{L}$ (called a *permutation labeling*) such that

$$ij \in E \Leftrightarrow [\mathcal{L}(i) - \mathcal{L}(j)][\pi_{\mathcal{L}(i)}^{-1} - \pi_{\mathcal{L}(j)}^{-1}] < 0. \tag{3}$$

Thus two vertices in a permutation graph are adjacent iff their permutation labels appear in reverse order in the defining permutation. From this characterization it is clear that the complement graph of $G(\pi)$ has π^R as a defining permutation and is therefore also a permutation graph. An orientation of the edges of a permutation graph toward the vertices with the larger (or equivalently the smaller) permutation labels is clearly transitive. Thus a permutation graph is a comparability graph and a cocomparability graph as previously stated. A permutation labeling of vertices will always be assumed in this paper. Moreover, the vertices V of a permutation graph are identified with their permutation labels.

3 Euclidean Representation of Permutation Graphs

As documented in a 1962 paper [15], the Euclidean representation was originally applied by Ore to partial orders as a tool for studying their dimension. Supowit

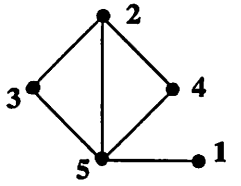
applied the Euclidean representation to the study of permutation graphs in 1985 [20]. Riedesel used the Euclidean representation to solve a number of problems on permutation graphs [18]. In 1995 Felsner and Wernisch adapted the Euclidean representation for the study of trapezoid graphs, a superclass of permutation graphs [10]. We demonstrate its usefulness for studying permutation graphs by appealing to this representation in solving the HC problem.

Ore showed that a partial order of dimension d can be represented in a d-dimensional Cartesian graph in which the axes correspond to the d linear extensions of a realizer [15]. The elements are represented as points in a quadrant of the space. The coordinates of each point are the positions of the element in the corresponding linear extensions. A permutation graph $G(\pi) = (V, E)$ is easily seen to correspond to a partial order of (at most) dimension 2 by letting one linear extension of a realizer be the natural ordering of its permutation labeling $\mathcal{L}$, (i.e. $1, 2, \ldots, n$), and the other linear extension be the permutation π [7]. Thus E represents the elements **not** in the partial order.

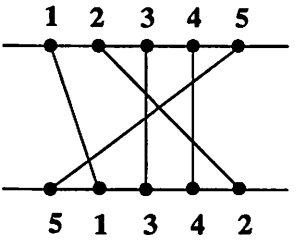
Let $G = G(\pi)$ be a permutation graph and $\mathcal{L}$ be a permutation labeling for G. We display the 2-D Euclidean representation, $\mathcal{E}(G, \mathcal{L})$, in the fourth quadrant for the simple reason that the axes and their labels will appear at the top and left sides of the drawing. The horizontal axis is labeled with the natural numbers 1 through n, i.e. $\mathcal{L}$, and the vertical axis with the permutation π of those numbers. Each vertex is represented by a point with coordinates corresponding to its label on both axes. For example, the Euclidean representation for the permutation graph with the permutation labeling given in Figure 1a is shown in Figure 1c. From the correspondence of a permutation graph with a partial order, it is easy to see that two vertices are adjacent if and only if a line joining them has a positive slope.

Each vertex v can be seen to divide $\mathcal{E}$ into five sets or regions: $Q_0(v) = \{v\}$ and quadrants $Q_i(v)$, $i = 1, 2, 3, 4$, having their origin at v and partitioning the remaining vertices according to their slope and thus adjacency to v. (See Figure 2). A vertex in quadrant 1 or 3 is on a positively sloped line and thus is adjacent to v while a vertex in quadrant 2 or 4 is on a negatively sloped line and thus is not adjacent to v. Furthermore, any two vertices such that one is in quadrant 1 and the other in quadrant 3 must be adjacent, while any two vertices such that one is in quadrant 2 and the other in quadrant 4 can not be adjacent. These relationships are called the *adjacency properties of quadrants*. An immediate observation is that in the Euclidean representation any set of vertices that has pairwise positive slope has a southwest to northeast ordering in $\mathcal{E}$ and induces a clique in G. Similarly any set of vertices that has pairwise negative slope induces an independent set in G.

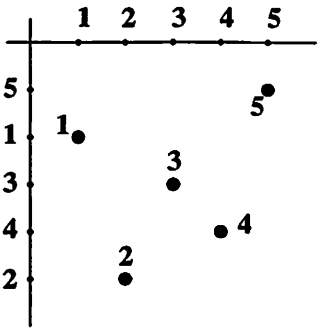
The concept of *quadrants* centered on a single vertex can be generalized as



(a) A Permutation Graph



(b) Permutation Diagram



(c) Euclidean Representation

Figure 1: Representations of a Permutation Graph

follows for a set of vertices $V' \subseteq V$ and the adjacency properties still hold:

$$\begin{aligned} Q_0(V') &= V' \\ Q_i(V') &= \bigcap_{v \in V'} Q_i(v),\ i = 1,2,3,4 \end{aligned} \tag{4}$$

The HC solution makes use of quadrants centered on sets of vertices comprising *layers* in the Euclidean representation.

The vertex sets *Top* and *Bottom* are defined as follows. The names are indicative of their respective positions in $\mathcal{E}$, as demonstrated in Figure 3(a).

Definition.

$$Top = \{v \mid Q_2(v) = \varnothing\} \tag{5}$$

$$Bottom = \{v \mid Q_4(v) = \varnothing\} \tag{6}$$

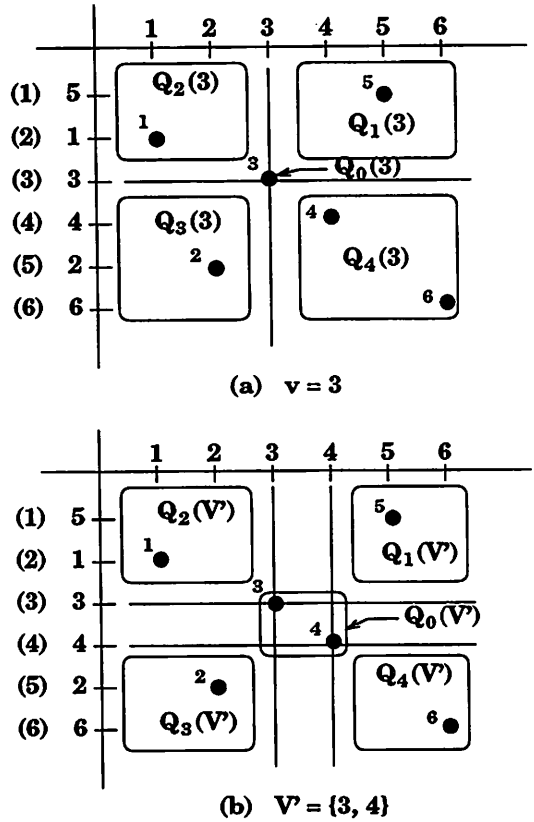


Figure 2: Quadrants for a (a) Vertex and a (b) Set of Vertices

A vertex in the set Top is called a *top* vertex, and similarly a vertex in the set $Bottom$ is called a *bottom* vertex. By the adjacency properties of quadrants, it is clear that the sets Top and $Bottom$ induce cliques in the graph.

A *height* relation consistent with the concept of top and bottom vertices can be defined for the pairs of non-adjacent vertices in G. Given two non-adjacent vertices u and v, u is said to be *higher* than v, denoted by $u \succ v$, if $u \in Q_2(v)$, or equivalently, $v \in Q_4(u)$. That is, u is northwest of v. If $u \succ v$, then $v \prec u$ and v is said to be *lower* than u. It follows from the definition that the *higher* (*lower*) relation is antisymmetric and transitive. Furthermore, no vertex is higher than a top vertex or lower than a bottom vertex. The height relation corresponds to the ordering relation of a partial order that is represented by the graph when

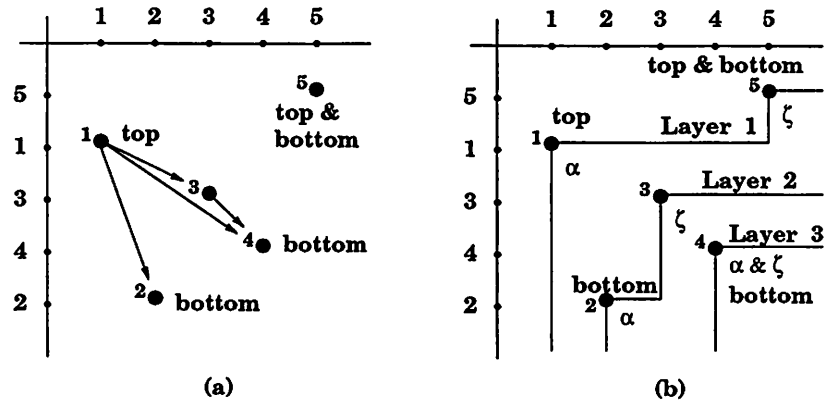


Figure 3: (a) – Top and Bottom Vertices and the Height Relation. (b) – The Highest Layering shown in Staircase Fashion.

considered as a cocomparability graph.

3.1 Layers in a Euclidean Representation

Let $G = (V, E)$ be a permutation graph with a permutation labeling $\mathcal{L} = \mathcal{L}(V)$ and maximum independent set size Σ.

Definition. A *layering* of $\mathcal{E}(G, \mathcal{L})$ is defined to be a partitioning of V into subsets $V_1, V_2, \ldots, V_\Sigma$ (called *layers*) such that the following statements are true:

1. V_i, $1 \leq i \leq \Sigma$, induces a clique in G.
2. Given $u \in V_i$ and $v \in V_j$, $i < j$, then either
   - u and v are adjacent or
   - $u \succ v$.

Clearly, there can be more than one layering for a given Euclidean representation. That there is at least one layering is also clear given the construction that follows.

To aid in the visual comprehension of a layer in the Euclidean representation, we draw a staircase pattern, rising to the right, of line segments using each vertex of the layer as the outer tip of a step. See Figure 3(b). For convenience of reference, the layers are considered to be ordered vertically from top to bottom, with each layer being a set of vertices arranged from left to right which are labeled $\alpha, \beta, \gamma, \ldots \chi, \psi, \zeta$ respectively.

For this paper we use a particular layering called the *highest layering*. In the highest layering, the staircases do not overlap. The highest layering can be identified iteratively: The first layer consists of the vertices in the set *Top* for the graph. Each subsequent layer is identified similarly from the induced subgraph of the remaining vertices. This layering places each vertex in the highest possible layer. Thus, while the initial set *Top* is exactly the first (highest) layer, the set *Bottom* contains the vertices in the lowest layer and in addition may contain vertices in higher layers. For an example see Figure 3(b) in which vertices 1 and 5 are top vertices and vertices 2, 4, and 5 are bottom vertices. Vertices 2 and 5 are bottom vertices even though they are not in the lowest layer. A vertex which can be assigned to another layer such that the new arrangement is a valid layering is said to be *assignable* to that layer. A highest layering is always assumed in this paper unless otherwise indicated.

Definition. A *highest* layering of a permutation graph is a layering created by the greedy strategy described above.

3.2 Layered Cycles

Our algorithm constructs a LAYERED HAMILTONIAN CYCLE. The definition for an LAYERED HAMILTONIAN CYCLE in a permutation graph is based on the concept of *layered paths*, in that a cycle can be viewed as two paths joined at their terminal vertices, and that the paths are layered in some fashion.

Definition. A *layered path* is a simple path $P = [v_{p1}, v_{p2}, \ldots, v_{pk}]$, $k \leq n$, in a permutation graph in which $v_{pi} \preceq v_{pj}$, $1 \leq i < j \leq k$, for some permutation labeling.

Thus, in a layered Hamiltonian path, the first vertex of a traversal is a bottom vertex of the graph and each subsequent vertex is a bottom vertex of the subgraph induced by the untraversed vertices. In [18] it was proved that a layered Hamiltonian path exists in a traceable permutation graph, and that such a path can be found using an efficient greedy algorithm. Its construction involves an attempt to build the path layer by layer through the highest layering of the graph beginning with the lowest layer.

Definition. A LAYERED HAMILTONIAN CYCLE is a Hamiltonian cycle that can be divided into two paths such that each path is a layered Hamiltonian path in the subgraph induced by its vertices. The two paths are referred to as HC-*paths*.

We prove that a LAYERED HAMILTONIAN CYCLE can be constructed in a Hamiltonian permutation graph. Its construction is similar to that of the layered Hamiltonian path except that two disjoint paths must be found through the layers.

4 Interlayer Edges

The challenge in constructing a LAYERED HAMILTONIAN CYCLE is in choosing interlayer edges such that the HC-paths can collectively be extended through all the layers. The set of interlayer edges for each HC-path is called a *path framework*. Together, the two path frameworks constitute a CYCLE FRAMEWORK. In this section an HC-path is viewed as a Hamiltonian path in the subgraph induced by the vertices of the path.

The ordering of vertices within a layer has implications for the existence of interlayer edges. In this section we explore the structure of layers and develop the notion of a *transition* as a sequence of interlayer edges used to continue a path between two layers. A path framework then consists of a set of these transitions. In a later section we apply these concepts to the combined view of both HC-paths in the entire graph.

A *transition* extends a path from a given layer called an *exit layer* to a higher layer called an *entry layer*. In the case of a Hamiltonian path, the entry layer is always the next higher layer. The ideal situation is a transition consisting of exactly one edge between the exit and entry layers. However, if no such edge is found, then a two edge transition is sought. This consists of an edge to a vertex in a higher *transition layer* and a second edge back to the entry layer. See Figure 4 for examples of one and two edge transitions. To be sure that such a transition does lead to a layered path, the first edge is sought to a vertex in the transition layer that is assignable to the entry layer. We showed in [18] that such a transition vertex can be found efficiently by utilizing the layer structure. Moreover, we showed that only one and two edge transitions are needed to construct a Hamiltonian path framework in a traceable graph. Therefore a transition either consists of one edge from the exit layer directly to the entry layer, or two edges, the first from the exit layer to a higher transition layer and the second back to the entry layer. The vertices involved in a two edge transition are called the *exit*, *transition*, and *entry* vertices respectively. In a single edge transition, the transition vertex coincides with the entry vertex.

A greedy approach for finding transition vertices is presented in [18]. The following lemma provides the basis for identifying the potential transition vertices in the lowest possible transition layer. It is the case that each vertex in the lowest possible transition layer to which there exists an interlayer edge (from the exit layer) is a valid transition vertex.

Lemma 4.1 (Transition). *[18] Let $G = (V, E)$ be a permutation graph, and let I be the lowest layer above K for which $\exists\, v_i \in I$, $v_k \in K$ such that $v_i v_k \in E$. Then for all J, $I \leq J < K$, the induced subgraph $G(v_i \cup V_J)$ is a clique. Furthermore, v_i is assignable to layer J.*

The only quadrant of a vertex which can contain non-adjacent vertices of higher layers is Q_2. Q_2 quadrants for layers are easily seen to be geometrically

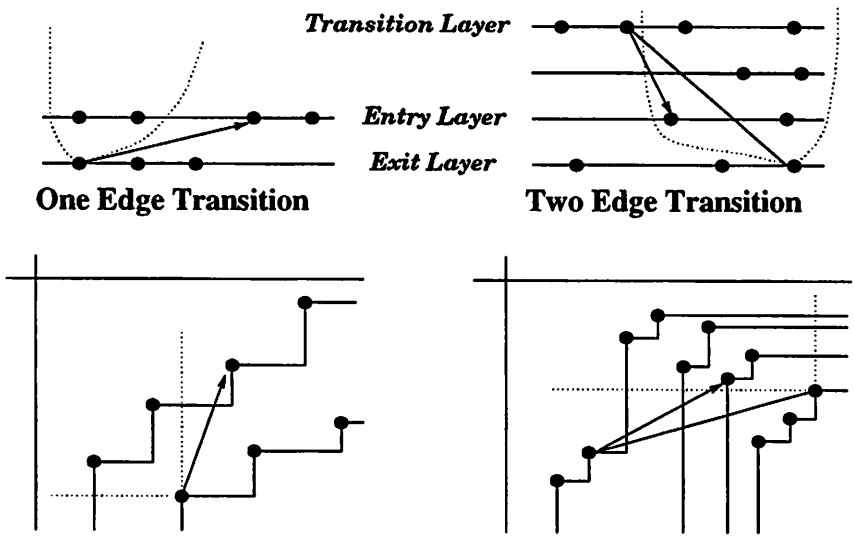


Figure 4: One and Two Edge Transitions

nested in the highest layering of the Euclidean representation. See Figure 5(b). Let V_k^i be the set of vertices in a layer I for which there are any adjacencies to vertices in a lower layer K. Then the nesting of quadrants Q_2 implies that $V_k^i \subseteq V_j^i$ for any intermediate layer J. Thus accessibility (via edges) to individual vertices of a given layer from a lower layer increases with the height of the latter layer. The following lemma implies that the nesting property of Q_2 quadrants for layers is maintained even when some transition vertices have been removed from higher layers (i.e. by being assigned to lower layers.)

Lemma 4.2 (Nesting). *[18] Let $G = (V, E)$ be a permutation graph, and I and J be two layers of G, I higher that J. If $V_{I'} \subset V_I$ is a set of bottom vertices of the induced subgraph $G[V_1 \cup V_2 \cup \ldots \cup V_J]$, then $Q_2(V_I - V_{I'}) \subseteq Q_2(V_J)$.*

A layered approach to cycle construction is also possible in the superclass of cocomparability graphs by defining layers based on vertex height in a Hasse diagram [4]. What is unique with permutation graphs is that each layer has a linear structure that admits a more efficient algorithm for finding optimal transitions. This structure is easily seen in the Euclidean representation. The Euclidean representation displays a permutation graph as being ordered both vertically and horizontally, i.e. 1) the linear ordering of layers, and 2) the linear ordering of vertices in a layer, respectively.

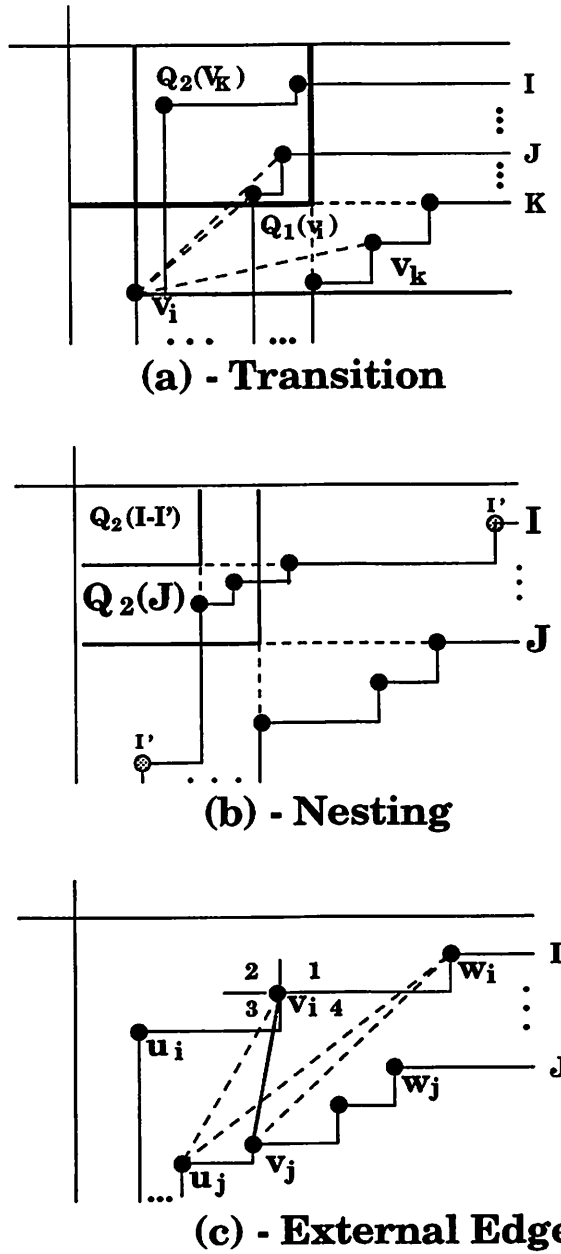


Figure 5: Illustrations for Lemmas

Each layer in a Euclidean representation is structured in such a way that the existence of interlayer edges between two layers can be determined in constant time by checking only for the two possible edges between vertices from opposite ends of the two layers. That is, if neither of the edges $\alpha_i\zeta_j$ or $\zeta_i\alpha_j$ exists for two layers I and J, then no edges exist between those two layers. These edges are referred to as *external edges* and, based on their appearance in a Euclidean representation, are said to be oriented *left to right* and *right to left* respectively. This property represents a specific case of the following lemma in which u is replaced with α and v with ζ.

Lemma 4.3 (External Edge). *[18] Let $G = (V, E)$ be a permutation graph, I and J be two layers of G, $I < J$, u_i, v_i, w_i be vertices in I, $u_i < v_i < w_i$, and u_j, v_j, w_j be vertices in J, $u_j < v_j < w_j \in J$. If $v_iv_j \in E$, then the following statements are true:*

I. Either $v_iu_j \in E$ or $v_iw_j \in E$,

II. Either $u_iv_j \in E$ or $w_iv_j \in E$, and

III. Either $u_iw_j \in E$ or $w_iu_j \in E$.

From these structural properties, an efficient algorithm for the construction of transitions for a Hamiltonian path framework emerges: Beginning with the entry layer and continuing consecutively with each higher layer until found, check for the existence of external edges from the exit layer. The first layer to which either external edge exists is the transition layer. A *closest* transition begins with any interlayer edge between the exit and transition layers. An optimal transition begins with any such closest interlayer edge except, unless no other closest choice exists, one that uses an external vertex as the transition vertex. This leaves the external vertices to be considered for a subsequent optimal transition from the transition layer. The choice of the second edge for a transition, if needed, is trivial because the transition vertex is adjacent to each vertex of the entry layer.

5 Existence of Layered Hamiltonian Cycles

In this section we prove that a permutation graph is Hamiltonian if and only if it admits a LAYERED HAMILTONIAN CYCLE. The proof involves partitioning an arbitrary Hamiltonian cycle into two paths, each of which has one end at the highest layer and the other at the lowest layer. If these two paths are layered, then we are done. Otherwise we will show that the two ends of each path can in fact be ends of a layered path. Thus all that remains is to reconstruct the partially layered paths into a fully layered paths. The following lemma provides a construction by which a partially layered path can be reconstructed into fully layered path, thus constituting a LAYERED HAMILTONIAN CYCLE. The process is iterative, each

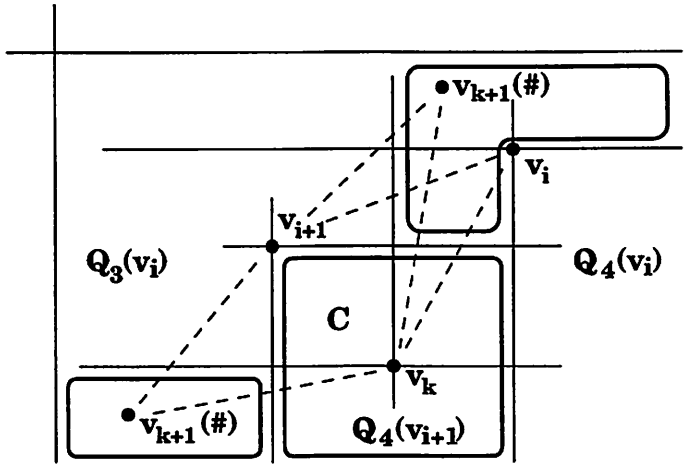


Figure 6: Illustration of Reversing Lemma. Vertex Configuration allows Segment Reversal. Vertex v_{k+1} may lie in either of the areas marked by (#).

iteration reversing an identifiable subpath so that progressively more vertices are in a layered position. Note that each path can be considered as a Hamiltonian path in the subgraph induced by its vertices.

Lemma 5.1 (Reversing Lemma). *[18] Let G be a permutation graph and $P = [v_1, v_2, \ldots, v_n]$ be a Hamiltonian path such that the subpath $P_{1,i}$, $1 \leq i \leq n$, is the initial layered subpath of maximum length in P. Then there exists a layered Hamiltonian path in G beginning with $P_{1,i}$.*

Discussion of Lemma 5.1. By `initial layered subpath' we mean a subpath that is layered in the subgraph induced by its vertices, and that each vertex not in the subpath is higher than or adjacent to the vertices in the subpath, i.e. vertices not in the subpath could follow the subpath in a fully layered path.

The proof consists of showing that edges exist in G which allow the subpath $P_{i+1,k}$ to be reversed in P, where v_k is the last vertex in P lower than v_{i+1}. Specifically, the edges v_iv_{i+1} and v_kv_{k+1} (assuming $k < n$) which bind the subpath $P_{i+1,k}$ into P are abandoned, and the edges v_iv_k and $v_{i+1}v_{k+1}$ are shown to exist and are used to attach the reversed subpath $P_{k,i+1}$ back into P. P becomes layered by iterating this procedure a finite number of times. Here we give an outline of the proof that is presented in detail in [18].

Proof. Proof of Lemma 5.1. If $i = n$ then the entire path is layered and there

is nothing to prove. Therefore assume $i < n$. This means that there is a non-empty set of vertices C following v_{i+1} in P which are lower than v_{i+1}. Let $v_k \in C$ be the last vertex in P which is lower than v_{i+1}. Figure 6 illustrates this configuration in a Euclidean representation.

Vertices v_i and v_{i+1} are adjacent and thus lie on a positively sloped line. The set C and in particular vertex v_k are pictured lower than v_{i+1} (i.e. in $Q_4(v_{i+1})$), but not lower than v_i (i.e. outside $Q_4(v_i)$). Vertex v_i is in the initial layered subpath, so no remaining vertices in P can be lower than it. These constraints on v_k imply that v_k lies on a positively sloped line with v_i and therefore is adjacent to v_i.

Similarly, if $k < n$ then v_k and v_{k+1} are adjacent and thus lie on a positively sloped line. Vertex v_{k+1} is not in set C, and therefore not in $Q_4(v_{i+1})$. These constraints, along with the placement of v_k to the southeast of v_{i+1}, force v_{k+1} to be on a positively sloped line with v_{i+1}, thus implying v_k is adjacent to v_{i+1}. □

The Reversing Lemma guarantees a reconstruction of a Hamiltonian path into a layered Hamiltonian path. However, the lemma applies only to partially layered paths. It remains to be shown that a careful separation of a Hamiltonian cycle into two paths will indeed result in the two paths being partially layered in the fashion required by the lemma. We define the concept of a *consistent labeling* and prove a lemma establishing this.

Let $G = (V, E)$ be a permutation graph and $\mathcal{L}$ be a permutation labeling for G. Let $\mathcal{L}'$ be a permutation labeling of the induced subgraph $G[V']$, $V' \subset V$. $\mathcal{L}'$ is said to be *consistent* with $\mathcal{L}$ if $\mathcal{L}'$ is order preserving with respect to $\mathcal{L}$. That is, $\mathcal{L}(u) > \mathcal{L}(v) \Leftrightarrow \mathcal{L}'(u) > \mathcal{L}'(v)$ for all $u, v \in V'$. The Euclidean representation for $G[V']$ with a consistent labeling can be obtained from a Euclidean representation for G by removing vertices not in V' and collapsing the representation by removing empty rows and columns. The vertices are then relabeled so that all labels are again consecutive from left to right. See Figure 7.

The following lemma provides a guarantee that top and bottom vertices in the original graph remain as top and bottom vertices if included in an induced subgraph. This implies that cutting a Hamiltonian cycle into two paths at a top vertex and at a bottom vertex will generate the required pair of partially ordered paths, assuming that the terminal vertices are cloned so that each path gets a copy.

Lemma 5.2 (Inheritance). *Let $G = (V, E)$ be a permutation graph, and $V' \subseteq V$. Given a permutation labeling $\mathcal{L} = \mathcal{L}(V)$ and a consistent labeling $\mathcal{L}' = \mathcal{L}(V')$ for $G' = G[V']$, then $\mathcal{E}(G', \mathcal{L}')$ inherits the higher (lower) relations from $\mathcal{E}(G, \mathcal{L})$, and vertices of V' that are top (bottom) vertices in $\mathcal{E}(G, \mathcal{L})$ are also top (bottom) vertices in $\mathcal{E}(G', \mathcal{L}')$.*

Proof. The proof follows from the fact that orientation (by vertex quadrant) of vertices is not affected as vertices in the Euclidean representation are removed,

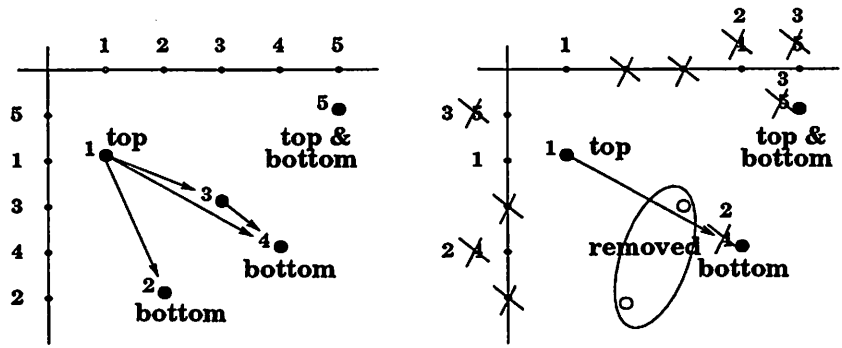


Figure 7: Illustration of the Inheritance Lemma

because only the corresponding empty rows and columns are removed. Thus the higher (lower) relation does not change even though the height of vertices may change. It follows that top (bottom) vertices in G remain so in $G[V']$. □

The preceding lemma is the basis for the main result of this section which states that a LAYERED HAMILTONIAN CYCLE can be constructed in a Hamiltonian permutation graph. The algorithm which is presented in this paper is specifically designed to generate a LAYERED HAMILTONIAN CYCLE.

Theorem 5.3 (Layered Hamiltonian Cycle Existence). *A permutation graph is Hamiltonian if and only if it admits a* LAYERED HAMILTONIAN CYCLE.

Proof. A Hamiltonian cycle must include a top vertex and a bottom vertex. Let path P_A be one path in the cycle between a top vertex and a bottom vertex, and let path P_B be the other path. Thus P_A and P_B share only the terminal vertices, a top vertex and a bottom vertex. Each path is a Hamiltonian path in the subgraph induced by its vertices. Given a Hamiltonian path in a permutation graph, there exists a layered Hamiltonian path [18]. By Lemma 5.2, the top and bottom vertices of a graph exist as top and bottom vertices in the induced subgraphs. Therefore P_A and P_B each have an initial layered subpath and Lemma 5.1 applies. By the reconstruction strategy in the proof of the lemma, the end vertices will remain as end vertices because they are already top and bottom vertices. (Neither vertex can be within a subpath being reversed.) After the reconstruction, the shared end vertices can be assigned to either of the paths without affecting the layering of the paths. A Hamiltonian cycle constructed from the two layered paths will by definition be layered. □

6 Construction of Layered Hamiltonian Cycles

Individually, the two HC-paths of a LAYERED HAMILTONIAN CYCLE might not have vertices from each layer of the graph. Together, the two sets of transitions, i.e. the two path frameworks, representing the paths must be constructed so as to incorporate vertices representing all the layers and to contain those vertices in a cycle. Furthermore, the transitions must be constructed in such a way that all the remaining vertices of the graph can be inserted into that cycle. These sets then constitute the CYCLE FRAMEWORK.

We first consider construction of a path framework for a layered Hamiltonian path. As detailed in [18], the layered Hamiltonian path construction begins with the identification of a path framework, and if one can not be found, then the graph is declared to be not traceable. The path framework is constructed by using a set of greedy rules to find a complete set of transitions, i.e. one transition spanning each consecutive pair of layers of a highest layering. The greedy rules are based on the results in Section 4. The transitions must be disjoint so that the Hamiltonian path can be completed by arbitrarily inserting any unused vertices of a layer between the entering and exiting transitions for the layer. Alternatively the path framework can be seen as a path intersecting all the layers in such a way that the remaining vertices of a layer can be inserted into it at the intersection.

The proof of Theorem 5.3 presumes an existing cycle (not necessarily layered) and thus also the two vertex partitions for the two constituent HC-paths. Given the partitions, the construction problem for a LAYERED HAMILTONIAN CYCLE is reduced to two independent Hamiltonian path constructions. However, we do not know of a method to precompute the partitions. Instead we consider and prove the correctness of a joint construction of the two path frameworks, partitioning the graph layer by layer as work proceeds. The transitions are generated in an optimal manner starting with the lowest layer such that the choice of vertices for each transition has the least negative effect on constructing subsequent transitions for either path, thus maximizing the potential of a Hamiltonian cycle to be constructed. The algorithm is therefore a greedy algorithm.

6.1 Cycle Frameworks

As indicated above, in the non-partitioned view of the graph, the two path frameworks of the CYCLE FRAMEWORK might not appear as they would for Hamiltonian paths. Some layers may be assigned entirely to one framework, thus giving the appearance of being missed by the other framework. Other layers may have single vertices assigned to one of the path frameworks, thus giving the appearance of having entering and exiting transitions intersecting at those vertices. This is not possible for a layered path in a highest layering. However, in the subgraph induced by that path framework, portions of two layers may merge as one. In still other layers the vertices may be apportioned between the two path frameworks.

Examples of a CYCLE FRAMEWORK and its constituent path frameworks in their induced subgraphs are shown in Figures 6.1, 6.1, and 6.1 respectively. We begin this section with some discussion and terminology that expands the notions of *transitions* and *attachability* to accommodate these perspectives.

An intermediate layer on which the entering and exiting transitions for one of the paths appear to share a vertex is said to be *weakly attached* to that path. No other vertices of the layer can be inserted at that intersection. On the other hand, a layer is said to be *strongly attached* to a path if the entering and exiting transitions are disjoint. This is required for all intermediate layers of a Hamiltonian path framework. However, it is sufficient for a CYCLE FRAMEWORK that each intermediate layer be strongly attached to one of the paths and weakly attached or even not attached at all to the other path. Moreover, for a two vertex layer, it is sufficient that the layer be only weakly attached to both sets.

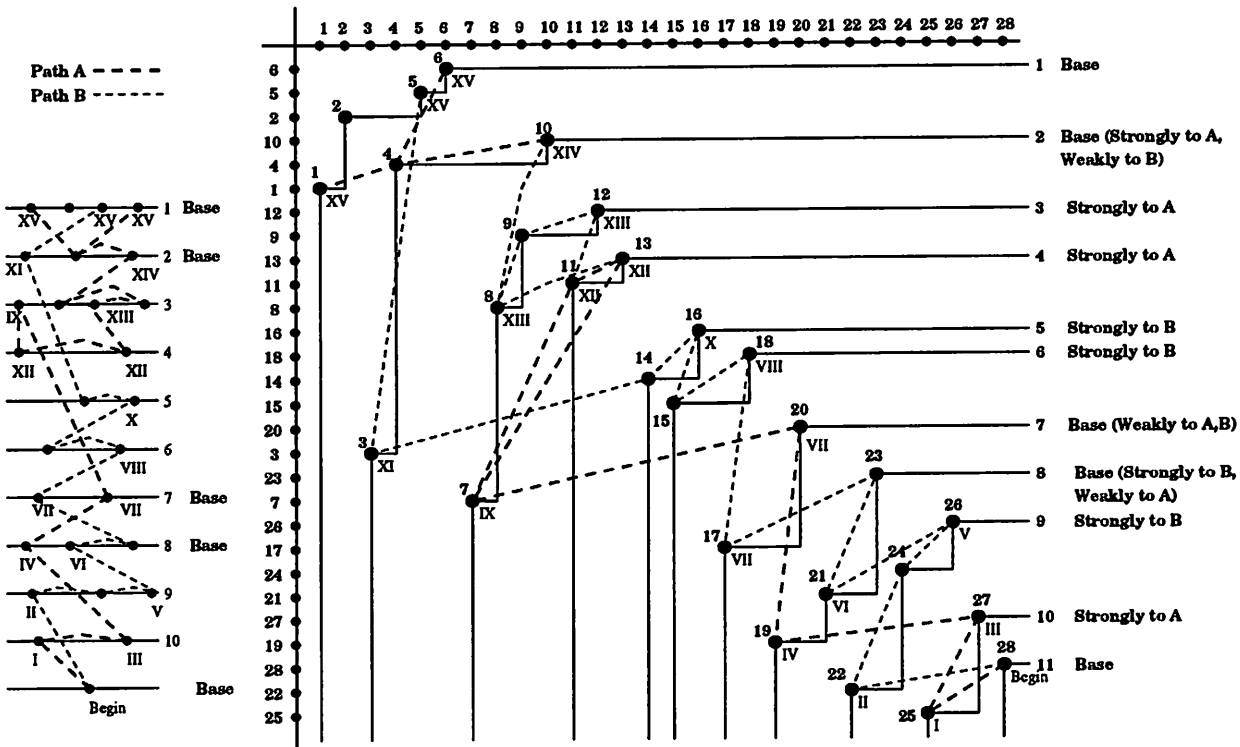
The conditions for attachment by the lowest layer and the highest layer are somewhat different. This is because the two sets of transitions only exit or enter them respectively. A lowest layer consisting of a single vertex may use that vertex as the exit vertex for both transitions. This may appear to be an exception to the condition that the two sets of transitions in a CYCLE FRAMEWORK are disjoint, however it is easily seen not to conflict with the conditions for Theorem 5.3 by considering that lone vertex as being cloned, one copy per path. It may be noted that the highest layer must have at least two vertices for the graph to be connected. This is easily seen in a Euclidean representation in which a single vertex in the highest layer implies that all other vertices are in quadrant 4 of that vertex. The lowest layer and the highest layer are considered to be strongly attached to both sets of transitions if either 1) the pair of exiting or entering transitions (respectively) are disjoint, or 2) the pair of exiting transitions (for the former) share the singleton vertex of the layer.

6.2 Greedy Algorithm for Cycle Framework

In discussing the construction of the framework for a Hamiltonian cycle, it is helpful to be able to distinguish certain layers based on their attachments to the two paths, and the paths themselves. We present them here:

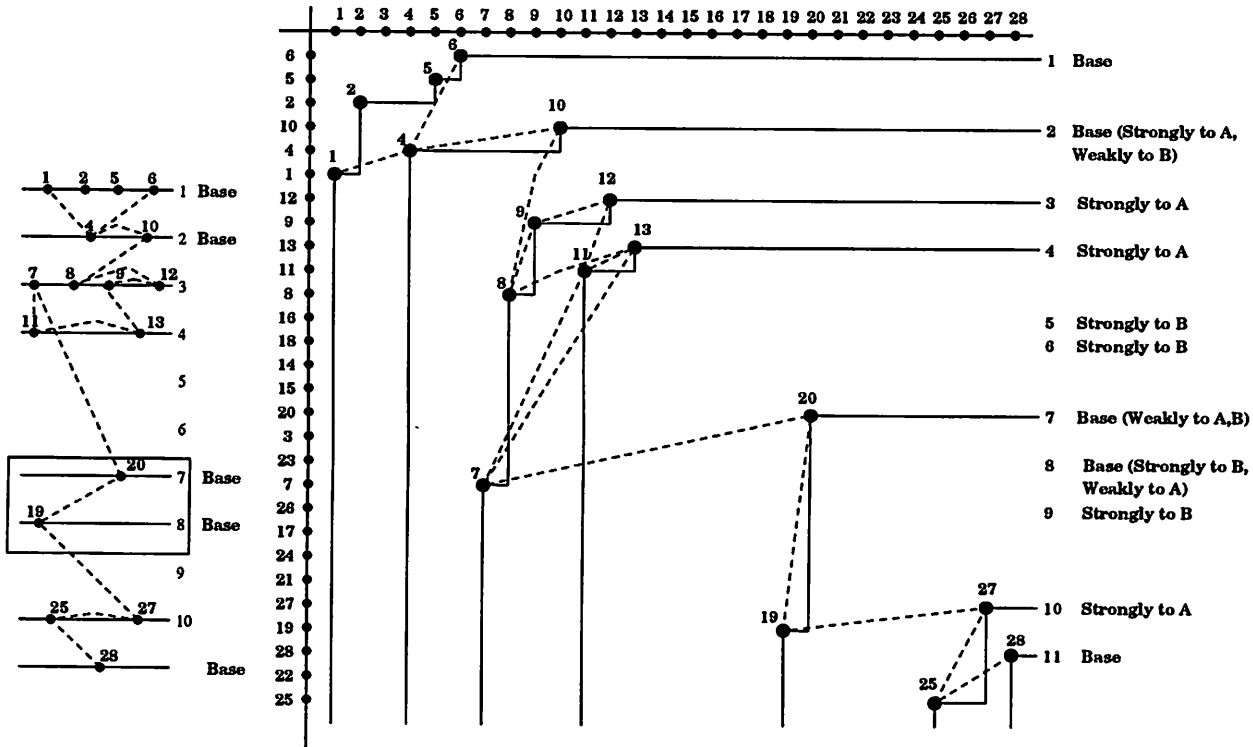
Base Layer. A layer that is at least weakly attached to both paths is called a *base layer*. It is easy to see that in a complete CYCLE FRAMEWORK the lowest layer and highest layer must be base layers. There may be additional base layers in the graph. The CYCLE FRAMEWORK can be pictured as a series of individual cycles between consecutive base layers through the intervening sequences of non-base layers. The cycles intersect at base layers in such a way that one cycle through all the layers can be constructed.

Potential Base Layer. At certain stages of construction, it is possible to tentatively identify a layer as the next higher base layer. Construction proceeds



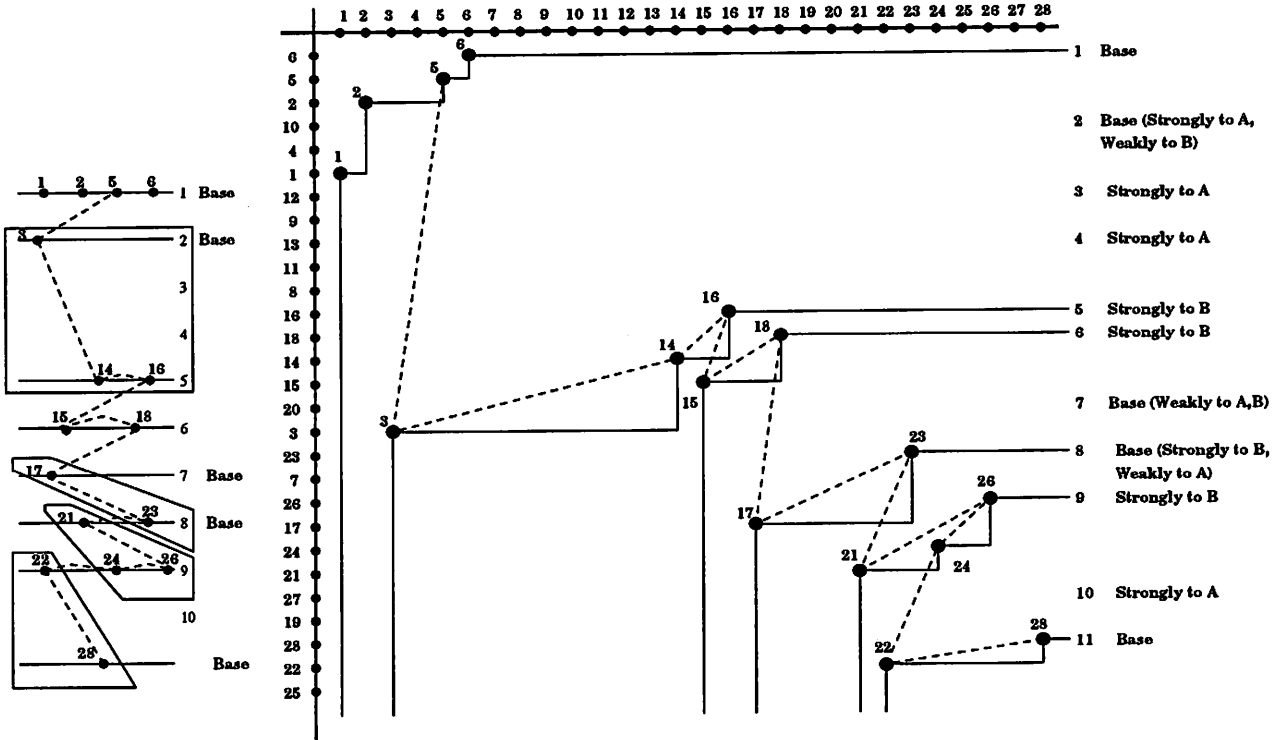
LEGEND: Both flattened and standard views of layers given. Construction sequence is indicated by Roman numerals.

Figure 6.8: Example of a Cycle Framework



LEGEND: Vertices of layers 7 and 8 merge into one layer as shown by the box.

Figure 6.9: Path Framework A of the Cycle Framework



LEGEND: Some layers are reconfigured as indicated by boxing.

Figure 6.10: Path Framework B of the Cycle Framework

with the assumption that this *potential base layer* will become a base layer. This imposes constraints to the selection of subsequent transition vertices, because two exiting transitions rather than one are needed from a base layer.

Working Layer. Construction is always layer by layer from the lowest to the highest. The *working layer* represents the layer from which a transition(s) is currently being sought. The working layer and all lower layers have been attached to at least one of the paths, and no higher layer has yet been attached to either path.

Leading Path. The *leading path* refers to the set of transitions which has a transition vertex from the highest layer yet reached, i.e. a potential base layer. If both sets have transition vertices from the highest layer yet reached, then the first set (if there is a first) to get such a vertex is the leading path. Otherwise neither set represents a leading path. This can occur if the working layer is a base layer. The potential base layer is tentatively attached to the leading path.

Trailing Path. If one set of transitions represents the leading path, the other set represents the *trailing path.* The objective is to attach each layer up to and including the potential base layer to the trailing path, thus establishing the potential base layer as truly a base layer.

In the greedy algorithm, a CYCLE FRAMEWORK is constructed layer by layer, identifying one optimal transition for each non-base layer and two optimal transitions, one per HC-path, for each base layer. In some cases the optimality can not be determined while the layer is being processed. In these instances we show that candidate vertices for optimal transitions can be marked as reserved, and processing continues. Furthermore, we demonstrate that this method does not result in a combinatorial explosion of backtracking for resolving the reserved choices.

6.2.1 General Discussion

During construction, a newly identified highest transition layer becomes a *potential base layer.* The path corresponding to this transition becomes the *leading path* and its construction is interrupted at the transition vertex in the potential base layer. Construction then proceeds for the other path which now is the *trailing path.* If both paths simultaneously reach a new highest transition layer (this is possible if the working layer is a base layer), thus jointly identifying a potential base layer, then the identification of leading and trailing path is arbitrary.

If a new higher potential base layer is discovered by using a non-base layer transition, the next step is to complete the last transition of the (former) leading path with an entry vertex in the next working layer. Note that the exit and entry layers for this transition are not consecutive as they would have been for a

Hamiltonian path framework. In an implementation only the entry layer and not the entry vertex needs to be recorded, because the choice of an entry vertex in the layer is arbitrary and can be deferred.

The potential base layer will become a *base* layer, and the transition vertex of the interrupted transition becomes an entry vertex if all lower layers have been processed, (i.e. attached to the trailing path,) without a higher transition layer being needed. Otherwise a transition for the trailing path is identified with a still higher transition layer which becomes the potential base layer, and then that path becomes the leading path, and the process continues in a leap frog fashion until a new base layer is finally established. Thus processing alternates on the sets of transitions for the two paths, always extending the set of transitions for the trailing path.

The general picture begins with a base layer with two exiting transitions, the higher to a potential base; then a sequence of layers strongly attached to one path; then a new potential base when a transition from that path "jumps" the original potential base; then a sequence of layers strongly attached to the other path; then still another new potential base, etc. until eventually a sequence of layers all the way up to the latest potential base is strongly attached with no higher jump, thereby identifying a new base layer. The entry layer for each transition is the layer above the exit layer except for a transition that identifies a new potential base, in which case a new sequence is begun in the alternating pattern.

An example in a flattened view of the Euclidean representation is given in Figure 8. The two HC-paths are distinguished by solid and dashed lines. Only the transitions identifying new potential bases are drawn. Each numbered group indicates a sequence of working layers attached to one PATH between the "leap frog" jumps. All transition vertices are indicated by group number.

Because layers are processed in sequence, all layers up to the working layer are fully processed except possibly some having vertices which were reserved when the optimality of transitions could not be immediately determined. If at some point there is no transition(s) to advance the working layer, the algorithm declares failure, and we prove in Section 7 that the graph is not Hamiltonian.

As long as the working layer is not a base layer, a transition is sought for the trailing path from the working layer's external vertices via (ideally) non-external transition layer vertices so as to attach each new working layer. This is in accordance with the strategy presented in Section 4. If the potential base layer is used as a transition layer for one of these transitions, then some extra care is needed in the selection of the transition vertex, because it may be that the potential base layer will become a base layer for which two exiting transitions are needed.

When the working layer is a base layer, two transitions must be found, one for each HC-path. A number of cases exist for choosing the optimal pair of transitions, a situation more complicated than for a single transition. The optimal exit vertices are end vertices of the layer, either both from the left end, the right end, or one from each end. Because the working layer only needs to be strongly at-

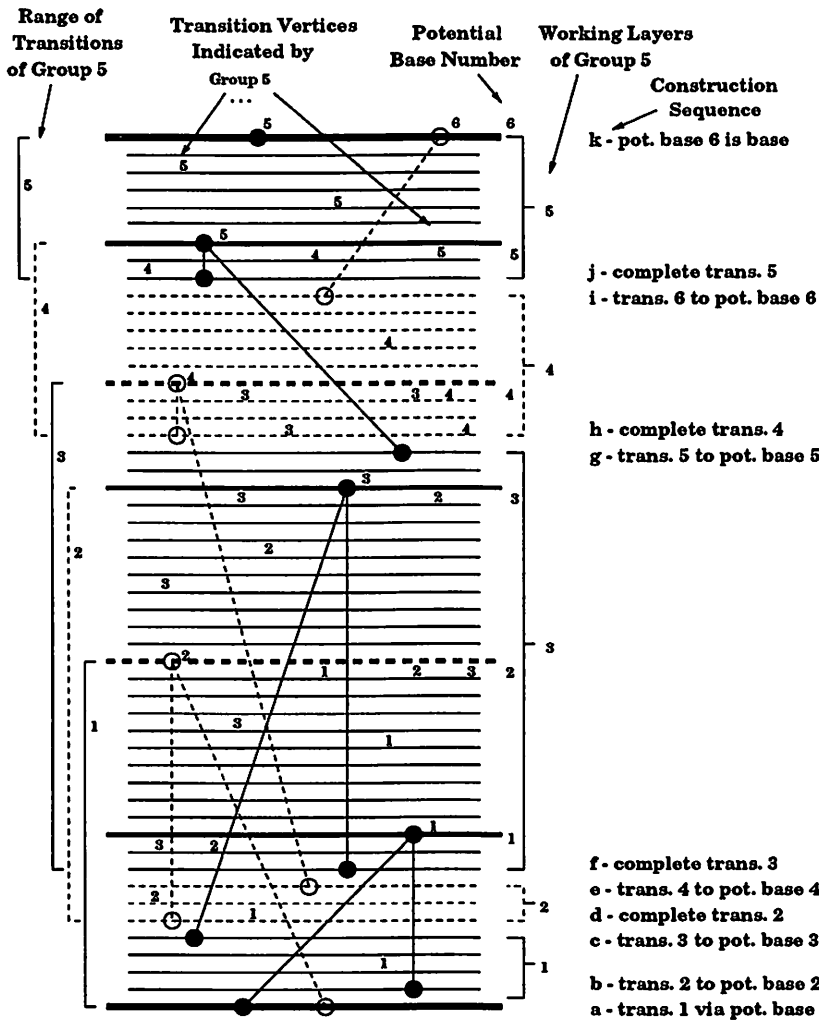


Figure 8: Alternating Designation of Leading Path between Two Base Layers

tached to one of the paths, it is sufficient for it to be weakly attached to the other path, i.e. an entering and an exiting transition may intersect. We prove that the path frameworks can be reconstructed so that an original external vertex of a base layer, which may have been claimed as a transition vertex, can instead become the entry vertex for a weak attachment. The entry for the trailing path is optimally selected (or reserved) after all other transitions using this layer as a transition layer have been constructed. Therefore, this reconstruction can be applied at most once on a layer, resulting in reassignment of the entry status from the leading path. Therefore, if both external vertices were claimed, only one can be used as an exit vertex. See Figure 9 for illustrations of the reconstruction.

Lemma 6.1. *A partial* CYCLE FRAMEWORK *can be reconstructed so that an external transition vertex in a base layer can be made into an entry vertex.*

Proof. Assume an external vertex in a base layer, say α^o, is currently used as a transition vertex for a transition to layer I, and this vertex is desired as an entry vertex. There are two cases to consider.

1. There are two vertices in the base layer which belong to transitions from a common (preceding base) layer, each representing one of the paths. One of these transitions has no second edge, i.e. its path was treated as the leading path. While the transition for this leading path does not need a second edge (because the other path was attached to all the intermediate layers), there certainly exist edges to potentially extend the transition to any vertex of intermediate layer I. Now the transition using α^o can be truncated so that α^o becomes an entry vertex, and the path back down to layer I be made instead by using any of those edges which can extend the last transition of the leading path.

2. There is one vertex, an entry vertex of a transition, in the base layer which belongs to the leading path. This is a single edge transition. By the algorithm, layer I must be higher than the exit layer for this transition. The only vertices of the intermediate layers not considered in the construction of this transition are transition vertices of previous transitions, and none of these vertices therefore can be the the entry vertex of a transition, in particular of the transition using α^o. By the properties of a transition, there certainly exist edges to potentially extend the single edge transition to any of the non-transition vertices of intermediate layer I. The reconstruction of Case 1 can be applied in this case also.

In the second case it was important to consider the `scope' of the single edge transition because it is possible that there is no edge from the entry vertex down to a vertex outside the scope. In both cases there may be multiple transition vertices for the trailing path, but by construction only one for the leading path. □

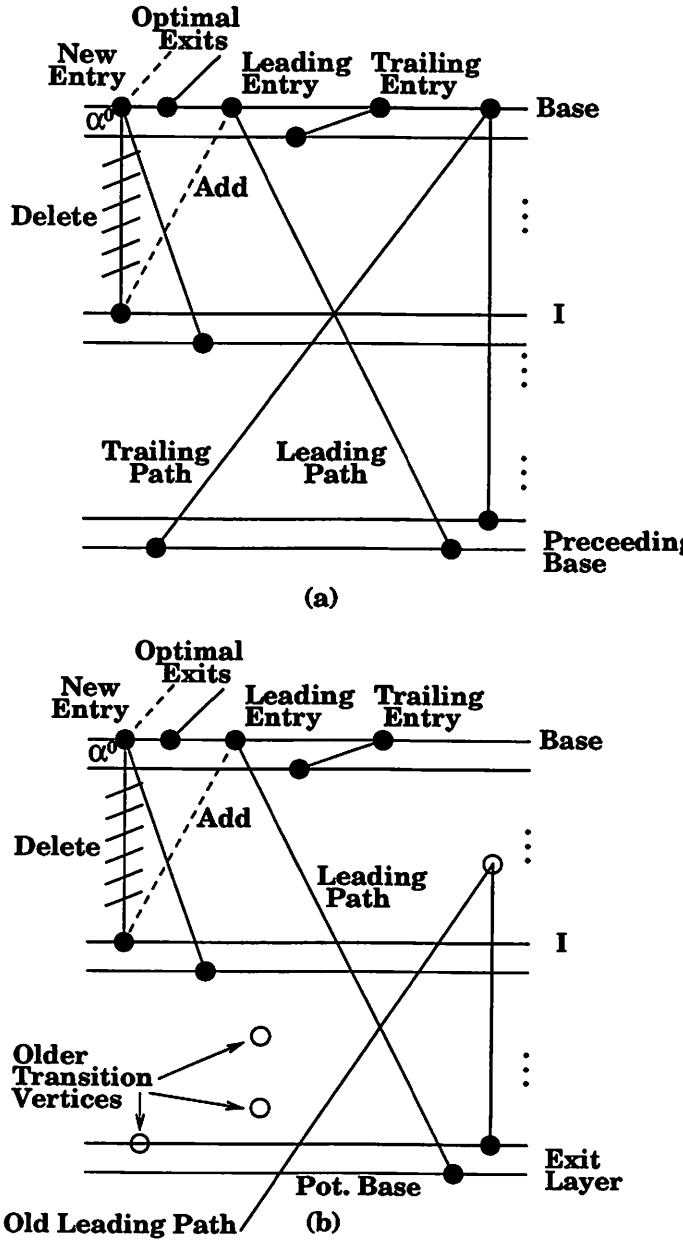
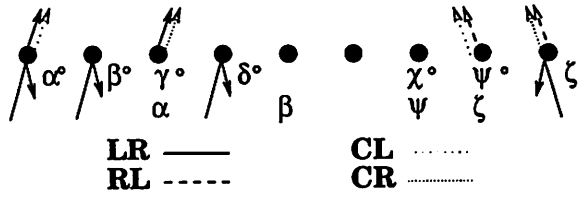
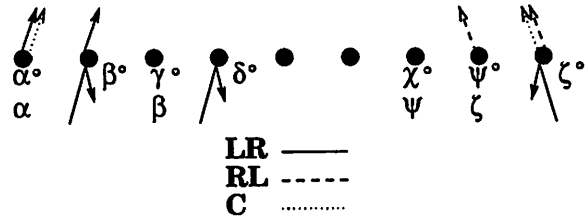


Figure 9: Reconstruction of Path Frameworks



(a) Both External Vertices Claimed



(b) At Most One External Vertex Claimed

Figure 10: Optimal Base Layer Exit Pairs

To be precise, the optimal exit vertices from a base layer are one of the following pairs. The labels for the pairs are indicative of their appearance in a Euclidean representation. For reference in this more complicated situation, the original α, β, etc. vertices will be indicated by superscripting them as α^o, β^o, etc. Illustrative examples are given in Figure 10.

Optimal Exit Vertices for Base Layers:

- **LR** – The pair α^o, β^o if $\alpha = \alpha^o$, else the pair α^o, α.
- **RL** – The pair ζ^o, ψ^o if $\zeta = \zeta^o$, else the pair ζ^o, ζ.
- **C** – The pair α^o, ζ^o if either $\alpha = \alpha^o$ or $\zeta = \zeta^o$, else one of these pairs:
  - **CL** – The pair α^o, ζ or
  - **CR** – The pair α, ζ^o.

The last two possibilities demonstrate the fact that the choices of two exiting transitions may not be independent, even though the vertex choices for each transition are disjoint. Therefore a search for two transitions should always be a search for a *viable* pair of transitions. Nevertheless, we show that at most four vertices in the base layer (specified above) need to be checked for possible transition edges to at most six vertices, α, β, γ, and χ, ψ, ζ, in the transition layer, the worst case occurring when both transitions use the same transition layer.

If possible, transition or entry vertices in the new potential base layer should be chosen such that all three or four potentially optimal pairs are available as exit vertices. In practice this means avoiding the external vertices α and ζ as transition vertices. Thus single transitions may optimally use β or ψ, and double transitions may optimally use β and γ, β and ψ, or χ and ψ. Although vertex α or ζ of a base layer can be reused after reconstruction, it is not always known in advance that the potential base layer will become a base layer.

As the case by case analysis of the base and non-base layer transitions given in the next two subsections show, there is no combinatorial explosion of candidate transitions when optimality can not be immediately determined. The choices for each layer can always be narrowed to a pair of vertices per path which can be reserved until a selection is forced at a later stage or the selection is found to be arbitrary. This is true even when optimality for immediately lower layers remains unresolved.

6.2.2 Non-Base Layer Transition Cases

There is a single transition to be found from each non-base layer. The closest transition will have a transition vertex in a layer that is either 1) below the potential base layer, 2) in the potential base layer, or 3) above the potential base layer. These three cases are discussed in this section.

CASE 1: A Transition below a Potential Base. This case is comparable to finding the optimal transition for a Hamiltonian path, as only one path is involved and no layer can be skipped as a working layer. Three greedy rules were proposed in [18]. Given an exit layer X and the closest transition layer T, the following three rules are examined in order until a rule applies. A subscript indicates the layer of the vertex.

Transition Greedy Rules:

A. If either $\alpha_x\psi_t \in E$ or $\zeta_x\beta_t \in E$, then choose that edge.

B. If both $\alpha_x\zeta_t$, $\zeta_x\alpha_t \in E$ and

if ζ_t and α_t are not currently reserved,
then reserve both, else

if ζ_t and α_t are currently reserved,

then resolve the reservation by selecting either of the edges for the reserved transition and the remaining edge for the current transition.

C. Only $\alpha_x\zeta_t \in E$ or $\zeta_x\alpha_t \in E$, so choose that edge for the transition.

The justification for these greedy rules is as follows: Some external vertex will provide the optimal exit for a later transition from the transition layer. Choosing any other vertex as the transition vertex preserves the external vertices. Increasing accessibility to the transition layer as the working layer rises guarantees that any other choice of transition vertex is equivalent in terms of admitting non-external vertex choices for later transitions. If an external vertex must be used as the transition vertex, it is better, if possible, to reserve both until it can be later determined which is the optimal exit.

Note that the Greek alphabet labeling is reapplied to the transition layer vertices remaining after a transition vertex is definitively identified (not just reserved). If the exit vertex of this transition was reserved, then possibly a chain of earlier reserved transitions can be resolved.

CASE 2: A Transition to a Potential Base. This case is very similar to the previous case. The only difference is that the transition layer may later become a base layer from which two exit vertices must be found. The greedy rules of Case 1 can be used for this case also, because one claimed vertex can be recast if needed as an entry vertex by reconstructing a pair of transitions. Avoidance of the external vertices in the transition layer is indicated if the potential base is not known to be a base layer, and does not add to the complexity in any case.

CASE 3: A Transition Identifying a New Potential Base. The optimal transition selection is the same as in the first case, again because the potential base might not become a base layer. Assuming it actually becomes a base layer, this transition will remain a one edge transition and the transition vertex become an entry vertex.

6.2.3 Base Layer Transition Cases

The cases for optimal transition construction from a base layer are distinguished by six factors involving sizes, commonality, and presence of a base layer for the transition layer(s); size of and previous vertex use for the base layer; and transition orientations.

Distinguishing Features for Cases of Base Layer Exits:

1. **The number of vertices in the closest transition layer.** *The closest transition layer may have 2, 3, or more than 3 vertices, and is classified as type $T2$, $T3$, and TM respectively. Exiting the transition layer from two vertices when there are no more than three in the layer inevitably means using at least one vertex from each of the rightmost pair, the leftmost pair, and the external pair.*

2. **The commonality of transition layers of the two closest viable transitions.** *The two closest viable transitions may or may not have the same transition layer, and are classified as type S and T respectively. When the two closest transitions have the same transition layer, care must be taken that the two transition vertices chosen do not impair the optimality of exiting transitions from what will be the next potential base layer.*

3. **The orientation of the closest possible viable transitions.** *The orientation of closest possible viable transitions may be left to right and/or right to left. If both orientations are possible, the number of combinations for choosing transitions is higher.*

4. **Whether the vertices α^o and ζ^o of the base layer are both claimed.** *The vertices α^o and ζ^o of the base layer may both have been claimed as transition vertices by lower transitions. Base layers are classified as type NC if no more than one is claimed or as type CC if both are claimed. One vertex being claimed does not interfere with its dual use as an entry and exit vertex, because in this case the layer can still be strongly attached to the other path. If its use is as a transition vertex, the framework can be be reconstructed so that it is an entry vertex.*

5. **The number of vertices in the base layer.** *The number of vertices in the base layer may be 1, 2, or more than 2, and the layer is classified as type $B1$, $B2$, and BM respectively. Only the lowest base layer in a Hamiltonian graph may have a single vertex. However, any layer can have exactly two vertices. If a base layer has two vertices, then it may be that the optimality of one as the exit vertex for the closest transition can not be determined until resolution of reserves for the next closest transition (the second transition.)*

6. **If a transition layer is a base layer.** *In some configurations it can be determined that a newly discovered potential base layer must be a base layer. If this can be determined, then an external vertex can be used in certain combinations with other vertices without loss of optimality. Otherwise an exiting transition must not intersect an entering transition.*

We present a notation to express the *claim* pattern in a layer of vertices, i.e. the combination of vertices that are claimed, reserved, or unused for transitions. The

pattern determines the viability of optimal exit vertices from the layer, and is constrained by the availability of edges to the layer for use by transitions. A claimed vertex is indicated by a *c*, a reserved vertex by an *r*, and an unused vertex by an *o*. Vertices other than the external two or three vertices on either end are generally not involved in the discussion, so their possible presence is indicated simply with ellipsis (...). The leftmost vertex is the α vertex, and the rightmost is the ζ vertex. Thus *ro*...*cr* describes a layer with α and ζ as a reserved pair and ψ as claimed.

Claim Notation for Vertices	
claimed	c
reserved	r
unused	u
don't care	...

This notation is used in Figure 11 which categorizes the optimality of potential base layer claim patterns resulting from double transitions from the preceding base layer. Optimality is in terms of subsequent exiting transitions. Assuming the potential base becomes a base layer, a first choice is one that admits subsequent pairs of exits using any of the potentially optimal combinations of external vertices. A second choice is one that disallows reuse of one of those vertices. There is no ambiguity for the second choice because a choice of one transition vertex can always be reused. Assuming the potential base does not become a base layer, a first choice is one that admits a subsequent exit using either external vertex; a second choice disallows use of one of the two external vertices; and a third choice disallows use of some outer pair of the four external vertices, i.e. α and β, α and ζ, ψ and ζ. Additional optimal claim patterns are possible, however, a close examination of Figure 13 will reveal that this set is sufficient.

We also present a notation to express the *accessibility* (presence of edges) of vertices in a layer for use in transitions from a given lower layer. (See Figure 12 for an illustration.) Certain patterns may constrain transitions to use some of the external vertices that would be optimal as later exit vertices. The notation is at most a four digit number, the digits from left to right designating the availability of transition vertices to the ψ, ζ, α, and β exit vertices respectively. The first two digits count positions from the left of the transition layer, and the second two digits count positions from the right. A 1 indicates no vertex, a 2 indicates the external vertex only, etc. Because any more than three vertices yields no increased optimality, patterns with digits above 4 are not considered. For example, the accessibility for the example in Figure 12 is 1432.

Figure 13 is a table of all patterns (excepting those that are mirror images) that admit a pair of viable transitions from a lower (base) layer. The optimal claim pattern for a given accessibility pattern and base and transition layer characteristics

CATEGORY	FIRST CHOICE	SECOND CHOICE	THIRD CHOICE
Three vertex transition layer becomes a base layer.	crr rcr rrc	cco coc occ	
Three vertex transition layer does not become a base layer.	rcr	crr rrc	cco coc occ
Multi-vertex transition layer becomes a base layer.	co...co oc...oc coc...o o...coc cr...or ro...rc	cc...oo oo...cc co...oc	
Multi-vertex transition layer does not become a base layer.	occ...oo oo...cco oc...co rc...or ro...cr	co...co oc...oc coc...oo oo...coc cr...or ro...rc	cc...oo oo...cc co...oc

Figure 11: Optimal Double Claims

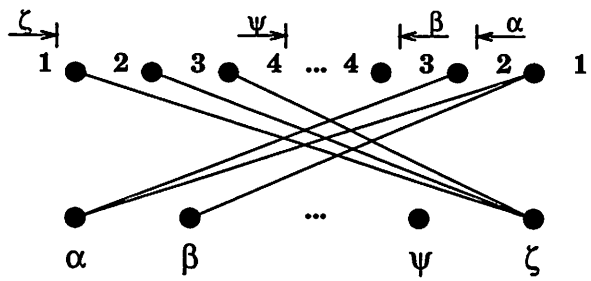


Figure 12: Illustration of Notation for Accessibility

is specified, along with a sketch of the optimal transition configuration.

The cases for pairs of exits to different layers is simpler, and are listed in Figure 14 for the closer of the two transitions. The rules, which upon examination are not ambiguous, are to avoid the external vertices of both the base layer and closest transition layer. If this is not possible, then try to reserve the external edges. One unique case is for a base layer of two vertices with an edge from both. The edges should be reserved until it can be determined which provides the optimal longer transition.

The requirement of finding the closest transitions is the same for two transitions as for a single transition. We show that the relation "closest" is well defined for pairs of transitions. If the two closest viable transitions are independently chosen, then clearly no other pair could be considered to be closer. However, considering that the choice of two transitions from the same layer may not be independent, it is possible that the choice of a first closest transition A precludes the choice of the second closest transition $B1$ and thus forces a higher transition C as the second transition of a viable pair. However, selection of an intermediate transition $B2$ between A and C as the first transition invariably forces the second closest viable transition to be no closer than C. Thus "closest" is well defined.

Lemma 6.2. *The closest pair of transitions from a base layer is well defined.*

Proof. Some combination of two vertices from four specified vertices in the base layer will provide the closest transitions. If any two vertices can be used, then the proof follows. Assume the external vertices are claimed so that either one but not both can be used for an exiting transition. If either type LR or type RL exits provide an optimal pair, then again the proof follows. This leaves either type CR or type CL. The only way a problem can arise is by having four possible transition layers, two represented by α vertices and two by ζ vertices to be the transition vertices. However, any arrangement by which these two types yield pairs of transitions in which one pair takes the lowest and highest of the four and the other pair takes the middle two of the pair also yields either an optimal type RL or type LR exit pair. This can quickly be seen by checking each possible combination. Therefore the closest pair of transitions is well defined. □

The lemmas of Section 3.2 pertaining to transitions apply to closest transitions. In the cases of pairs of transitions from a base layer which reach different transition layers, the question arises as to whether the second closest transition has similar characteristics. Namely, is there a possible second edge to each vertex of each intermediate layer, with the possible allowable exception of the transition vertex of the closest transition? By showing this is the case, it is established that the algorithm can be executed. In Figure 15 we utilize the Euclidean representation to clearly demonstrate this result.

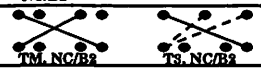
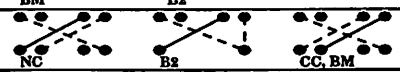
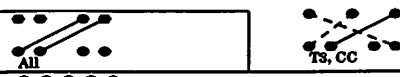
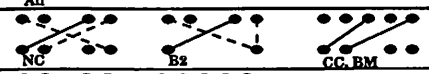
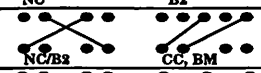
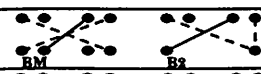
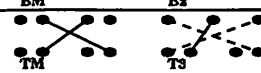
ACCESS	OPTIMUM	CONDITIONS	PATTERNS
1221	co...oc	NC/B2	
1231, 1241	co...co crr	NC/B2 T3,NC/B2	
1331, 1341, 1441	oc...co rcr	NC/B2 T3,NC/B2	
1222, 2222	co...oc		
1232	ro...cr ro...rc	NC/B2 CC	
1332	oc...co oc...oc rcr rrc	TM,NC/B2 TM,CC T3,NC T3,CC	
1132, 1133	oo...cc		
1142	oo...coc		
1242	ro...cr oo...coc	NC/B2 CC	
1342, 1442	oc...co oo...coc	NC/B2 CC	
2231, 2241	co...co crr	TM T3	
2341	oc...co co...co	NC/B2 CC	
1143, 1144, 1243, 1244, 2243, 2244, 1343, 1344, 2343, 2344, 3343, 3344, 1443, 1444, 2443, 2444, 3443, 3444, 4444	oo...cco		
1233, 2233	ro...cr		
2232, 2242, 2332, 2342, 2442	ro...cr		
1333, 2333, 3333	oc...co rcr	TM T3	
3341, 3342	oc...co		

Figure 13: Optimal Pairs of Transitions from a Base Layer

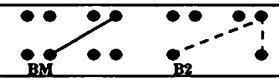
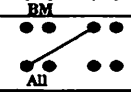
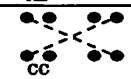
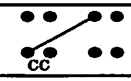
ACCESS	OPTIMUM	CONDITIONS	PATTERNS
1121	oo...oc		
1122	oo...oc		
1131 1141	oo...co		
1221	ro...or	CC	
1231 1241	oo...co	CC	
1331 1341 1441	oo...co		

Figure 14: Optimal Single Transition from a Base Layer

7 Correctness and Complexity of Algorithm

Theorem 7.1. *The greedy layered Hamiltonian cycle algorithm correctly constructs a Hamiltonian cycle in a Hamiltonian permutation graph in $O(n^2)$ time.*

Proof. By design, each HC-path is layered and is optimally constructed within its vertex partition. By Theorem 5.3 there exists a LAYERED HAMILTONIAN CYCLE in a Hamiltonian permutation graph. By the lemmas in Section 3.2, an optimal set of transitions is obtained by choosing closest transitions according to the set of greedy rules. From Lemma 6.2 it is established that well defined closest pairs of transitions can be found from a base layer, assuming of course that there exist at least two transitions. The cases discussed above cover all possible exit combinations from both base and non-base layers. Therefore a LAYERED HAMILTONIAN CYCLE is constructed in an optimal fashion, and the result follows.

The $O(n^2)$ time complexity arises from the search for transitions. From each of at most $O(n)$ layers, at most 2 transitions must be determined. A transition can be found by searching the layers above the exit layer. Each layer can be searched in constant time, as was previously demonstrated. Thus a transition can be found in $O(n)$ time, and the complete set of transitions (the CYCLE FRAMEWORK) can be constructed in $O(n^2)$ time. Finding a highest layering can be performed by extracting decreasing subsequences from a defining permutation in a greedy fash-

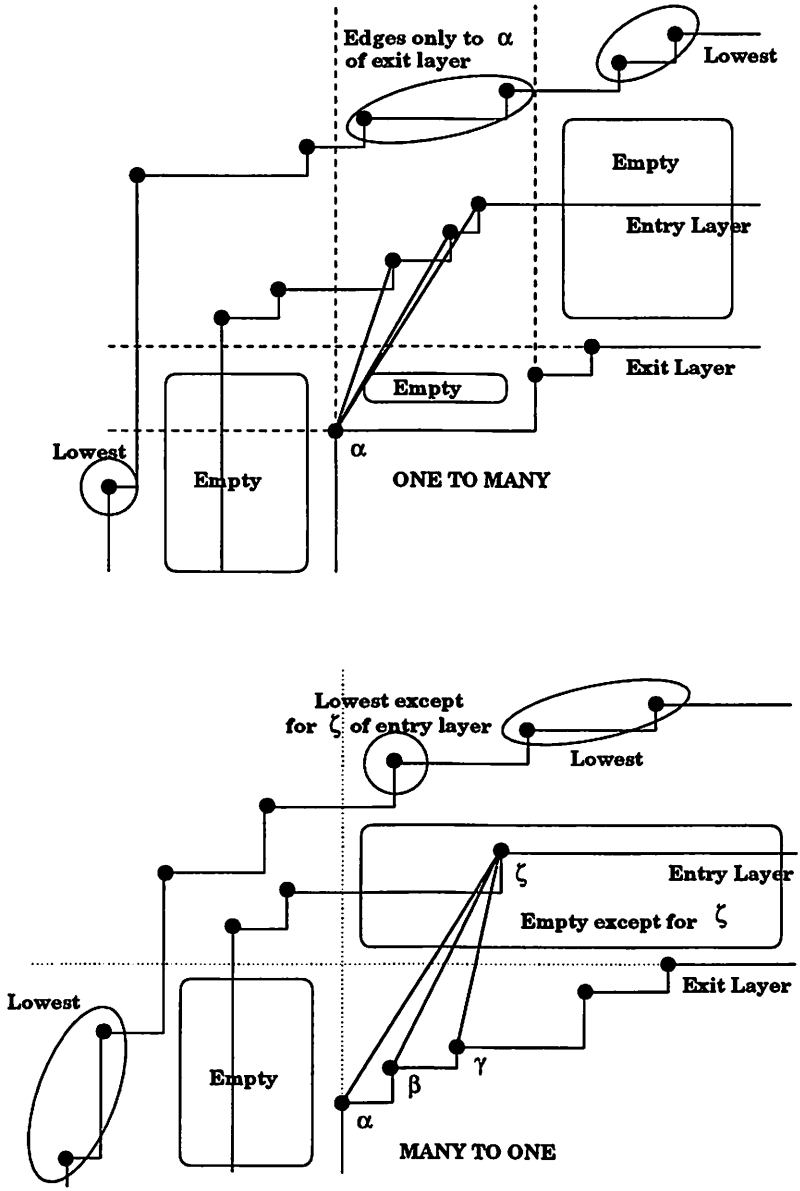


Figure 15: Second Closest Transition is a Transition

ion. This can be performed in $O(n \log n)$ time. Inserting the remaining vertices into the CYCLE FRAMEWORK is trivial, involving $O(n)$ time. □

8 Example

The example being presented is sufficiently long to represent a reasonable sampling of cases. An illustration of the CYCLE FRAMEWORK is given in Figure 6.1 with separate illustrations of the constituent path frameworks in Figures 6.1 and 6.1. "Flattened" views of the layers are shown to the left of each Euclidean representation. The layering for the constituent path frameworks are shown boxed where they differ from the CYCLE FRAMEWORK. It can easily be determined that the path frameworks are indeed layered, and thus the CYCLE FRAMEWORK also.

Let $G(\pi)$ be a permutation graph with $\pi = [6, 5, 2, 10, 4, 1, 12, 9, 13, 11, 8, 16, 18, 14, 15, 20, 3, 23, 7, 26, 17, 24, 21, 27, 19, 28, 22, 25]$. A highest layering can be found by extracting in a greedy fashion maximal decreasing sequences from π.

Working Layer: 11. The lowest layer, a base layer, consists of a single vertex. The two closest transitions are to vertex 25 in layer 10 and vertex 22 in layer 9. Case 1.1 for base layer transitions indicates the former for the trailing path (path A) and the latter for the leading path (path B).

Working Layer: 10. Construction proceeds for the trailing path A. Vertices 25 and 22 have been claimed so the closest remaining transition is from vertex 27 to vertex 19 of layer 8, a new potential base. This is taken according to Case 3 for non-base layer transitions. Path A becomes the leading path.

Working Layer: 9. Construction proceeds for the trailing path B. Several transitions are possible to layer 8. Both paths will intersect layer 8, making it a base layer. By the transition greedy rule A, the edge from external vertex 26 to non-external vertex 21 is used according to Case 2 for non-base layer transitions.

Working Layer: 8. Two closest transitions are available to layer 7, making it a base layer. Case 2.1 for base layer transitions applies, and the two transitions are from vertex 19 to vertex 20 (path A) and from vertex 23 to vertex 17 (path B).

Working Layer: 7. The two closest transitions are to vertex 18 of layer 6 (trailing path B) and vertex 7 of layer 3 (leading path A). These are taken according to Case 1.1 for base layer transitions.

Working Layer: 6. Construction proceeds for the trailing path B. The closest transition to vertex 16 of layer 5 is taken by transition greedy rule C according to Case 1 for non-base layer transitions.

Working Layer: 5. Construction continues for the trailing path B. The closest transition to vertex 3 of layer 2, identifying a new potential base layer, is taken by transition greedy rule C according to Case 3 for non-base layer transitions. Path B becomes the leading path.

Working Layer: 4. Construction proceeds for the trailing path A. The second transition edge back from vertex 7 of layer 5 is arbitrary. Both external edges are available as transitions to layer 3 vertices 9 and 12. Reserve both by transition greedy rule B according to Case 1 for non-base layer transitions.

Working Layer: 3. Construction continues for the trailing path A. An external edge from vertex 8 to vertex 10 of layer 2 is chosen by transition greedy rule C according to Case 2 for non-base layer transitions. The reserved transitions from layer 4 is forcibly resolved as vertex 11 to vertex 12. Layer 2 becomes a base layer.

Working Layer: 2. Several transitions are possible to layer 1. Case 2.3 for base layer transitions applies. Both external vertices are claimed, so only one can be used as an exit vertex. Type RL, LR, CR, and CL transitions are sought for an optimal pair. Layer 1 can be determined at this time to be a base layer, so avoidance of one external vertex is not necessary. An optimal pair of transitions from vertex 4 to vertex 5 and from vertex 10 to vertex 1.

The two path frameworks are complete, and any remaining vertices can be inserted. In this limited example only vertices 2 and 6 of the highest layer are attached. A LAYERED HAMILTONIAN CYCLE is identified as [28, 25, 27, 19, 20, 7, 13, 11, 12, 9, 8, 10, 1, 2, 6, 5, 4, 3, 14, 16, 15, 18, 17, 23, 21, 26, 24, 22, 28].

9 Summary

Permutation graphs have been studied extensively in recent years. Of three representations, we have employed the Euclidean representation to reveal structures of a permutation graph that aid in the development of efficient algorithms. We have demonstrated the power of the Euclidean representation for aiding the study of permutation graphs. With the Euclidean representation a permutation graph can partitioned into layers, each layer having a structure useful for determining interlayer edges.

It was proved that a LAYERED HAMILTONIAN CYCLE exists in a Hamiltonian permutation graph by reconstructing an arbitrary existing Hamiltonian cycle. An algorithm for constructing such a cycle is presented. Our approach consists of finding a CYCLE FRAMEWORK of interlayer edges. Vertices not part of the framework can be inserted in arbitrary sequence between the framework edges entering and exiting their layers.

We have demonstrated the power of the Euclidean representation for aiding the study of permutation graphs. The algorithm presented has an improved $O(n^2)$ time complexity. (The previous best known algorithm for the superclass of cocomparability graphs had an $O(n^3)$ time complexity.) In the process, the structure of a permutation graph has been more greatly elucidated. It is our belief that the Euclidean representation will be of significant use in further work on permutation graphs, especially for problems involving adjacency and distances.

References

[1] C. Berge, *Färbung von Graphen, deren sämliche bzw. deren ungerade Kreise starr sind*, Wiss. Z. Martin-Luther-Univ., Halle-Wittenberg Math.-Natur, (1961) Reihe, 114-115

[2] J. A. Bondy and U. S. R. Murty. *Graph Theory with Applications*, MacMillan Press Ltd., 1976

[3] A. Brandstädt and D. Kratsch, *On Domination Problems for Permutation and Other Graphs*, Theoretical Computer Science, **54** (1987), 181-198

[4] J. S. Deogun and G. Steiner, *Polynomial Algorithms for Hamiltonian Cycle in Cocomparability Graphs*, SIAM Journal on Computing, Vol. 23, 3 (June 1994), 520-552

[5] J. S. Deogun, C. Riedesel, *Traceability and Toughness in Permutation Graphs*. Accepted to appear in Journal of Combinatorial Mathematics and Combinatorial Computing.

[6] J. S. Deogun, D. Kratsch, C. Riedesel, and G. Steiner. *Graph Clustering*. In preparation.

[7] B. Duschnik and E. Miller, *Partially Ordered Sets*. Amer. J. Math., 63:600-610 (1941)

[8] S. Even, A. Pnueli, and A. Lempel, *Permutation Graphs and Transitive Graphs*, Journal of the ACM, Vol 19, 3 (1972), 400-410

[9] S. Felsner and L. Wernisch, *Maximum k-Chains in Planar Point Sets: Combinatorial Structure and Algorithms*, Proc. 25th Ann. ACM Symp. on the Theory of Computing, (1993), 146-153

[10] S. Felsner, Rudolf Müller, and L. Wernisch, *Trapezoid Graphs and Generalizations, Geometry and Algorithms*, (1995)

[11] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York (1980)

[12] M. Habib, R. H. Möring, and g. Steiner, *Computing the Bump Number is Easy*, Order **5** (1984), 107-129

[13] D. S. Johnson, *The* NP*-completeness Column: An Ongoing Guide*, Journal of Algorithms **6** (1985), 434-451

[14] J. M. Keil, *Finding Hamiltonian Circuits in Interval Graphs*, Information Processing Letters **20** (1985), 201-206

[15] O. Ore, *Theory of Graphs*, Section 10.4 Amer. Math. Soc. Colloq. Publ. 38, Providence, RI. MR27 #740 (1962)

[16] A. Pnueli, S. Even, and A. Lempel, *Transitive Orientation of Graphs and Identification of Permutation Graphs*, Canad. J. Math., **23** (1971), 160-175

[17] C. Riedesel, *Permutation Graphs: Hamiltonian Paths*, The Journal of Combinatorial Mathematics and Combinatorial Computing, **19** (1995), 55-64

[18] C. Riedesel, *Structure of Permutation Graphs*, Doctoral Dissertation, University of Nebraska - Lincoln, (1995)

[19] J. Spinrad, A. Brandstadt, and L. Stewart, *Bipartite Permutation Graphs*, Discrete Applied Mathematics **18** (1987), 279-292

[20] K. J. Supowit, *Decomposing a Set of Points into Chains with an Application to Permutation and Circle Graphs*, Information Processing Letters **21** (1985), 249-252

Appendix – GLHC Algorithm

```
(* Input:  Highest Layering of Vertices *)
(* Output:  Cycle Framework *)


(* Lowest layer is a Base layer *)
Working layer ← Lowest layer
Repeat
    Find two optimal transitions from Base (Working) layer
    Potential Base layer ← Higher transition layer
    Increment Working layer
    Repeat
        Find optimal exit from Working layer
        Case Transition vertex
            Below Potential Base layer:
                Finish transition
            At Potential Base layer:
                Finish transition
            Above Potential Base layer:
                Create new Potential Base layer
                Finish old transition
        Increment Working layer
    Until Working layer is a Base layer
Until Highest layer is Working layer or Failure
```