Dynamic Search in Graphs

F. R. K. Chung

Bell Communications Research
Morristown, New Jersey 07960

R. L. Graham

Bell Laboratories
Murray Hill, New Jersey 07974

M. E. Saks

Bell Communications Research
Morristown, New Jersey 07960

ABSTRACT

Suppose $G$ is a fixed finite connected graph and for any two vertices $x$ and $y$ in $G$, $d_G(x,y)$ denotes the distance in $G$ between $x$ and $y$, i.e., the number of edges in a shortest path connecting $x$ and $y$. Given an infinite sequence $Q = (q_1, q_2, \ldots)$ of vertices in $G$, suppose we would like to find another sequence $P = P_Q = (p_0, p_1, \ldots)$ of vertices so that the quantity

$$\nu(P) = \lim \sup_{N \to \infty} \frac{1}{N} \left( \sum_{i=1}^{N} (d(p_{i-1}, p_i) + d(p_i, q_i)) \right)$$

is as small as possible. This question represents a general formulation of a class of problems arising in self-adjusting data structures.

In this paper we will investigate this and a number of related graph searching problems, such as requiring $p_n$ to be chosen before $q_{n+i}$, $i \geq k$, is known, and show how a number of interesting structural and algorithmic concepts from graph theory come together rather naturally, e.g., isometric embedding, Steiner points in graphs, retracts, diameters and linear programming.
Dynamic Search in Graphs

F. R. K. Chung
Bell Communications Research
Morristown, New Jersey 07960

R. L. Graham
Bell Laboratories
Murray Hill, New Jersey 07974

M. E. Saks
Bell Communications Research
Morristown, New Jersey 07960

Introduction and Background

In a sequential search file, a set of records is arranged in a linear list \( L = (\ell_1, \ell_2, \ldots) \). When a record \( \rho \) is requested, the list \( L \) is searched from the first entry \( \ell_1 \) of \( L \) and consecutive entries are probed until the the requested record \( \rho \) is found. If \( \rho \) occupies the \( i \)th position in \( L \), the cost of this access will be \( i \). Such a model of sequential search has long been in use and has an extensive literature (see [31,32,42]).

It is not difficult to show that if the access frequencies for the requests are known then the best list \( L_{OPT} \), i.e., one for which the average access cost is as small as possible, is formed by arranging the records in order of decreasing frequency. However, it may happen that the access frequencies are not known a priori, and that in general the average cost per access can be decreased by rearranging the list from time to time. A number of different such self-adjusting schemes have been investigated in the literature. These include "move-to-front" [12,31,32], "transport" [42], "more-ahead-k" [9,10], "k-in-a-row" [21,29] and "k-in-a-batch" [21]. Various analyses of these and other schemes, both
mathematical and experimental, can be found in [8,9,10].

However, a fundamental question still remains unresolved, namely, what is the *optimal* self-adjusting algorithm. By this we mean an algorithm which results in the *least* cost per access for any sequence of requests. In contrast to the earlier situation of a *static* list $L_{OPT}$, we now allow *dynamically changing* lists. However to change a list $L$ to some other list $L'$ entails a *cost*. The cost measure we will use for our analysis is just the *minimum number of transpositions* needed to transform $L$ to $L'$. Thus, in our model, each probe costs 1 and each transposition costs 1. (We will discuss the possibility of different weightings at the end of the paper.) In order to fix ideas, let us examine a simple special case, namely, the case in which we have just *three* records, say, $a,b$ and $c$. We are given some arbitrary sequence $Q = \{q_1,q_2,\ldots\}$ with $q_i \in \{a,b,c\}$. We are required to produce a sequence of lists $L_0,L_1,L_2,\ldots$, each $L_i$ being some permutation of $\{a,b,c\}$, where we choose without loss of generality $L_0 = (a,b,c)$. For $i = 1,2,3,\ldots$, the cost associated with the $i$th request $q_i$ is the *sum* of the cost of transforming list $L_{i-1}$ to list $L_i$, and the cost of finding record $q_i$ in list $L_i$ (i.e., 1, 2 or 3 depending on whether $q_i$ is the first, second or third entry in $L_i$).

We can model this process in terms of moving a *pebble* $\pi$ on a graph $G$, in this case consisting of a 6-cycle $C_6$, labelled as shown in Fig. 1.
In fact, this is just the graph formed by taking the set of six permutations of 
\{a, b, c\} as its vertex set, and placing an edge between two vertices if the 
corresponding permutations differ by a single transposition.

Let \( d \) denote the usual (path-metric) distance on this graph, where if \( X, Y \subseteq V = \{a, b, c\} \) then \( d(X, Y) = \min\{d(x, y) : x \in X, y \in Y\} \). Any sequence of 
lists \( L_0, L_1, L_2, \ldots \) can be regarded as successive positions occupied by the pebble \( \pi \), 
starting from the initial vertex \( L_0 = (a, b, c) \) (which are identify with vertex \( abc \), 
etc.). Partition the vertex set \( V \) into three sets: \( V_a = \{abc, acb\}, V_b = \{bac, bcd\} \) and \( V_c = \{cab, cba\} \). Thus, for the request sequence \( Q = (q_1, q_2, \ldots) \) and the 
"pebble sequence" \( P = (L_0, L_1, L_2, \ldots) \), the cost of the \( i \)-the access is just

\[
c_i(Q, P) = d(L_{i-1}, L_i) + d(V_{q_i}, L_i) + 1,
\]

where the term +1 comes from the fact that the cost of probing the list \( L_i \) to find 
the record \( q_i \) is one more than the distance of (vertex) \( L_i \) to the corresponding set 
\( V_{q_i} \). One goal might be, given \( Q \), to determine \( P \) so that
\[ c(Q, P) = \limsup_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} c_i(Q, P) \]

is minimized. Other possible objectives will be mentioned in subsequent sections.

It was shown by Tarjan and Wei [45] that the following algorithm achieves this desired minimum value for this case. Suppose \( L_i = abc \) (without loss of generality) and \( Q_{i+1} = (q_{i+1}, q_{i+2}, q_{i+3}, \ldots) \) is the current request sequence seen after \( i \) steps. To form the list \( L_{i+1} \) move \( b \) in front of \( a \) only if two \( b \)'s occur in \( Q_{i+1} \) before one \( a \) occurs. Similarly, move \( c \) in front of \( a \) only if two \( c \)'s occur before one \( a \) occurs, and do the same for \( b \) and \( c \). Thus, the relative order of each pair in \( \{a, b, c\} \) for \( L_{i+1} \) is determined, which thereby determines \( L_{i+1} \).

This same technique gives an algorithm for generating an optimal sequence of lists in the case of two records, in which \( L_{i+1} \) can be determined by only knowing the next two symbols \( q_{i+1} \) and \( q_{i+2} \). This is in contrast to this algorithm for the case of three records which may require unbounded look-ahead. In fact, already for the case of four or more records, the corresponding questions appear to be substantially more difficult and optimal list selection algorithms are not currently known. One problem with the preceding approach for the use of four records is that the adjoining graph \( G_{24} \) of lists now has 24 vertices and has a certain amount of structure (see Fig. 2).
Our approach in this paper will be to focus on these generic questions with two changes: (1) We will consider all connected graphs $G$ rather than just those arising from permutations of an $n$-set; (2) The requests will always consist of single vertices of $G$ rather than more general subsets of vertices.

It will be seen that for this problem we can say a fair amount, although we are still far from having a complete understanding even here.

**Moving pebbles on graphs.** We now give a more precise formulation of our problem. For a given connected graph $G = (V, E)$, let $d = d_G$ denote the usual (path-metric) distance on $G$, i.e., for $x, y \in V$, $d(x, y)$ is equal to the minimum number of edges in any path between $x$ and $y$. For a request sequence $Q = (q_1, q_2, \ldots)$ and a pebbling sequence $P = (p_0, p_1, p_2, \ldots)$, with $q_i, p_j \in V$, define
\[ c_N(Q,P) = \sum_{i=1}^{N} \left( d(p_{i-1}, p_i) + d(p_i, q_i) \right), \]

\[ \bar{c}(Q, P) = \limsup_{N \to \infty} \frac{1}{N} c_N(Q, P), \]

\[ \bar{c}(Q) = \inf_{P} \bar{c}(Q, P), \]

\[ \lambda(G) = \sup_{Q} \bar{c}(Q), \]

We call \( \lambda(G) \) the \textit{search value} of \( G \). We currently know of no polynomial-time algorithm for determining \( \lambda(G) \). Note that we have normalized \( c_N \) by omitting the automatic +1 term occurring in (1).

Let us call the sequence \( P \) \textit{Q-optimal} if

\[ \sup_{N} (c_N(Q, P) - c_N(Q, \hat{P})) \]

is bounded for all pebbling sequences \( \hat{P} \). For any \( Q \), \textit{Q-optimal} sequences always exist, as the following argument shows. Let \( Q_k \) denote the finite request sequence \((q_1, q_2, \ldots, q_k)\) and suppose \( P_k = (p_{k0}, p_{k1}, \ldots, p_{kk}) \) denotes an \textit{optimal} pebbling sequence for \( Q_k \). That is, \( P_k \) minimizes

\[ c_k(Q_k, P_k) = \sum_{i=1}^{k} \left( d(p_{k,i-1}, p_{ki}) + d(p_{ki}, q_i) \right) \]

over all possible pebbling sequences of length \( k + 1 \). Define an infinite pebbling sequence \( P^* = (p_0^*, p_1^*, \ldots) \) using the König infinity lemma, so that any initial segment \( P_m^* = (p_0^*, p_1^*, \ldots, p_m^*) \) occurs as an initial segment of infinitely many of the \( P_k \). However, for any \( i \leq k \), if \( P_k(i) \) denotes the initial segment \((p_{k0}, p_{k1}, \ldots, p_{ki})\) then
\[(3) \quad c_i(Q_i, P_k(i)) - c_i(Q_i, P_i) \leq \text{diam}(G)\]

where \(\text{diam}(G)\) denotes the \textit{diameter} of \(G\). This follows from the observation that if (3) did not hold then the first \(i+1\) terms of \(P_k\) would be replaced by \(P_i\), thereby forming a pebbling sequence \(\hat{P}_k\) with

\[c_k(Q_k, \hat{P}_k) < c_k(Q_k, P_k),\]

which contradicts the definition of \(P_k\).

For a finite request sequence \(Q_k = (q_1, \ldots, q_k)\), we can characterize an optimal pebbling sequence \(P_k = (p_0, p_1, \ldots, p_k)\) in other terms as follows. Consider the tree \(S(Q, P)\) shown in Fig. 3.

Such a graph is often called a \textit{caterpillar}, with \textit{leaves} \(p_0, q_1, q_2, \ldots, q_k\) and \textit{internal vertices} \(p_1, p_2, \ldots, p_k\). Since \(P_k\) is optimal for \(Q_k\) then we must have for all \(i\),

\[\]
(4) \[ d(p_{i-1}, p_i) + d(p_i, q_i) + d(p_i, p_{i+1}) \leq d(p_{i-1}, x) + d(x, q_i) + c_k(x, p_{i+1}) \]

for all vertices \( x \in G \) (otherwise, replacing \( p_i \) by \( x \) would decrease \( c_k(Q_k, P_k) \)). Such a point \( p_i \) is called a Steiner point for the set \( \{p_{i-1}, q_i, p_{i+1}\} \). The set of all such Steiner points will be denoted by \( S(p_{i-1}, q_i, p_{i+1}) \). Thus, \( p_i \in S(p_{i-1}, q_i, p_{i+1}) \) for \( 1 \leq i < k \). We will call the corresponding caterpillar a Steiner minimal caterpillar for \( Q \).

The windex of \( G \). An algorithm \( A \) which produces a \( Q \)-optimal pebbling sequence \( A(Q) \) for each request sequence \( Q \) will be said to be an optimal algorithm for \( G \).

It can happen that an algorithm \( A \) can produce \( Q \)-optimal algorithms even though at any time only a finite portion of \( Q \) can be seen by \( A \).

Definition. A graph \( G \) is said to have windex \( k \), written \( \omega x(G) = k \), if there is an optimal algorithm \( A \) for \( G \) with the property that \( A \) always determines \( p_i \) with only knowledge of \( q_j \) for \( j < i + k \).

If there is no such \( k \) for \( G \), we write \( \omega x(G) = \infty \). The name windex, a shortened form of window index, refers to the fact that one can think of \( A \) as having a window through which exactly \( k \) future request symbols of \( Q \) can be seen.

In this section we discuss various elementary properties of the windex function. In studying graphs with windex \( k \) it is useful to consider the process as a game between two players, Red and Blue. At the \( i^{th} \) step of the game:

(a) Red selects the \((i+k)^{th}\) request vertex \( q_{i+k} \);
(b) Blue then selects the $i$th pebble vertex $p_i$ and pays Red the amount
\[ d_G(p_{i-1}, p_i) + d_G(p_i, q_i). \]
The initial choices of $p_0$ and $q_1, \ldots, q_k$ can be made arbitrarily.

Of course, the object of Blue is to minimize the amount paid to Red, whereas Red would like to maximize this amount.

**Lemma 1.** For any nontrivial graph $G$, $\omega x(G) \geq 2$.

**Proof:** Let $\{u, v\}$ be some edge of $G$ and suppose Blue has available only a window of length 1, i.e., at the $i$th step Blue can only see $q_{i+1}$. Suppose $p_i = u$ and Blue sees $q_{i+1} = v$. If Blue elects to move the pebble $\pi$ to $v$, i.e., selects $p_{i+1} = v$, then Blue pays 1 and Red can select $q_{i+2} = u$, reversing the preceding situation. Blue pays (possibly) even more if some $p_i \notin \{u, v\}$ is selected. On the other hand, if Blue choose $p_{i+1} = u$ then Blue pays 1 and Red can select $q_{i+2} = v$, duplicating the preceding situation. Thus, in any case, Red can choose the request sequence so that Blue pays at least 1 unit per request. However, for any sequence $Q = \{q_1, q_2, \ldots, q_N\}$ with all $q_i \notin \{u, v\}$, Blue never has to pay more than $\frac{1}{2}N + \text{diam} (G)$, by just going to and staying at the more frequently occurring symbol. Thus, we must have $\omega x(G) \geq 2$. □

**Lemma 2.** If $T$ is any nontrivial tree then $\omega x(T) = 2$.

**Proof:** By Lemma 1, it suffices to show $\omega x(T) \leq 2$. Suppose the pebble is at $p_i$, Blue sees $q_{i+1}$ $q_{i+2}$ and wants to determine $p_{i+1}$. We know that $p_{i+1}$ should be a Steiner point of $\{p_i, q_{i+1}, p_{i+2}\}$ (although $p_{i+2}$ is not yet determined). However,
since $T$ is a tree, $S(p_i, q_{i+1}, p_{i+2})$ will always consist of a unique vertex, which in fact, is just the same as the Steiner point $S(p_i, q_{i+1}, q_{i+2})$. Thus, Blue can construct a $Q$-optimal sequence using a window of length 2, and the proof is completed. ■

Lemma 3. For the complete graph $K_n$ on $n$ vertices, $\omega x(K_n) = n$.

Proof: We first show that $\omega x(K_n) > n - 1$. Let $V = \{v_1, \ldots, v_n\}$ and suppose (without loss of generality) $p_i = v_i$ and a length $n - 1$ window shows $q_{i+1} = v_2, q_{i+2} = v_3, \ldots, q_{i+n-1} = v_n$. If $q_{i+n} = v_1$ then $\pi$ should stay at $v_1$ (since otherwise Red is paying more than is necessary for this segment). On the other hand, if $q_{i+n} = v_2$ then $\pi$ should move to $v_2$ (otherwise Red again pays too much). Thus, $\omega x(K_n) > n - 1$.

In the other direction, it is not difficult to prove by induction that the following algorithm with a window of length $n$ is optimal: choose $p_{i+1}$ to be the first repeated vertex in the sequence $p_i, q_{i+1}, q_{i+2}, \ldots, q_{i+n}$. This shows that $\omega x(G) \leq n$. ■

For two graphs $G$ and $H$, the product of $G$ and $H$, denoted by $G \Box H$, is defined to be the graph with vertex set $\{(u, v) : v \in V(G), v \in V(H)\}$ and having as edges all pairs $\{(u, v), (u', v')\}$ where either $u = u'$ and $(v, v') \in E(H)$ or $v = v'$ and $(u, u') \in E(G)$. It is easy to check that

$$d_{G \Box H}((u, v), (u', v')) = d_G(u, u') + d_H(v, v').$$
Lemma 4.

\[ \omega_x(G \sqcup H) = \max \{ \omega_x(G), \omega_x(H) \} \]

Proof: Let \((q_1, q_1'), (q_2, q_2'), \ldots\) be a request sequence in \(G \sqcup H\). Suppose \(p_0, p_1, p_2, \ldots\) forms an optimal pebbling sequence for \(q_1, q_2, \ldots\) in \(G\), and \(p'_0, p'_1, p'_2, \ldots\) forms an optimal pebbling sequence for \(q_1', q_2', \ldots\) in \(H\). It is straightforward to check that in fact \((p_0, p'_0), (p_1, p'_1), (p_2, p'_2), \ldots\) forms an optimal pebbling sequence for \((q_1, q_1''), (q_2, q_2''), \ldots\) in \(G \sqcup H\). Thus,

\[ \omega_x(G \sqcup H) \leq \max \{ \omega_x(G), \omega_x(H) \}. \]

The reverse inequality is immediate and the lemma is proved. \(\Box\)

By the \(n\)-cube \(Q_n\) we mean the graph \(K_2 \sqcup K_2 \sqcup \cdots \sqcup K_2\) (\(n\) factors). As an immediate corollary of Lemma 4, we have

\[ \omega_x(Q_n) = 2. \]

Suppose \(G\) and \(H\) are graphs sharing exactly one common vertex \(v\). Let \(G \cup H\) denote the union of \(G\) and \(H\).

Lemma 5.

\[ \omega_x(G \cup H) = \max \{ \omega_x(G), \omega_x(H) \} \]

Proof: Let \(Q = (q_1, q_2, \ldots, v)\) be a request sequence in \(G \cup H\). We will construct an optimal pebbling sequence \(P = (p_0, p_1, p_2, \ldots)\) inductively. Let \(k\) denote the maximum of \(\omega_x(G)\) and \(\omega_x(H)\). We will define \(p_i\) using only knowledge of
\( Q_k = (q_{i+1}, \ldots, q_{i+k}) \). Without loss of generality we may assume \( p_{i-1} \in V(G) \).

Suppose either \( q_i \) or \( q_{i+1} \) is in \( V(G) \). It then easily follows that \( S(p_{i-1}, q_i, p_{i+1}) \) must be contained in \( V(G) \). Form \( Q'_k \) from \( Q_k \) by replacing each \( q_j \) in \( V(H) \) by \( v \). Now use \( Q'_k \) to determine an optimal choice (in \( G \)) for \( p_i \). On the other hand, if both \( q_i \) and \( q_{i+1} \) are in \( V(H) \) then form \( Q''_k \) from \( Q_k \) by replacing each \( q_j \) in \( V(G) \) by \( v_1 \) and use \( Q''_k \) to determine an optimal choice (in \( H \)) for \( p_i \). In either case, we have managed to determine one more internal vertex in a Steiner minimal caterpillar for \( Q \), using a window of length \( k \).

Note that Lemma 2 is just a consequence of Lemmas 5 and 3 (with \( n = 2 \)). Using the preceding results we can construct large families of graphs having windex 2. These include not only trees, \( n \)-cubes and grids but various types of recursive combinations of these (using products and unions).

However, it turns out that induced or even isometric subgraphs of windex 2 graphs may not themselves have windex 2. A simple example is the 6-cycle \( C_6 \), an isometric subgraph of \( Q_3 \), which happens to have infinite windex.

We next recall a concept from topology which will be very relevant to our study of the windex of a graph.

**Definition.** A subgraph \( H \) of \( G \) is called a retract of \( G \) if there is a mapping from \( V(G) \) to \( V(H) \) which preserves edges, i.e., which maps adjacent vertices in \( G \) to adjacent vertices in \( H \). Similarly, \( H \) is called a weak retract of \( G \) if there is a mapping from \( V(G) \) to \( V(H) \) such that adjacent vertices in \( G \) are mapped to either adjacent vertices or a single vertex in \( H \).
Lemma 6. If $H$ is a weak retract of $G$ then

$$\omega(H) \leq \omega(G).$$

Proof: For a request sequence $Q = (q_1, q_2, \ldots, q_N)$ in $H$, let a corresponding optimal pebbling sequence in $G$ be denoted by $P = (p_0, p_1, p_2, \ldots, p_N)$. Let $f: V(H) \to V(G)$ be a mapping making $H$ a weak retract of $G$, and consider the pebbling sequence $f(P) = (f(p_0), f(p_1), f(p_2), \ldots, f(p_N))$ in $H$.

Clearly,

$$c_N(Q, f(P)) \leq c_N(Q, P).$$

Thus, we may restrict our search for a $Q$-optimal pebbling sequence to the subgraph $H$. Thus, if $\omega(H) \geq k$ then $\omega(G) \geq k$ as well. This proves the lemma. □

We conclude this section with several examples of graphs having infinite windex.

Lemma 7. Let $K_{2,3}$ denote the complete bipartite graph on vertex sets of sizes two and three. Then

$$\omega(K_{2,3}) = \infty.$$ 

Proof: Let the vertex sets of $K_{2,3}$ be denoted by $\{x_1, x_2\}$ and $\{y_1, y_2, y_3\}$, where the edges of $K_{2,3}$ are exactly all the pairs $\{x_i, y_j\}$. Suppose $\omega(K_{2,3}) = k < \infty$.

Consider a request sequence formed by repeating the sequence $(y_1y_2y_3)^* z$ where $(y_1y_2y_3)^*$ means that the string $y_1y_2y_3$ is repeated so that it has length greater
than \(k + 10\), and \(z\) is either \(x_1\) or \(x_2\). When Blue sees the window with (just) an initial segment of \((y_1y_2y_3)^*\) visible, there are several options for the pebble \(\pi\). If \(\pi\) is moved to \(x_1\) then this is not optimal if \(z\) were to be \(x_2\). Similarly, if \(\pi\) is moved to \(x_2\) then this is not optimal if \(z\) were to be \(x_1\). On the other hand, to move to (or stay at) any of the \(y_i\) is also suboptimal since Blue pays at least 4 for each occurrence of the three requests \(y_1y_2y_3\) with this strategy, which costs more than moving to some \(x_i\) right away. We can repeat this process infinitely often in a request sequence \(Q\), causing Blue to pay an unbounded amount more than \(N\) for the first \(N\) requests. However, an \(Q\)-optimal pebbling sequence \(P^*\) satisfies \(c_N(Q, P^*) \leq N + 2\). Thus, \(\omega_x(K_{2,3}) > k\). Since \(k\) was arbitrary, the lemma is proved. ■

**Lemma 8.** For the 5-cycle \(C_5\), \(\omega_x(C_5) = \infty\).

**Proof:** Let \(V(C_5) = \{0, 1, 2, 3, 4\}\) and suppose \(\omega_x(C_5) = k < \infty\). Consider a request sequence \(Q\) formed by concatenating subsequences of the form \(S = 002414(24130)^*z\) where \(z\) is either 0 or 2. The block 24130 is repeated \(\omega\) times where \(5\omega > k\). Let \(p_0, p_1, p_2, \ldots\) denote the (purported) optimal pebbling sequence. We may assume \(p_0 = p_1 = 0\) without increasing the cost. If \(p_5 \neq 0\) or \(4\), \(z\) could be chosen to be 0. The total cost of accessing \(S\) is at least \(6(\omega + 1)\), one more than the optimal cost of \(6(\omega + 1) - 1\) achieved by choosing \(p_5 = 0\) or 4. On the other hand, if \(p_5 = 0\) or 4 then \(z\) could be chosen to be 2. In this case, the total cost Blue pays is at least \(6(\omega + 1) + 1\), again which is one more than the optimal cost. Thus, as in the previous lemma, \(\omega_x(C_5) > k\) for any \(k\), and the
proof is complete. ■

Graphs of windex 2. In this section we will characterize the class of graphs having windex 2. A consequence of this characterization will be a polynomial-time algorithm for deciding if \( \omega_r(G) = 2 \).

Before we state the main theorem we need a definition. A graph \( G \) is called a median graph if for any three distinct vertices \( a, b \) and \( c \) of \( G \), there is a unique vertex of \( G \) which lies simultaneously on shortest paths joining \( a \) and \( b \), \( a \) and \( c \), and \( b \) and \( c \). Median graphs arise naturally in the study of ordered sets and discrete distributive lattices, and have an extensive literature (cf. [3,4,5,6,28,33,34,35,36]). We say that \( G \) has the unique Steiner point property if \( S(a,b,c) \) contains exactly one element.

Theorem 1. For a (nontrivial) connected graph \( G \), the following four statements are equivalent:

(a) \( \omega_r(G) = 2 \);

(b) \( G \) has the unique Steiner point property;

(c) \( G \) is a median graph;

(d) \( G \) is a retract of \( Q_n \) for some \( n \).

Proof: First, observe that the implication \((d) \Rightarrow (a)\) is an immediate consequence of Lemmas 1 and 6, and equation (5). We also point out that the equivalence of \((c)\) and \((d)\) has been proved by H. J. Bandelt [7]. We will prove \((a) \Rightarrow (b) \Rightarrow (c)\).
Before beginning, we need several definitions. For vertices \( u \) and \( v \) in \( G \), a shortest path joining \( u \) and \( v \) is called a \((u,v)\)-path. Let \( SP(u,v) \) denote the union of all vertices on all \((u,v)\)-paths.

**Proof of (a) \( \Rightarrow \) (b):** Suppose \( \omega_G(G) = 2 \). Let \( a, b \) and \( c \) be three distinct vertices in \( G \) having two distinct Steiner points, say \( s \) and \( s' \). Furthermore, among all such triples, choose \( a, b \) and \( c \) so that \( t = d(a,s) + d(b,s) + d(c,s) \) is as small as possible. First, consider a request subsequence \( aabczz \) where \( z \) is either \( s \) or \( s' \). Let the corresponding pebbling sequence be denoted by \( p_0, p_1, p_2, ... \). Thus, we must have \( p_1 = a = p_2 \) and the window shows \( b, c \). Without loss of generality we can assume \( p_3 \neq s' \). Take \( z = s' \). The Steiner minimal caterpillar for the sequence \( b, c, s', s' \) with \( p_2 = a \) has length \( t \). Then \( d(p_4, s') \) must be 0, i.e., \( p_4 = s' \), if \( \omega_G(G) \) is to be 2. Since we assumed \( p_3 \neq s' \), and since it must be some Steiner point for \( \{a, b, c\} \) then we can assume without loss of generality that \( p_3 = s \). Therefore, \( d(s, s') + d(s', c) = d(s, c) \), and so, \( s' \) is on an \((s, c)\)-path. By the minimality assumption in choosing \( \{a, b, c\} \) we must have \( c = s' \), since otherwise would could have chosen the set \( \{a, b, s'\} \).

For this set, since

\[
d(a, s') + d(b, s') + d(c, s') = t = d(a, s) + d(b, s) + d(c, s)
\]

\[
= d(a, s) + d(b, s) + d(s, s') + d(s', c)
\]

then

\[
d(a, s') + d(b, s') = d(a, s) + d(b, s) + d(s', s') = t'.
\]
But $s$ is a Steiner point of $\{a,b,s\}$ which then implies that $s'$ also is a Steiner point of $\{a,b,s\}$. Thus, we can conclude that $c = s'$ must be a Steiner point of $\{a,b,c\}$. By symmetry, $a$ and $b$ must also be Steiner points of $\{a,b,c\}$.

Now, in the request subsequence $aabcezz$, suppose $z = a$. Then it follows that $p_1 = a = p_2$, and also $p_5 = p_4 = a$, and therefore $p_3 = a$. However, if $z = b$ then the same argument forces $p_5 = p_4 = p_3 = b$. Since $a \neq b$ then we have a contradiction. This shows $(a) \Rightarrow (b)$.

**Proof of $(b) \Rightarrow (c)$:** Suppose $G$ has the unique Steiner point property. We will show that for any three distinct vertices $a$, $b$ and $c$, the unique Steiner point $s$ in fact satisfies

$$s = \text{SP}(a,b) \cap \text{SP}(a,c) \cap \text{SP}(b,c)$$

By symmetry, it will be enough to show that

$$s \in \text{SP}(a,b).$$

Suppose $s \notin \text{SP}(a,b)$. Let $t \in \text{SP}(a,b)$ so that $d(s,t) = i > 0$ is as small as possible. Let $t = u_0, u_1, \ldots, u_i = s$ be a $(t,s)$-path.

**Claim 1.** $d(a,u_i) = d(a,t) + 1$.

**Proof:** Since $|d(a,u_i) - d(a,t)| \leq 1$ then $d(a,u_i)$ is either $d(a,t) - 1$, $d(a,t)$ or $d(a,t) + 1$. However, if $d(a,u_i) = d(a,t) - 1$ then $u_1 \in \text{SP}(a,b)$ which implies $d(s,\text{SP}(a,b)) < i$, a contradiction. Also, if $d(a,u) = d(a,t)$ then the Steiner minimal tree for the three vertices $a$, $t$ and $u_1$ has length $d(a,t) + 1$, and there are two possible Steiner points, $t$ and $u_1$, which achieve this minimum total
length. This is also impossible, so the claim is proved.

Claim 2. \( d(a, u_j) = d(a, t) + j \) for \( j \leq i \).

Proof: The claim holds for \( j = 0 \) and \( j = 1 \). Suppose that for some \( j \) with \( 2 \leq j \leq i \) the claim holds for all \( j' < j \). Since \( |d(a, u_j) - d(a, u_{j-1})| \leq 1 \) then \( d(a, u_j) \) is either \( d(a, u_{j-1}) - 1, d(a, u_{j-1}) \) or \( d(a, u_{j-1}) + 1 \).

Case 1. Suppose \( d(a, u_j) = d(a, u_{j-1}) - 1 \). Consider the set \( \{a, u_{j-2}, u_j\} \), and let \( \omega \) denote the length of its Steiner minimal tree. Clearly

\[
d(a, u_{j-2}) \leq \omega \leq d(a, u_{j-2}) + 2.
\]

If \( \omega = d(a, u_{j-2}) \) then \( u_j \) is on a \((a, u_{j-2})\)-path which implies \( d(a, u_j) = d(a, u_{j-1}) - 1 = d(a, u_{j-2}) \) by induction, which is impossible. On the other hand, if \( \omega = d(a, u_{j-2}) + 2 \) then there would have to be at least two Steiner points, namely \( u_{j-2} \) and \( u_j \), which is a contradiction.

Thus, we must have \( \omega = d(a, u_{j-2}) + 1 \). Let \( s' \) denote the (unique) Steiner point of the set \( \{a, u_{j-2}, u_j\} \). It is easily checked that \( d(s', u_j) = d(s', u_{j-2}) = 1 \) and \( d(a, s') = d(a, u_{j-2}) - 1 = d(a, t) + j - 3 \). This implies \( d(s', SP(a, b)) = j - 1 \) and that \( s' \) is on an \((s, t)\)-path. However, by induction we have \( d(a, s') = d(a, t) + j - 1 \), which contradicts the preceding equation. Thus, Case 1 cannot occur.

Case 2. Suppose \( d(a, u_j) = d(a, u_{j-1}) \). Therefore, the Steiner minimal tree for the set \( \{a, u_j, u_{j-1}\} \) has length \( d(a, u_j) + 1 \), and furthermore, there are two Steiner points \( u_{j-1} \) and \( u_j \), which is impossible. Thus, Case 2 cannot occur, and we are
left with one possibility, namely
\[ d(a, u_j) = d(a, u_{j-1}) + 1 = d(a, t) + j \]
which proves Claim 2.

In the same way we can prove that
\[ d(b, u_i) = d(b, t) + i \]
Therefore,
\[ d(a, s) + d(b, s) = d(a, t) + d(b, t) + 2i = d(a, t) + d(b, t) + d(t, s) + i. \]
Since \( s \) is a Steiner point for \( \{a, b, c\} \) and therefore, also for \( \{a, b, s\} \), then we have \( i = 0 \), i.e., \( s \in P(a, b) \), a contradiction. This complete the proof of the theorem. ■

We next give an alternative characterization of graphs of win dex 2. This will lead to an efficient algorithm for determining if \( \omega(G) = 2 \). First, we need a definition. For two (connected) graphs \( G \) and \( H \), we say that \( G \) can be isometrically embedded into \( H \) if there is a map \( \phi: V(G) \to V(H) \) such that for all \( u, v \in V(G) \),
\[ d_G(u, v) = d_H(\phi(u), \phi(v)). \]
Various aspects of isometric embeddings can be found in [1, 2, 14, 15, 17, 18, 19, 20]. Suppose \( G \) is isometrically embeddable into the \( n \)-cube \( Q_n \). Thus, each vertex \( v \) of \( G \) is associated with a binary \( n \)-tuple \( \phi(v) = (v_1, \ldots, v_n) \in V(Q_n) \). For three
vertices $\vec{a} = (a_1, \ldots, a_n), \vec{b} = (b_1, \ldots, b_n)$ and $\vec{c} = (c_1, \ldots, c_n)$ of $H$, define the majority vertex $M(\vec{a}, \vec{b}, \vec{c}) = (m_1, \ldots, m_n)$ of $H$ by choosing for each $i$, $m_i = z_i$ where at least two of the values $a_i, b_i, c_i$ are equal to $z_i$. Let us call a subset $X \subseteq V(H)$, majority-closed if for any $x, y, z \in X$, $M(x, y, z) \in X$.

**Theorem 2.** A (nontrivial) graph $G$ has windex $2$ if and only if $G$ can be isometrically embedded into some $Q_n$, say by the map $\phi$, and $\phi(V(G))$ is majority-closed.

**Proof:** Suppose $\phi : V(G) \to V(Q_n)$ is an isometric embedding of $G$ into $Q_n$, and $\phi(V(G))$ is majority-closed. It is easy to see that for any three distinct vertices $\vec{a}, \vec{b}, \vec{c}$ of $Q_n$, $M(\vec{a}, \vec{b}, \vec{c})$ is their unique Steiner point. Thus, since $\phi$ preserves distances then for any three distinct vertices $a, b, c$ of $G$, $\phi^{-1}(M(\phi(a), \phi(b), \phi(c)))$ is their unique Steiner point in $G$. Therefore, by Theorem 1, $\omega_x(G) = 2$.

In the other direction, suppose $\omega_x(G) = 2$. By Theorem 1, $G$ is a retract of $Q_n$ for some $r$. Thus, for vertices $u, v$ of $G$, $d_G(u, v) \leq d_{Q_n}(u, v)$. Since (by the definition of retract) $G$ is a subgraph of $Q_n$ then $d_G(u, v) \geq d_{Q_n}(u, v)$. Therefore, $d_G(u, v) = d_{Q_n}(u, v)$, i.e., $G$ is isometrically embeddable in $Q_n$. Also from Theorem 1, we know that $G$ is a median graph. For three vertices $a, b$ and $c$ of $Q_n$, the unique vertex in $SP(a, b) \cap SP(a, c) \cap SP(b, c)$ is exactly $M(a, b, c)$. This implies that $G$ is also closed under $M$. This proves Theorem 2.  ■
Theorem 3. If $G$ has $n$ vertices then \( \omega(G) = 2? \) can be tested in $O(n^4)$ steps.

Proof: By a result in [20.5], any $n$-vertex subgraph of a $Q_n$ can have at most $c n \log n$ edges (for a fixed small $c$). First, check to verify that this holds for $G$. Next, compute a list of the distances between all pairs of vertices in $G$. This can be done in $O(ne)$ steps. Then use the decomposition algorithm given in [18] to determine whether (and, if so, how) $G$ can be isometrically embedded into some $Q_n$. This requires $O(e^2)$ steps. Finally, in $O(n^4)$ steps, determine if $G$ is closed under the majority function $M$. Since $e = O(n \log n)$ then this algorithm requires at most $O(n^4)$ steps, as required.

We point out here that the following generalization for graphs of windex $k$ has been proved by F. R. K. Chung and M. E. Saks and will appear elsewhere.

Theorem A graph $G$ has $\omega(G) \leq k$ if and only if $G$ is a retract of $K_k \Box K_k \Box \cdots \Box K_k$, ($n$ factors) for some $n$.

The search value of a graph. Recall that for request sequences $Q = (q_1, q_2, \ldots)$ and pebbling sequences $P = (p_0, p_1, p_2, \ldots)$ we have defined the search value $\lambda(G)$ of $G$ to be:

$$\lambda(G) = \sup_Q \inf_P \limsup_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} (d(p_{i-1}, p_i) + d(p_i, q_i))$$

In this section we will discuss several results relating to $\lambda(G)$. As we remarked earlier, however, we do not currently know of a polynomial time algorithm for determining $\lambda(G)$. It can be shown however ([43]) that $\lambda(G)$ is always rational. It is not difficult to show that
\[ \lambda(G \cup H) = \max\{\lambda(G), \lambda(H)\}, \]
\[ \lambda(G \square H) = \lambda(G) + \lambda(H) \]

For any graph \( G \), if we choose \( Q = (u, v, u, v, u, v, \ldots) \) where \( d(u, v) = \text{diam}(G) \) then we obtain
\[ \lambda(G) \geq \frac{1}{2} \text{diam}(G) \]

On the other hand, let \( \text{rad}(G) \) denote the radius of \( G \), defined by
\[ \text{rad}(G) = \inf \sup_{u, v} d(u, v). \]

Let \( c \) denote a vertex in the center of \( G \), i.e., such that
\[ \sup_{v} d(c, v) = \text{rad}(G). \]

By choosing \( P = (c, c, c, c, \ldots) \) then we have
\[ \lambda(G) \leq \text{rad}(G) \]

This proves the following.

*Lemma 9.*
\[ \frac{1}{2} \text{diam}(G) \leq \lambda(G) \leq \text{rad}(G) \leq \text{diam}(G). \]

It turns out that for trees, the lower bound in Lemma 9 is tight.

*Lemma 10.* If \( T \) is a tree then
\[ \lambda(T) = \frac{1}{2} \text{diam}(T) \]
Proof: For any tree $T$, 

$$2 \text{rad}(T) - 1 \leq \text{diam}(T) \leq 2 \text{rad}(T).$$

If $\text{diam}(T) = 2 \text{rad}(T)$ then the desired conclusion follows by Lemma 9. So, suppose $\text{diam}(T) = 2 \text{rad}(T) - 1$. Thus, $T$ has two centers $c_1$ and $c_2$, joined by an edge. It is easy to see that by moving the pebble on the set $\{c_1, c_2\}$ appropriately (and, of course, a window of length 2 is enough here) we can bound the cost per request by 

$$\frac{1}{2} (2 \text{rad}(T) - 1) = \frac{1}{2} \text{diam}(T).$$

This proves the lemma. \f

We next consider cycles $C_n$. As is often the case in graph theory, even cycles are somewhat easier to deal with than odd cycles.

Lemma 11.

(i) $\lambda(C_{2m}) = \frac{m}{2}$;

(ii) $\lambda(C_{2m+1}) = \frac{m(m+1)}{2m+1}$.

Proof (i): By Lemma 9 we have

$$\lambda(C_{2m}) \geq \frac{1}{2} \text{diam}(C_{2m}) = \frac{m}{2}.$$ 

We will show the reverse inequality by using a pebbling strategy which does not move the pebble! For any finite request sequence $Q_N = (q_1, q_2, \ldots, q_N)$, let $Q_N(v)$ denote the number of $q_i$ equal to $v$, where $v$ is a vertex of $C_{2m}$. Thus, $\sum_{v} Q_N(v) = N$. The cost per request $\bar{c}(v)$ of staying at a vertex $v$, i.e., selecting
$P = (v, v, v, \ldots, v)$ is

$$\bar{c}(v) = \frac{1}{N} \sum_u Q_N(v) d(u, v)$$

where $u$ ranges over all vertices of $C_{2m}$. The average of $\bar{c}(v)$ over all $v$ is just

$$\frac{1}{2m} \sum_v \bar{c}(v) = \frac{1}{2mN} \sum_v \sum_u Q_N(v) d(u, v)$$

$$= \frac{1}{2mN} \sum_v Q_N(v) \sum_u d(u, v)$$

$$= \frac{1}{2m} \sum_u d(u, v) = \frac{m}{2}$$

Thus, for some vertex $v^*$, $\bar{c}(v^*) \leq \frac{m}{2}$. Therefore, $\lambda(C_{2m}) \leq \frac{m}{2}$ as required.

(ii): An averaging argument similar to that used in the proof of (i) shows that

$$\lambda(C_{2m+1}) \leq \frac{1}{2m+1} \sum_u d(u, v) = \frac{m(m+1)}{2m+1}.$$ 

To prove the reverse inequality, we will use a request sequence of the form

$$Q = (0, m+1, m+2, 2, m+3, \ldots)$$

where we take for $V(C_{2m+1})$ the set of integers modulo $(2m+1)$. Thus, consecutive requests $q_i, q_{i+1}$ in $Q$ always satisfy

$$d(q_i, q_{i+1}) = m = \text{diam}(C_{2m+1})$$

A straightforward (but slightly messy) analysis of this choice shows that the cost the pebbler must pay for any $2m+1$ requests is always at least $m(m+1)$, no matter where the pebble happens to be at the start of the request sequence. This
then shows that \( \lambda(C_{2m+1}) \geq \frac{m(m+1)}{2m+1} \) and the proof of the lemma is complete. ■

We can apply the preceding averaging argument to general graphs \( G \) and obtain a bound \( \lambda_{LP}(G) \) on \( \lambda(G) \), which we call the linear programming bound, which is usually quite good for small graphs. It is obtained as follows. As before, for a finite request sequence \( Q_N = (q_1, q_2, \ldots, q_N) \), let \( Q_N(v) \) denote the number of occurrences of \( v \) in \( Q_N \). The cost per request of staying at \( v \) is just

\[
\bar{c}(v) = \frac{1}{N} \sum_u Q_N(v)d(u,v)
\]

\[
= \sum_u \frac{1}{N} Q_N(v)d(u,v)
\]

\[
= \sum_u \alpha(v)d(u,v)
\]

where \( \alpha(v) = \frac{1}{N} Q_N(v) \). Suppose we now consider the linear program:

\[
\sum_u x(v)d(u,v) - z \geq 0, \quad v \in V(G)
\]

\[
\sum_v x(v) = 1, \quad x(v) \geq 0,
\]

maximize \( z \).

Denote the maximum value of \( z \) by \( \lambda_{LP}(G) \). Thus, if \( \lambda > \lambda_{LP}(G) \) and \( x(v) = \alpha(v) \) satisfy \( \sum_v x(v) = 1 \) then for at least one vertex \( v_0 \),
\[
\bar{c}(v_0) = \sum_u \alpha(v_0) \, d(u, v_0) \\
\quad = \sum_u x(v_0) \, d(u, v_0) < \lambda_{LP}(G)
\]
so that keeping the pebble \( \pi \) at \( v_0 \) results in a cost per request of less than \( \lambda_{LP}(G) \).

Therefore,

\[
\lambda(G) \leq \lambda_{LP}(G).
\]

Although as we have remarked earlier this bound is usually rather good for small graphs, for "most" graphs it is off by a factor of 2! This follows from the following observation of Joel Spencer. For a fixed \( \rho, 0 < \rho < 1 \), consider the random graph \( G_\rho(n) \) on \( n \) vertices formed by selecting each potential edge independently with probability \( \rho \). Thus, almost certainly, \( \text{deg}(v) = (1 + o(1))\rho n \) and \( d(u, v) \leq 2 \) for all vertices \( u, v \) of \( G_\rho(n) \). We can bound \( \lambda_{LP}(G_\rho(n)) \) by choosing all \( x(v) = \frac{1}{n} \) in the linear program, giving

\[
\lambda_{LP}(G_\rho(n)) \geq (1 + o(1))(\rho n \cdot 1 + (1 - \rho)n \cdot 2) \cdot \frac{1}{n}
\]

\[
= 2 - \rho + o(1)
\]

However, for any fixed \( k \), if \( n \) is sufficiently large then \( G_\rho(n) \) almost certainly has the property that for any \( k \) vertices \( q_1, \ldots, q_k \), there is some vertex \( p \) with \( d(p, q_i) = 1 \) for \( 1 \leq i \leq k \). Thus, the pebblers can partition the request sequence \( Q \) into consecutive blocks of length \( k \), say \( B_1, B_2, \ldots \). For each \( B_j \), the pebble \( \pi \) is moved a distance of at most 2 to a vertex adjacent to all vertices in \( B_j \), resulting in a cost per request of at most \( \frac{1}{k} (2 + k \cdot 1) = 1 + \frac{2}{k} \). Since \( k \) can be taken
arbitrarily large then \( \lambda(G_\rho(n)) = 1 \).

A specific example in which this behavior can be demonstrated can be constructed as follows. Let \( S = PG(3,F) \) denote projective 3-space over the field \( F = GF(5) \) (cf. [3], [5]). Thus, \( S \) has 156 points and 156 planes, with each plane containing 31 points and each point lying in 13 planes. To each point \( s \in S \) we can associate a plane \( s'^\perp \), consisting of all \( t \in S \) orthogonal to \( s \), i.e., with \( s \cdot t = 0 \). Our graph \( G^* \) will have \( V(G^*) = S \) and edges \( \{u,v\} \) where \( u \in v'^\perp \) (and loops \( \{u,u\} \) are deleted). Then \( G^* \) has maximum degree 13 and diameter 2. Thus,

\[
\lambda_{LP}(G^*) \geq \frac{1}{156} (31 + 124 \cdot 2) = \frac{279}{156}
\]

by choosing all \( x(v) = \frac{1}{156} \). On the other hand, since any three points lie in some plane, we can always choose a pebbling sequence \( P \) (by partitioning \( Q \) into blocks of length 3 as described earlier) which has cost per request of at most \( 5/3 \). Since \( 5/3 < 279/156 \) then

\[
\lambda(G^*) < \lambda_{LP}(G^*)
\]

It would be interesting to find small graphs for which this holds.

The linear programming bound can be strengthened by allowing \( \pi \) to have more mobility in the following way. For a fixed integer \( k \), we will partition \( Q \) into blocks of length \( k \). The pebble will only be moved at the beginning of each block, and will remain fixed for all requests from the block. The bound we get by this strategy corresponds to the solution of the following integer programming problem:
\[ \sum_{\nu} a(\nu) d(u, \nu) - z_k, \quad \nu \in V(G), \]
\[ \sum_{\nu} a(\nu) = k, \quad a(\nu) \text{-nonnegative integers}, \]
maximize \((z_k + \text{diam}(G))/k\)

The maximum value of \((z_k + \text{diam} G)/k\) is denoted by \(\lambda_k(G)\). It is clear that

\[ \lambda(G) \leq \inf_k \lambda_k(G) = \lambda_f(G) \leq \lambda_{LP}(G) \]

Observe that for the random graph \(G_{p}(n)\),

\[ \lambda_f(G_{p}(n)) \to 1 \text{ as } n \to \infty. \]

Concluding remarks. There are numerous questions concerning dynamic search on graphs which currently remain unanswered. We will close by discussing these and some related issues.

(i) Is there a polynomial-time algorithm for computing \(\lambda(G)\)? The algorithm of Saks [43] runs in time \(O(n^n)\) where \(G\) has \(n\) vertices.

(ii) We have already mentioned that it can be shown that \(\lambda(G)\) is always rational. What is

\[ q(n) = \max\{q : \lambda(G) = P/q, \; G \text{ has } n \text{ vertices}\}? \]

It seems likely that \(q(n)\) can grow exponentially with \(n\). If \(C_5^+\) denotes the graph formed by adding one chord to a 5-cycle then it is not hard to show that \(\lambda(C_5^+) = 7/6\), thus giving an example showing \(q(5) > 5\) (this can be easily generalized to show that \(q(n) > n\)).
(iii) In all of the examples we have seen thus far, request sequences $Q = (q_1, q_2, \ldots)$ which achieve $\overline{c}(Q) = \lambda(G)$ have had the property that

$$d(q_i, q_{i+1}) = \text{diam}(G),$$

i.e., consecutive requests are as far away from each other as possible. While there is a certain intuitive justification for this property, it can sometimes fail to produce the extremal $Q$, as the following example shows. Let $G_{11}$ denote the graph shown in Fig. 4.

![Graph $G_{11}$](image)

**Figure 4**

It is easy to see that if $Q$ has for all $i$

$$d(q_i, q_{i+1}) = 5 = \text{diam}(G_{11})$$

then $\overline{c}(Q) = 5/2$ (since $d(x, y) = 5 \Rightarrow \{x, y\} = \{a, b\}$). On the other hand, it can be checked that $\lambda(G_{11}) = 8/3$ and this is achieved by $Q = (a, b, c, a, b, c, \ldots)$. We remark that $G_{11}$ also occurred as a counterexample in [11].
In this connection, the following question arises. For a (connected) graph $G$, define $\Delta(G)$, the diameter graph of $G$, by defining $V(\Delta(G)) = V(G)$ and $\{v, v\}$ is an edge of $\Delta(G)$ provided $d_G(v, v') = \text{diam}(G)$. Which graphs $H$ occur as $\Delta(G)$ for some $G$? In fact, it can be shown that all graphs $H$ occur as (connected components of) diameter graphs. Typically, $\text{diam}(G)$ contains many components.

(iv) The $\lambda$-windex of $G$. We will define $\omega_\lambda(G)$, the $\lambda$-windex of $G$, in the same way that $\omega(G)$ was defined, except that only $Q$-optimal algorithms with $\bar{c}(Q) = \lambda(G)$ must be produced (using a window of length $\omega_\lambda(G)$). All of the questions for $\omega(G)$ can also be asked for $\omega_\lambda(G)$. These are not the same functions as shown, for example, by the graph $K_{2,3}$. As we have seen, $\omega(K_{2,3}) = \infty$. However, $\lambda(K_{2,3}) = 1$ and it is not difficult to show that $\omega_\lambda(K_{2,3}) = 2$.

Another such example is given by the graph $P_{2n}$, a path with $2n$ vertices. Here, $\omega(P_{2n}) = 2$ while $\omega_\lambda(P_{2n}) = 1$. Is there a structural characterization of graphs $G$ with $\omega_\lambda(G) = k$?

(v) Of course, our choice to charge the same cost for moving the pebble across one edge as for having the pebble location $p_i$ “miss” the requested vertex $q_i$ by a distance of 1, was arbitrary (it is in some sense the simplest choice).
One could more generally define for some $\alpha > 0$,

$$C_N(Q,P) = \sum_{i=1}^{N} \left( d(p_{i-1}, p_i) + \alpha d(p_i, q_i) \right).$$

*What are the analogues of the preceding results for $\alpha \neq 1$?*
REFERENCES


Comp. 8 (1979), 82-110.


