Median computation in graphs using consensus strategies

Kannan Balakrishnan
Department of Computer Applications
Cochin University of Science and Technology
Cochin-682 022, India
e-mail:bkannan@cusat.ac.in

Manoj Changat*
Department of Futures Studies, University of Kerala
Trivandrum- 695 034, India
e-mail: changat@vsnl.com

Henry Martyn Mulder
Econometrisch Instituut, Erasmus Universiteit
P.O. Box 1738, 3000 DR Rotterdam, The Netherlands
e-mail: hmmulder@few.eur.nl

ECONOMETRIC INSTITUTE REPORT EI 2007-34

This Report is a *preprint*. Its purpose is to ascertain the right of intellectual ownership by the authors. In no way should this preprint be considered a formal publication.

Abstract

Following the Majority Strategy in graphs, other consensus strategies, namely Plurality Strategy, Hill Climbing and Steepest Ascent Hill Climbing strategies on graphs are discussed as methods for the computation of median sets of profiles. A review of algorithms for median computation on median graphs is discussed and their time complexities are compared. Implementation of the consensus strategies on median computation in arbitrary graphs is discussed.

Keywords: Consensus Strategy, Majority Strategy, Plurality Strategy, Hill Climbing, Median computation

*This work was done done under the DST, Govt. of India grant M/04-1999 awarded to this author. The financial support of the DST, New Delhi is is greatfully acknowledged.

1 Introduction

The Median Problem is a typical problem in location: given a set of clients one wants to find an optimal location for a facility serving the clients. The criterion for optimality is minimizing the sum of the distances from the location of the facility to the clients. The solution of this location problem is generally known as a median. One way to model this is using a network, where clients are positioned on points and the facility has to be placed on a point as well, see for instance [22, 9]. Another approach to formulate the median problem is in terms of achieving consensus amongst the clients (profiles). This approach has been fruitful in may other applications, e.g. in social choice theory, clustering and biology, see for instance [6, 14]. From the view point of consensus the result of Goldman [8] is very interesting: to find the median in a tree just move towards the majority of the profile. In [18] this majority strategy was formulated for arbitrary graphs. The problem now is that in general this strategy does not necessarily find a median for every profiles. It was proved that majority strategy finds the set of medians for arbitrary profiles if and only if the graph is a so-called median graph. The class of median graphs allows a rich structure theory and has many and diverse applications, see e.g. [16, 17, 14, 13]. In the majority strategy we compare the two ends of an edge v and w: if we are at v and at least half of the elements of profile is strictly nearer to w than to v, then we move to w. One could relax the requirement for making a move as follows: one may move to w if there are at least as many elements of the profile is closer to w than to v. Note that in this case less than half may actually be closer to w because there are many members of the profile having equal distance to v and w. We call this strategy the Plurality Strategy. Also two other strategies, which are well-known search strategies in Artificial Intelligence, namely Hill climbing and Steepest Ascent Hill Climbing were formulated in [5]. The graph classes where theses three strategies always produce the median of arbitrary profiles were characterized in [5]. In [4] the conditions for a graph to have connected medians for arbitrary weight functions were established. The graph classes in both the papers turned out to be the same. The aim of this paper is to compare the various consensus strategies for median computation. Also we analyze the computational and implementation aspects of these strategies on various classes of graphs.

2 Consensus Strategies

All graphs in this paper are finite, connected, undirected, simple graphs without loops. Let $G = (V, E)$ be a graph with vertex set V and edge set E. Let $n = |V|$ and $m = |E|$. The distance function of G is denoted by d, where $d(u, v)$ is the length of a shortest u, v-path. The *interval function* I of G defined by

$$I(u, v) = \{z \in V | d(u, z) + d(z, v) = d(u, v)\}.$$

So the set $I(u, v)$ consists of the vertices on shortest u, v-paths, see [16], for a systematic study of the interval function.

A *profile* $\pi = (x_1, x_2, \ldots, x_k)$ in a graph is a finite sequence of vertices, and $|\pi| = k$ is the *length* of the profile. Note that the definition of a profile allows multiple occurrences of a vertex. The *distance* of a vertex v to π is defined as

$$D(v,\pi) = \sum_{i=1}^{k} d(x_i, v).$$

A vertex minimizing $D(v,\pi)$ is a *median vertex* of the profile. The set of all median vertices of the profile π is the *median set* of π and is denoted by $M(\pi)$. For an edge vw in G, we denote by π_{vw}, the subprofile of π consisting of the elements of π strictly closer to v than to w, and by $\pi_{v=w}$ the subprofile of elements having equal distance to v and w.

There are various strategies to find "optimal" vertices with respect to profile π. Loosely speaking, in each strategy we start at an arbitrary vertex and then move along edges depending on which side of the edge the elements of the profile are. We can measure this in two ways: using the number of vertices closer to one end $|\pi_{vw}|$, or using the distance sums $D(v,\pi)$. We stop if we are stuck at a vertex or if we were able to visit vertices twice and cannot get away from such vertices. The *output* is then the vertex where we get stuck or the vertices visited at least twice.

In the Majority Strategy we move "towards majority". In the Condorcet Strategy we move "away from minority". In the Plurality Strategy we move "towards more". Finally, in the last two strategies we move away from vertices with $D(v,\pi)$ "too small".

Majority Strategy
Input: A connected graph G, a profile π on G, and an *initial vertex* in V.
Output: The unique vertex where we get stuck or the set of vertices visited at least twice.

1. Start at the initial vertex.

2. [**MoveMS**] If we are in v and w is a neighbor of v with $|\pi_{wv}| \geq \frac{1}{2}|\pi|$, then we *move* to w.

3. We move only to a vertex already visited if there is no alternative.

4. We stop when
(i) we are stuck at a vertex v *or*
(ii) [**TwiceMS**] we have visited vertices at least twice, and, for each vertex v visited at least twice and each neighbor w of v, either $|\pi_{wv}| < \frac{1}{2}|\pi|$ or w is also visited at least twice.

In the following strategies we only list the steps in which they differ from the Majority Strategy.

Condorcet Strategy

2. [**MoveCS**] If we are in v and w is a neighbor of v with $|\pi_{vw}| \leq \frac{1}{2}|\pi|$, then we *move* to w.

4. (ii) [**TwiceCS**] we have visited vertices at least twice, and, for each vertex v visited at least twice and each neighbor w of v, either $|\pi_{vw}| > \frac{1}{2}|\pi|$ or w is also visited at least twice.

Plurality Strategy

2. [**MovePS**] If we are in v and w is a neighbor of v with $|\pi_{wv}| \geq |\pi_{vw}|$, then we *move* to w.

4. (ii) [**TwicePS**] we have visited vertices at least twice, and, for each vertex v visited at least twice and each neighbor w of v, either $|\pi_{wv}| < |\pi_{vw}|$ or w is also visited at least twice.

The next two strategies were introduced to find a (local) minimum based on a heuristic function in a search graph. So the versions as in [?] make a move only to previously unexplored vertices. Because our purpose in this paper is to find all medians (i.e. the *median set*) of a profile, we have adapted the strategies such that we are able to visit vertices more than once (as in the above description of the Majority Strategy).

Hill Climbing

2. [**MoveHC**] If we are in v and w is a neighbor of v with $D(w,\pi) \leq D(v,\pi)$, then we *move* to w.

4. (ii) [**TwiceHC**] we have visited vertices at least twice, and, for each vertex v visited at least twice and each neighbor w of v, either $D(w,\pi) > D(v,\pi)$ or w is also visited at least twice.

Steepest Ascent Hill Climbing

2. [**MoveSA**] If we are in v and w is a neighbor of v with $D(w,\pi) \leq D(v,\pi)$ and $D(w,\pi)$ is minimum among all neighbors of v, then we move to w.

4. (ii) [**TwiceSA**] = [**TwiceHC**].

Although the strategies are different in their computations, they may have the same outcome depending on the graph on which they are applied. Plurality and Hill Climbing have the same outcome anyway, see the next Lemma.

Lemma 1 *Let G be a connected graph and π a profile on G. Plurality Strategy makes a move from vertex v to vertex w if and only if $D(w,\pi) \leq D(v,\pi)$.*

Proof. The assertion follows immediately from the following computation:

$$D(v,\pi) - D(w,\pi) = \sum_{x\in\pi_{vw}} d(v,x) + \sum_{x\in\pi_{wv}} d(v,x) - \sum_{x\in\pi_{vw}} d(w,x) - \sum_{x\in\pi_{wv}} d(w,x) =$$

$$= \sum_{x\in\pi_{vw}} d(v,x) + \sum_{x\in\pi_{wv}} d(v,x) - \sum_{x\in\pi_{vw}} (d(v,x)+1) - \sum_{x\in\pi_{wv}} (d(v,x)-1) = |\pi_{wv}| - |\pi_{vw}|.$$

□

For bipartite graphs, $\pi_{v=w}$ is empty for any profile π, so the following Lemma is obvious.

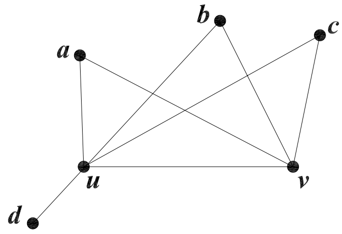

Figure 1: Illustration of consensus strategies differing on a graph

Lemma 2 *Majority, Condorcet, and Plurality Strategy coincide on connected bipartite graphs.*

On non-bipartite graphs Plurality Strategy, Condorcet Strategy and Majority Strategy are pairwise different as the following example show.

Example 1 *Consider the profile* $\pi = (a, b, c, d)$ *on the graph of Fig. 1.1. Then* $|\pi_{vu}| = 0$, $|\pi_{uv}| = 1$, $|\pi_{ua}| = |\pi_{ub}| = |\pi_{uc}| = |\pi_{ud}| = 3$, $|\pi_{au}| = |\pi_{bu}| = |\pi_{cu}| = |\pi_{du}| = 1$, $|\pi_{va}| = |\pi_{vb}| = |\pi_{vc}| = 2$, $|\pi_{av}| = |\pi_{bv}| = |\pi_{cv}| = 1$.

We apply all the strategies starting from v. For the Plurality Strategy, we move from v to u and get stuck there. So the outcome of Plurality Strategy starting from v is $\{u\}$, which is the median. Majority Strategy will not make a move from v, so we get $\{v\}$ as the outcome. Condorcet Strategy moves from v to u and from u to v, from v also to a, b, c and back. It cannot go from u to d. So the outcome of the Condorcet Strategy is $\{a, b, c, u, v\}$. Hill Climbing and Steepest Ascent Hill Climbing also produce $\{u\}$ starting from v.

Next we present an example that shows that Steepest Ascent Hill Climbing is essentially different from the other strategies. Note that the other strategies might make a move from v as soon as they find a neighbor w of v that satisfies the condition for a move, while Steepest Ascent has to check *all* neighbors of v before it can make a move. Consider the graph $K_{2,3}$ with vertices a, b and $1, 2, 3$, where two vertices are adjacent if and only if one is a 'letter'and the other a 'numeral'. Now take the profile $\pi = (b, 1, 1, 1, 2, 2, 2, 3, 3, 3)$. Then we have $D(a, \pi) = 11$, $D(b, \pi) = 9$, and $D(i, \pi) = 13$, for $i = 1, 2, 3$. Take 1 as initial vertex and assume that we check its neighbors in alphabetical order. Then Majority, Condorcet, Plurality and Hill Climbing move to a and get stuck there, whereas Steepest Ascent moves to b and thus finds the median vertex of π.

This example also shows that the first four strategies might not even find the median vertex at all, even if the graph is bipartite. The special thing about this example is that the profile $\rho(1, 2, 3)$ has median set $\{a, b\}$, which is *not* connected. In [5] it is shown that Plurality Strategy, and both Hill Climbing and Steepest Ascent, find the median set from any initial vertex, for all profiles if and only if the median sets of all profiles induce connected subgraphs.

3 Median computation in median graphs

The Majority Strategy was first formulated for trees by Goldman [8] to find the median set of a profile on a tree. In [18], it was shown that the median graphs are precisely the graphs on which this strategy always finds the median set of any profile. We quote the full theorem here, because it was the starting point for this paper. We postpone the definition of median graph until after the theorem.

Theorem 1 (Mulder [18]) *The following are equivalent for a connected graph G:*

1. *G is a median graph.*
2. *The Majority Strategy produces the median set $M(\pi)$ from any initial vertex, for each profile π on G.*
3. *The Majority Strategy produces the same set from any initial position, for each profile on G.*

It is easy to see that median graphs are bipartite. Hence, in view of Lemma 2, a similar theorem holds for Plurality Strategy, Condorcet Strategy and Hill Climbing provided we presuppose the graphs to be bipartite as well. Note that the outcome of the strategies will not always be the median set of the profile on arbitrary bipartite graphs, as our example with $K_{2,3}$ above shows.

The above strategies applied on median graphs give rise to algorithms for the computation of median sets of profiles π. In addition we present below some other algorithms for median computation on median graphs, which we develop based on the structural characterizations of median graphs from the literature.

Median graphs were introduced, independently by Avann [1], Nebeský [20] and Mulder & Schrijver [19]. For the first systematic study of median graphs, see [16], for recent surveys on median graphs and their applications, see e.g. [13, 14]. By now applications are available in such diverse areas as biology, theory of social choice, dissimilarity measures, voting theory, dynamic search, other mathematical disciplines. An important application of median graphs is within location theory. The location problem is "where to locate a service facility to minimize the costs of serving a set of given clients placed on vertices of a graph". Then any median vertex is a solution of this location problem.

A connected graph G is a *median graph* if, for every triple u, v, w of vertices, there exists a unique vertex x, called the *median* of u, v, w, such that x lies simultaneously on shortest joining u and v, v and w, and u and w. Trees are the simplest examples of median graphs. Another prime example is the n-cube Q_n. From the viewpoint of location theory the importance of median graphs is given by the following characterization: *A graph G is a median graph if and only if each profile of length 3 has a unique median*, see [7].

For our algorithms below we need some basic (but non-trivial) facts on median graphs. A *proper cover* of a graph G consists of two convex subgraphs G_1 and G_2 of G such that the union of G_1 and G_2 is G and the intersection of G_1 and G_2 is nonempty. Note that this intersection must be a convex subgraph as well (being the intersection of two convex subgraphs). Let $G' = (V', E')$ be properly covered by the subgraphs $G'_1 = (V'_1, E'_1)$ and $G'_2 = (V'_2, E'_2)$ and set $G'_0 = G'_1 \cap G'_2$. Let G_1 and G_2 be isomorphic copies of G'_1 and G'_2

respectively, and let λ_i be an isomorphism from G'_i to G_i, for $i = 1, 2$. Set $G_{0i} = \lambda_i[G'_0]$ and $\lambda_i(u') = u_i$, for u' in G'_0 and $i = 1, 2$. The *expansion* of G' with respect to the proper cover G'_1, G'_2 is the graph obtained from the disjoint union of G_1 and G_2 by inserting an edge between u_1 in G_{01} and u_2 in G_{02} for each u' in G'_0. We call the mappings λ_i the *lift maps* of the expansion. One of the main theorems in median graph theory (which basically started this area in graph theory) is the so-called *Expansion Theorem*: *A graph is a median graph if and only if it can be obtained from the one-vertex graph K_1 by successive expansions*, see [15, 16].

The "converse" of expansion is *contraction*. For an arbitrary edge ab of a connected graph G, we write

$$W_{ab} = \{u \in V \mid d(a, u) < d(b, u)\}.$$

The subgraph induced by W_{ab} is denoted by G_{ab}. Note that, if G is bipartite, then W_{ab} and W_{ba} partition V. In that case we call G_{ab}, G_{ba} a *split* of G. An essential step in the proof of the Expansion Theorem is to establish the following property of a median graph G: if G_{ab}, G_{ba} is a split of G, then any edge uv between G_{ab}, G_{ba}, with say u in G_{ab}, defines the same split, that is, $G_{ab} = G_{uv}$, whence $G_{ba} = G_{vu}$. So we may write a split as G_1, G_2. Let F_{12} be the set of edges between the two parts of the split, so that any edge u_1u_2 in F_{12} with u_i in G_i defines the same split. Let G_{0i} be the subgraph of G_i induced by the ends u_i of the edges in F_{12} lying in G_i, for $i = 1, 2$. Then it can be proved that the subgraphs G_{0i} are convex and that F_{12} induces an isomorphism "along its edges" between G_{01} and G_{02}, where u_1 is mapped onto u_2. Thus we can contract the edges of F_{12} obtaining the *contraction* G' of G with respect to the split G_1, G_2. This is the reverse of the above expansion. The *contraction map* κ of G onto G' is defined by $\kappa(u_i) = u$, for the edges u_1u_2 in F_{12} and $i = 1, 2$, and $\kappa(v) = v$ for all other vertices. It turns out that $\kappa|_{G_i} = \lambda_i^{-1}$, where λ_i are the lift maps of the expansion of G' with respect to the proper cover $\kappa(G_1), \kappa(G_2)$ of G'. In the case of trees, the expansion works as follows: we cover a tree with two subtrees sharing a vertex. In the expansion we obtain a tree with one vertex and one edge more. In the case of hypercubes the expansion works as follows: we cover an k-dimensional cube with two subgraphs both consisting of the whole n-cube. In the expansion we obtain an $(k + 1)$-dimensional cube. Another more intuitive way to describe median graphs is that a median graph, loosely speaking, is a tree-like structure consisting of hypercubes glued together along subcubes, just as a tree consists of edges (1-dimensional cubes) glued together along vertices (0-dimensional subcubes).

Let G be a median graph and G_1, G_2 a split. If π is a profile on G ,then let π_i denote the subprofile of π consisting of all elements of π lying in G_i, for $i = 1, 2$. If H is any subgraph of G let $\pi(H)$ be the subprofile of π comprising of vertices in H. We now have $\pi'_i = \kappa(\pi_i)$ and $\pi_i = \lambda_i(\pi'_i)$, where κ and λ are applied component-wise.

The algorithms below are based on the following two theorems.

Theorem 2 (McMorris, Mulder and Roberts [14]) *Let π be a profile in a median graph G with split G_1, G_2, where $G_i = (V_i, E_i)$, $i = 1, 2$. Let G' be the contraction of G based on the split and let G'_1, G'_2 be the convex cover of G' which when expanded results in G. Let $\pi_1 = \pi \cap V_1$ and $\pi_2 = \pi \cap V_2$. If $|\pi_1| \geq |\pi_2|$, then $M(\pi)\lambda_1(M(\pi'))$, where π' is obtained by contracting π to G'. If $|\pi_1| = |\pi_2|$, then $M(\pi)\lambda_1(M(\pi') \cup \lambda_2(M(\pi'))$, where λ_i, $i = 1, 2$ is an isomorphism of G'_i onto G_i.*

The next theorem is an easy consequence of the previous one.

Theorem 3 (Bandelt[2]) *Let G =(V,E) be a median graph, and let π be a profile on G then $M(\pi) = \cap\{G_1 | G_1, G_2$ is a split with $|\pi_1| \geq |\pi_2|\}$.*

For the complexities of the algorithms we need the following facts: the distance matrix of a median graph can be computed in $O(n^2)$ time [10]. Moreover, a median graph has $O(nlogn)$ edges [11].

First we give an algorithm to compute $|\pi_{uv}|$ for two adjacent vertices u and v in a bipartite graph. This algorithm uses a bi-directional *BFS*. Bi-directional search is a method used in artificial intelligence for example see [**?**, 23]. In classical bi-directional *BFS*, we start at the goal vertex and the initial vertex and expand from both directions. In our approach we start at the two adjacent vertices u and v and expand in different directions each time finding vertices closer to u and v, respectively. We assume that we have stored the frequency of occurrence of each vertex in a profile in an array called FREQ. This can be done in $O(k)$ time where k is the length of the profile.

Algorithm 1
Input: a bipartite graph G, a profile π, and two adjacent vertices u, v.
Output: the values of $|\pi_{uv}|$ and $|\pi_{vu}|$.

1. Set W_{uv} to $\{u\}$, W_{vu} to $\{v\}$.

2. Perform a Restricted BFS *to the next level from u through the vertices of W_{uv}. Add all unclassified vertices to W_{uv}, add their profile values to $|\pi_{uv}|$. If no vertices were added in this step go to step 5.*

3. Perform a Restricted BFS *to the next level from u through the vertices of W_{vu}. Add all unclassified vertices to W_{vu} and add their profile values to $|\pi_{vu}|$. If no vertices were added in this step go to step 7.*

4. If all the vertices in the profile have been classified go to step 7.

5. Add the remaining unclassified vertices to W_{vu} and set $|\pi_{vu}|$ to $|\pi| - |\pi_{uv}|$. Go to step 7.

6. Add the remaining unclassified vertices to W_{uv} and set $|\pi_{uv}|$ to $|\pi| - |\pi_{vu}|$.

7. Return the values $|\pi_{uv}|$ and $|\pi_{vu}|$.

Note that for Majority Strategy we need not have the exact values of $|\pi_{uv}|$ or $|\pi_{vu}|$. We only need to know whether $|\pi_{vu}| \geq \frac{1}{2}|\pi|$ or not. So the algorithm may be modified accordingly.

Theorem 4 *Given two adjacent vertices u and v in a bipartite graph G, Algorithm 1 correctly computes $|\pi_{uv}|$ and $|\pi_{vu}|$.*

Proof. First note that since G is bipartite, no vertex can be at equal distance from u and v. Also a vertex at level l in the *BFS* from u is closer to u than v (i.e. belongs to W_{uv}) if and only if it is at level $l+1$ in the *BFS* from v. So the algorithm will find it in the l^{th} iteration from u. Similarly a vertex closer to v at a distance l will be discovered

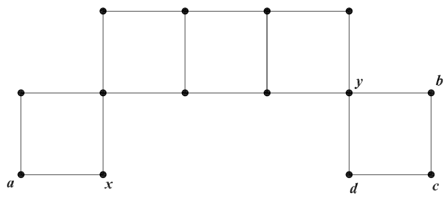

Figure 2: Majority strategy by bidirectional *BFS*

in the l^{th} iteration from v. Also note that when no vertex is closer to u than to v at level l, this means that every vertex at a distance l from u is at a distance $l-1$ from v, so every vertex at a distance $l+1$ from u will be at a distance l from v and so on. So all the remaining vertices will be closer to v. Hence the algorithm correctly computes $|\pi_{uv}|$ and $|\pi_{vu}|$.

Time complexity of Algorithm 1
Since we are visiting each vertex in a *BFS* manner from u and v, the algorithm has the same time complexity as *BFS*, that is $O(m+n)$ in a general bipartite graph, which is $O(nlogn)$ in a median graph.

Example 2 *Consider the median graph in the Figure 2. Let the profile be a, b, c, d. If we start the Majority from the vertex a, Majority Strategy makes a move from a to x after just two levels of* BFS. *Also the movement from y to b is achieved after a single level* BFS.

Time Complexity of Majority Strategy
Time for computing $|\pi_{vw}|$ for an adjacent pair of vertices is $O(n \log n)$, by Algorithm 1. Since there are $O(nlogn)$ edges the total time complexity is $O((n \log n)^2)$ and this can be stored in $O(n^2)$ space. Also an array of length n can be used to store the number each time a vertex is visited. When a vertex is visited twice the adjacent vertices can be checked to see if they satisfy the stopping condition. This can be done in $O(n)$ time. So the total time complexity is $O((nlogn)^2)$. Plurality strategy has the same complexity.

Time complexity of Steepest Ascent Hill Climbing
Evaluation of $D(x, \pi)$ takes $O(m+n)$ time on a general graph which is $O(nlogn)$ on a median graph.

BFS. Also the maximum number of computations of $D(x, \pi)$ is n.This is because we can store a computed values and lookup them . Also there is no move from a local median to a non local median median node,and no two time visits possible to a non local median graph. So when we start visiting a node,second time,we can include it in the median and after that only m visits will be taking place.

So the time complexity of the algorithm is $O(n^2logn + m)$ which is $O(n^2logn)$.

Comparison of various consensus strategies
Even though Majority Strategy and Steepest Ascent Hill Climbing both produce the median set $M(\pi)$, the paths taken by both may differ considerably. For example in the

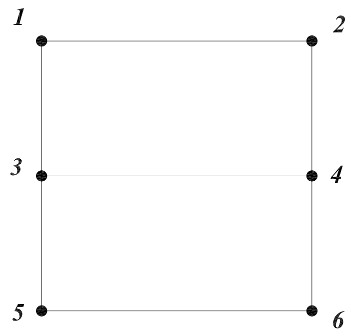

Figure 3: Comparison of Majority and Steepest Ascent Hill Climbing strategies in a median graph

median graph shown in Figure 3. consider the profile $(5, 6)$. If we start from vertex 1, the Majority Strategy may follow the path $1, 2, 4, 3, 5, 6, 5, 6$ and recognize $\{5, 6\}$ as the median of the profile. But Steepest Ascent Hill Climbing moves along the path $1, 3, 5, 6, 5, 6$ and recognizes $\{5, 6\}$ as the median. So Steepest Ascent Hill Climbing converges to a solution faster than the Majority Strategy. Also if one is interested in computing a single median vertex, but not the whole median $M(\pi)$, then Majority Strategy might use a longer route than Steepest Ascent Hill Climbing. But Steepest Ascent Hill Climbing has to perform a lot more comparisons before making a move.

Now we develop algorithms based on the structural properties of median graphs. In [11], Hagauer, Imrich and Klavžar have given an algorithm which recognizes median graphs by finding a split. The algorithm assumes that the graph has been preprocessed and a full *BFS* conducted. This requires $O(n \log n)$ time for finding a split, see [12]. Here we present an algorithm which finds a split in a median graph in $O(n \log n)$ without necessarily processing the entire graph. This algorithm is also based on bi-directional search.

Algorithm 2
Input: a median graph G, and two adjacent vertices u, v.
Output: a split G_1, G_2 of G.

1. Set W_{uv} to $\{u\}$, W_{vu} to $\{v\}$.

2. Perform a Restricted BFS *to the next level from u through the vertices of W_{uv}. Add all unclassified vertices discovered to W_{uv}. If no vertices were added in this step go to step 5.*

3. Perform a Restricted BFS *to the next level from u through the vertices of W_{vu}. Add all unclassified vertices discovered to W_{vu}. If no vertices were added in this step go to step 6.*

4. If all the vertices in the graph have been classified go to step 7.

5. Add the remaining unclassified vertices to W_{vu}. Go to step 7.

6. Add the remaining unclassified vertices to W_{uv}.

7. Return W_{uv} and W_{vu}.

Theorem 5 *Given a median graph G and two adjacent vertices u and v, Algorithm 2 correctly computes the split G_{uv}, G_{vu} of G.*

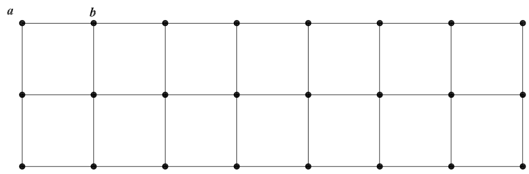

Figure 4: Using bidirectional *BFS* search to find a split in a median graph

Proof. Follows by an argument similar to that in the Proof of Theorem 4.

Algorithm 2 has also time complexity $O(n \log n)$, since it is a *BFS* conducted from two vertices. Note that the algorithm need not always process the entire graph before finding a split.

In the graph shown in Figure 4, for finding the split corresponding to the edge ab, the algorithm searches only three levels in *BFS* and does not search the entire graph. The time complexity of finding a split is $O(n \log n)$, since it is essentially a *BFS* and there are $O(n \log n)$ edges in a median graph. Also we note that in many cases it is faster because it will not have to process the entire graph.

Algorithm 3
Input: A median graph G and a profile π.
Output: The median $M(\pi)$.

1. *Find a split* G_1, G_2 *of* G.
2. *Find* π' *and* $M(\pi')$.
3. *If* $|\pi_1| \geq \frac{1}{2}|\pi|$, *go to step 4, else if* $|\pi_1| \leq \frac{1}{2}|\pi|$, *go to step5, else go to step 6.*
4. $M(\pi) = \lambda_1(M(\pi'))$, *go to step 7.*
5. $M(\pi) = \lambda_2(M(\pi'))$, *go to step 7.*
6. $M(\pi) = \lambda_1(M(\pi')) \cup \lambda_2(M(\pi'))$.
7. *Return* $M(\pi)$.

Time complexity analysis of Algorithm 3
Finding a split requires $O(n \log n)$ time by *BFS*. Finding π' can be done in $O(kn \log n)$ time. λ_1 can be computed in $O(k)$ time. Also at each step the sub graphs G_i together has at least one vertex less than their convex expansion. So if $f(n)$ is the time complexity of the algorithm, it is given by the recurrence formula $f(n) = O(kn \log n) + f(n-1)$. So we have $f(n) = O(k\{n \log n + \ldots + n \log 1\}) = O(kn^2 \log n)$.

Algorithm 4
Input: A median graph G and a profile π.
Output: The median M(π).

1. $M = V$.
2. *For each edge* e *in* E *do steps 3 to 6.*
3. *Find the split corresponding to* $e = G_1$, G_2 *with vertex sets* V_1 *and* V_2.

4. Let π_1 and π_2 be sub profiles of π in G_1 and G_2 respectively.

5. If $|\pi_1| \geq \frac{1}{2}|\pi|$, then $M = M \cap V_1$.

6. If $|\pi_2| \geq \frac{1}{2}|\pi|$, then $M = M \cap V_2$.

7. Return M.

Time complexity analysis of Algorithm 4

Finding a split requires $O(n \log n)$ time and evaluating the size of the sub profiles does not take any additional time. Also since we need only consider $O(n \log n)$ edges, the complexity is $O((n \log n)^2)$.

We have the following observations. The direct method is having the minimum time complexity $O(n^2)$, but in order to find the median it has to process the entire graph. Majority strategy can be advantageous in case the profile has very small diameter when compared to the graph. In this case we can stop by performing Algorithm 1 and finding which vertex has the majority of members near to it. So if we start with a vertex near to the profile we will not have to process the entire graph. Majority Strategy can converge very fast if we choose edges such that the number of vertices reduced in each iteration is also maximum. Also as we have seen that finding a split may be faster by Algorithm 2 if one vertex has a minority of vertices near to it. Algorithm 4 requires finding of all splits and hence it will take more time than other algorithms. Steepest Ascent Hill Climbing also need not process the entire graph and may converge faster in the case where the profile has less diameter compared to the graph.

4 Computer implementation of consensus strategies

In this section, we examine the computational results obtained by implementing the consensus strategies. We implemented the Majority, Plurality, and Condorcet Strategy, and Steepest Ascent Hill Climbing algorithms and tested them on randomly generated connected graphs and median graphs. We generated median graphs by implementing the convex expansion procedure described in [16, 17].

We tested the performance of the Majority Strategy and Steepest Ascent Hill Climbing on median graphs of sizes between 10 and 100. The relative number of moves for reaching the median set starting from a random vertex for 100 randomly chosen profiles was assessed.

We tested only these two algorithms, because all other strategies are the same as Majority Strategy on a median graph. We also compared the performance of the Majority, Plurality, Condorcet and Steepest Ascent Hill Climbing on the class of randomly generated graphs. We computed the number of graphs for which all the strategies computed the median set starting from an arbitrary vertex for 100 randomly generated profiles. Also, the same was done for the graphs, for which at least one median vertex was reached, for which a superset of the median was generated, and for which the strategies got stuck at a non-median vertex. We have not tested the statistical significance of the results, because it needs to take a sample of varying data sets of profiles, vertices and graphs. This is a problem which is to be pursued.

Weakly median graphs are a nice non-bipartite generalization of median graphs. An important property of these graphs is that median sets of profiles are always connected. In [5] it was shown that Plurality Strategy always produces the median set of a profile if and only if the graph is connected and has connected median sets for all profiles. In particular, Plurality Strategy always finds the median set on weakly median graphs.

On comparison of the strategies on 3000 median graphs, the average ratio of (*total moves Majority*)/(*total moves in Hill Climbing*) is 1.027266. This indicates that on median graphs the Steepest Ascent Hill Climbing and Majority Strategy are almost equivalent.

On comparison of strategies on 965 weakly median graphs it was found that steepest Ascent Hill Climbing and plurality have almost the same number of moves, the average ratio of (*total moves plurality*)/(*total moves in hill climbing)* is 0.984466. This indicates that in weakly median graphs Steepest Ascent Hill Climbing and Plurality Strategy are almost equivalent.

Effects of the strategies with 5000 arbitrary graphs of size between 10 and 20 with 100 profiles starting from the same random vertex

STRATEGY USED	Majority	Plurality	Condorcet	Steepest Ascent Hill climbing
Median Correctly computed	247	2872	153	3386
Computed set is not Median but contains a median vertex	0	0	2373	0
Computed Set Is a superset of $M(\pi)$	0	0	2360	0
Get stuck at a non median vertex.	53	0	0	0

Comparison of moves of Majority Strategy and Steepest Ascent Hill Climbing on arbitrary graphs

From the results, it can be seen that for arbitrary graphs, Majority Strategy computes median correctly in 5%, Plurality Strategy in 57%, Condorcet Strategy in 3% and Steepest Ascent Hill Climbing in 68% cases. This shows that the performance of Steepest Ascent Hill Climbing is slightly better than the Plurality Strategy. Also it is surprising that,only Condorcet Strategy produces at least one median vertex of all the 100 profiles(48%). Again same 48% of the cases are supersets of median for the Condorcet Strategy.

Effects of the strategies with 5000 arbitrary graphs of size between 10 and 20 with 100 profiles starting from the same random vertex

STRATEGY USED	Plurality	Steepest Ascent Hill Climbing
Median Correctly computed	2872	3386
Computed set is not Median but contains a median vertex	0	0

Effect of strategies on 3000 non-weakly median graphs of size between 10 and 20 with 100 profiles starting from the same random vertex

STRATEGY USED	Plurality	Steepest Ascent Hill Climbing
Median Correctly computed	129	173
Computed set is not Median but contains a median vertex	0	0

Comparison of moves between Steepest Ascent Hill Climbing and plurality strategies

From the results, it can be seen that for arbitrary graphs, Plurality Strategy computes median correctly in 57% and Steepest Ascent Hill Climbing in 68% cases which shows that the performance of Steepest Ascent Hill Climbing is slightly better than the Plurality Strategy. For non-weakly median graphs, Plurality Strategy computes median correctly in 4% and Steepest Ascent in 6% cases.

References

[1] *S.P. Avann*: Ternary distributive semi-lattices, Abstract 86, Bull. Amer. Math. Soc. 54 (1948) 79.

[2] *H.J. Bandelt*: Networks with Condorcet solutions, Europ. J. Oper. Research. 20 (1985) 314 – 326.

[3] *H.J. Bandelt and J.P. Barthelemy*: Medians in median graphs, Discr. Appl. Math. 8 (1984) 131 – 142.

[4] *H.J. Bandelt and V. Chepoi*: Graphs with connected medians, SIAM J. Discr. Math. 15 (2002) 268 – 282.

[5] *Kannan Balakrishnan, M. Changat and H.M. Mulder*: The Plurality Strategy on Graphs Econometric Institute Report EI 2006-??, pp. 14.

[6] *J.P. Barthelemy and B. Monjardet*: The median procedure in cluster analysis and social choice theory, Math. Social Sci. 1 (1981) 235 – 268.

[7] *F.R.K. Chung, R.L. Graham and M.E. Saks*: Dynamic search in graphs, in: " Discrete algorithms and complexity " H. Wilf et. al., Academic Press, New York, (1987) 351-387.

[8] *A.J. Goldman*: Optimal center location in simple networks, Transportation Science, (1971) 212 – 221.

[9] *A.J. Goldman and C.J. Witzgall*: A localization theorem for optimal facility location, Transportation Science, 4.(1970) 406 – 409.

[10] *J. Hagauer*: Skeletons, Recognition Algorithms and Distance Matrix of Quasi-Median Graphs, Int. J. Computer Math. Vol.55. (1995) 155 – 171.

[11] *J. Hagauer, W. Imrich and S. Klavzar*: Recognizing median graphs in subquadratic time, Theor. Comput. Sci. 215(1999) 123 – 136.

[12] *P.K. Jha and G. Slutzky*: Convex-expansion algorithms for recognizing and isometric embedding of median graphs, Ars Combin. 34 (1992) 75 – 92.

[13] *S. Klavžar and H.M. Mulder*: Median graphs: characterizations, location theory and related structures, J. Combin. Math. Combin. Comput 30(1999) pp 103 – 127

[14] *F.R. McMorris, H.M. Mulder and F.S. Roberts*: The Median Procedure on Median Graphs, Discr. Appl. Math. 84(1998) 165 – 181.

[15] *H.M. Mulder*: The structure of median graphs, Discr. Math. 24(1978) 197 – 204.

[16] *H.M. Mulder*: The Interval Function of a Graph, Mathematical Centre Tracts 132, Mathematisch Centrum, Amsterdam, 1980.

[17] *H.M. Mulder*: The expansion procedure for graphs, in: R. Bodendiek ed., Contemporary Methods in Graph Theory, B.I.- Wissenschaftsverlag , Mannheim/Wien/Zurich, 1990, 459 – 477.

[18] *H.M. Mulder*: The Majority Strategy on Graphs, Discr. Appl. Math. 80(1997) 97 – 105.

[19] *H.M. Mulder and A. Schrijver*: Median graphs and Helly hypergraphs, Discr. Math. 25 (1979) 41 – 50.

[20] *L. Nebeský*: Median graphs, Comment. Math. Univ. Carolinae 12 (1971) 317 – 325.

[21] *Rich Eline and Kevin Knight*: Artificial Intelligence, Tata McGraw-Hill 91.

[22] *R.C. Tansel, R.I. Francis and T.J. Lowe*: Location on Networks a survey I, II, Management Sci. 29 (1983) 482 – 511.

[23] *P.H. Winston*: Artificial Intelligence, 3rd edition, Addison-Wesley Publ. Co., (1992).