A New Model for Scheduling Packet Radio Networks

Arunabha Sen*and Mark L. Huson

Department of Computer Science and Engineering

Arizona State University

Tempe, AZ 85287

USA

Abstract

Packet radio networks are modeled as arbitrary graphs by most researchers. In this paper we show that an arbitrary graph is an inaccurate model of the radio networks. This is true because there exists a large class of graphs which will not model the radio networks. Radio networks can be modeled accurately by a restricted class of graphs called the planar point graphs. Since the radio networks can accurately be modeled only by a restricted class of graphs, the NP-completeness results for scheduling using an arbitrary graph as the model, do not correctly reflect the complexity of the problem. In this paper we study the broadcast scheduling problem using the restricted class as the model. We show that the problem remains NP-Complete even in this restricted domain. We give an $O(nlogn)$ *algorithm when all the transceivers are located on a line.*

Keywords: Packet radio network, Planar point graph, NP-complete problem, Broadcast scheduling, Transceiver.

1 Introduction

A packet radio network is a collection of radio transmitters and receivers (transceivers) located in a geographical region. Associated with each transceiver is a transmission range,

---

*Corresponding author, Telephone: 602-965-6153, Fax: 602-965-2751, e-mail: arunabha.sen@asu.edu

which depends on its transmission power. A transceiver T_j located within the transmission range of another transceiver T_i can receive the messages from T_i. The transceivers of the network share the communication medium for message transfer. This gives rise to conflict situations where two or more of the transceivers may want to use the same shared medium simultaneously. Such conflicts result in garbling and eventual loss of messages. Communication protocols are used to avoid such events. Such protocols may be divided into two broad classes, *static* and *dynamic*. In case of a *static* protocol an a priori assignment of the communication resources, such as time, frequency or code, is made among the transceivers to avoid contention. Dynamic protocols have no a priori resource allocation, instead resource requests are met on demand. *Time Division*, *Frequency Division* and *Code Division* multiple access (referred to as TDMA, FDMA and CDMA respectively) are examples of the a priori assignment of resources. ALOHA and CSMA are examples of dynamic protocols. Static allocation is well suited for a steady stream of traffic while dynamic allocation is well suited for a bursty traffic .

In this paper we restrict our attention to static allocation in general and TDMA in particular. However, it may be noted that the results presented in this paper are also valid for other static allocation schemes. In a packet radio network operating under a TDMA scheme, some transceivers are allowed to transmit simultaneously. However, the geographic locations and the transmitting powers of these transceivers should be such that not only do they not cause interference to each other but also do not cause interference at any other transceiver which may be within their transmission ranges. A *schedule* is a sequence of fixed length time slots where a slot is assigned to a set of non-interfering transceivers for transmission of messages. *Schedule length* is measured by the number of time slots in a schedule. A schedule is said to be *optimal* if it uses the minimum number of slots.

The problems associated with the construction of an optimal schedule have been studied extensively by researchers [1, 2, 3, 4, 5, 6, 7]. Most of these studies dealt with the construction of two different types of schedules, *broadcast schedules* and *link schedules*, under two different types of interferences, *primary interference* and *secondary interference* [1, 2]. In a broadcast schedule each transceiver is scheduled to ensure collision free transmission of messages to all transceivers within its range. In link scheduling, the trans-

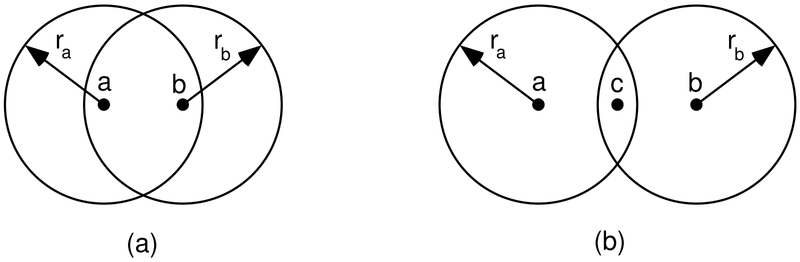

Figure 1: Type 1 and Type 2 Interferences

mission of a station is intended for only one specific transceiver within its range and the schedule needs to ensure that there are no collisions. Primary interference is said to occur when a transceiver is expected to perform more than one operation at the same time, such as receiving from two different transmitters at the same time or transmitting and receiving at the same time. Secondary interference occurs when a transmission from a neighboring transmitter unwittingly interferes at the receiving end of a communication between a transmitter and a receiver. Primary interference can be of two types. In fig. 1(a) the transceivers a and b are within the transmission range of one another. In this case if a and b start transmission at the same time, then both the transceivers will be expected to transmit and receive simultaneously. We refer to this as type 1 primary interference. In fig. 1(b) the transceivers a and b are not within the transmission range of one another, but there is a third transceiver c which is within the transmission range of both a and b. In this case if a and b start simultaneous transmissions then c will be expected to receive from both a and b at the same time. We refer to this as type 2 primary interference.

2 Prior work

For the purpose of constructing an optimal schedule all prior research [1, 2, 3, 4, 5, 6, 7] modeled a packet radio network as a graph, where a node in the graph represents a transceiver and there is a directed edge from the node v_i to the node v_j if the transceiver j is within the transmission range of the transceiver i. It may be noted that the resulting graph is a directed graph in which the existence of an edge from v_i to v_j does not necessarily imply the existence of an edge from v_j to v_i. However, in a network where the transmission range of all the transceivers are equal, existence of an edge from v_i to v_j will imply the existence of an edge from v_j to v_i. We will refer to such a network as an *uniform range* packet radio network and model it using an undirected graph.

Even [3] et. al. were among the earliest researchers to model a network as an arbitrary undirected graph to study the complexity of network testing under various conditions. Although their interest was primarily in network testing, under some conditions their testing problems were very close the optimal schedule construction problem. Using a slightly different definition of interference they showed that optimal schedule construction problems were NP-hard. Chlamtac et. al. [8, 9] also studied the scheduling problem of multi–hop radio networks using the same model.

Hajek and Sasaki [5] using the same network model presented two polynomial time algorithms for link scheduling in a spread spectrum radio network. The ability of spread spectrum modulation to allow multiple conversations to occur at the same time and place enabled the development of the polynomial algorithm, whereas in a single frequency narrow band system, the problem turns out to be an NP-hard scheduling problem [4]. Ogier in [10] presented more efficient algorithms for link scheduling than the algorithms proposed by Hajek and Sasaki.

Arikan [4] used a directed graph to model packet radio networks and showed that the decision problem regarding the membership of a node in the *capacity region* (the set of all origin-to-destination message rates that are achievable via any arbitrary protocol) of the network is NP-hard.

Ramaswami and Parhi [6] using an arbitrary undirected graph model of the network proved that the construction of a minimum length schedule is an NP-complete problem. The decision problem they proved to be NP-complete is stated as follows: Given an undirected graph $G = (V, E)$ and an integer constant k, can this graph be colored using at most k colors such that the node pairs that are connected by a path of length at most two have different colors? Ephremides and Truong in [7] essentially proved the same NP-completeness results as Ramaswami and Parhi.

Although Bar-Yehuda et.al. in [11] commented that most packet radio networks can be modeled as *trees*, Ramanathan and Lloyd [1, 2] were the first to suggest the use of a restricted class of graphs as network models. Since it is not too difficult to see that not all packet radio networks can be modeled as trees, they suggested the use of another class of graphs - the *planar graphs*, as network models. They showed that if the network can be modeled as a tree, then the optimum schedule can be constructed in polynomial time. However, the problem remains NP-complete even when the radio network is modeled as a planar graph instead of an arbitrary graph.

3 Network Model

The discussion in the previous section clearly shows that, with the sole exception of [1, 2], almost all prior research in this area used an arbitrary graph (either directed or undirected) as the model of the radio network. Many scheduling problems were shown by the researchers to be NP-complete using this model. As indicated in the previous section in connection with Ramaswami and Parhi's research, most of the scheduling problems are formulated as *graph coloring* problems. Broadcast scheduling problems are generally set up as the node coloring problem and the link scheduling problem is set up as the edge coloring problem [1, 2]. These problems were shown to be NP-complete in [1, 2, 3, 4, 6, 7].

It is very important to realize that *if a problem is NP-complete in a certain domain, it does not imply that the problem will continue to be NP-complete when the domain is substantially restricted.* More specifically, *NP-completeness of a graph problem does not necessarily imply the NP-completeness of the problem for a restricted class of graphs.* A classic example is the *chromatic number* problem, where we are required to find the minimum number of colors necessary to color the nodes of the graph such that no two adjacent nodes have the same color. It is well known that this problem is NP-complete [12]. However, if we restrict the class of graphs to be *connected* and *cycle free*, then the problem becomes trivial. The chromatic number of all non trivial (a graph with more than one node) connected, cycle free graphs, (i.e., trees) is two.

Models are designed to encapsulate the essential features of the system under study. The use of an arbitrary graph (directed or undirected) as the model of a radio network fails to capture some essential features of the network.

At the first level of abstraction of the radio networks, we think of the transceivers as some points (specified by their x, y-coordinates) on a two dimensional plane. Associated with each point is a range value, representing the transmission range of the transceiver.

At the second level of abstraction, we construct a graph where each node represents a point on the plane (transceiver) and there is a directed edge from node v_i to node v_j, if the range r_i, associated with point p_i exceeds or equals the Euclidean distance between the points p_i and p_j on the two dimensional plane. The graph constructed in this manner captures all the pertinent features of the radio network. This mechanism of constructing the graph model for a radio network was used by many other researchers, notably [1, 2, 4, 6]. Although, this graph construction mechanism was used, the earlier researchers assumed that any arbitrary graph could be constructed using this mechanism and based their NP-completeness proofs of the scheduling problems on this assumption.

In this paper we show that *the assumption is incorrect.* The class of graphs that will be generated by the above construction mechanism forms a *proper* subset of all graphs. Ramanathan and Lloyd [1, 2] were the first to recognize this possibility and, as a result, they restricted their attention to planar graphs and trees. However, in this paper we show that *the graphs generated by the above construction mechanism are not arbitrary graphs or planar graphs or trees.*

Point graphs are introduced in this paper for the purpose of modeling packet radio networks. Given a set of points $P = (p_1, ..., p_n)$ in an m-dimensional space, each specified by an m vector $(x_1, ..., x_m)$ and a range r_i associated with each p_i, the *point graph* constructed from this set of points is a graph $G = (V, E)$, where each node v_i represent a point p_i and there is a directed edge from node v_i to v_j if $d(p_i, p_j) \leq r_i$, where $d(p_i, p_j)$ is the Euclidean distance between the points p_i and p_j. *Planar point graphs* are point graphs constructed from the points on a two dimensional plane ($m = 2$). *Linear point graphs* are point graphs of points on a line ($m = 1$). *Symmetric point graphs* are point graphs where for all nodes v_i,v_j, there is a directed edge from v_i to v_j, if and only if there is a directed edge from v_j to v_i. Otherwise, the graph is an *asymmetric point graph.* Symmetric point graphs are drawn as undirected graphs by replacing two oppositely directed edges by an undirected edge.

It may be noted that the point graphs are generalizations of *space graphs* and *unit disk graphs* introduced in [13] and [14] respectively. If the range value r_i associated with a point p_i is equal for all points then a point graph reduces to a space graph. A space graph of dimension two is a unit disk graph. The relationship between point graphs, space

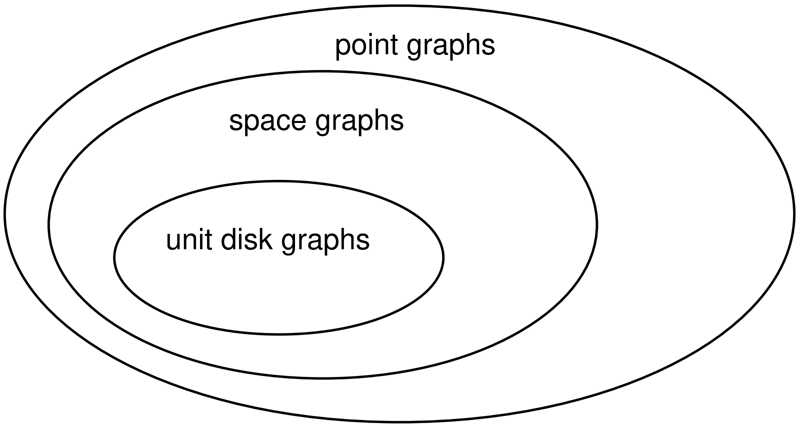

Figure 2: Relations between point graphs, space graphs and unit disk graphs

graphs and unit disk graphs is shown in fig. 2.

The *Location-Range* representation of an n node point graph is a set of n tuples (l_i, r_i), $1 \leq i \leq n$, where l_i represents the location of point p_i and r_i represents the range associated with the point. If p_i is a point in an m-dimensional space, its location l_i is specified by an m vector $(x_1, ..., x_m)$.

It is clear from the above definitions that planar point graphs capture the pertinent features of a radio network. We show that the modeling of a radio network using an arbitrary graph is inappropriate because planar point graphs are a proper subset of the set of all graphs. We demonstrate this by providing several graphs (fig. 3(a)-(g)) which are not planar point graphs. These graphs are some of the *forbidden subgraphs* of planar point graphs, i.e., any graph that contains a graph from fig. 3(a)-(g) as node induced subgraph cannot be a planar point graph. Next we explain why the graph in fig. 3(a) is not a planar point graph.

We assume that all transmitters are omnidirectional. We draw a circle with the transmitter position at the center and radius equal to the transmission range of the transmitter. This circle represents the area covered by the transmitter. We refer to this circle as the *transmission circle.*

Since the nodes of the graph correspond to the transceivers, transceiver 1 is located in a plane with its transmission circle as shown in fig. 4(a). Since transceiver 2 receives messages from transceiver 1, it must be within 1's transmission range. Moreover, as 2's transmission reaches 1, 1 must be within 2's transmission range. Fig. 4(b) shows 2's

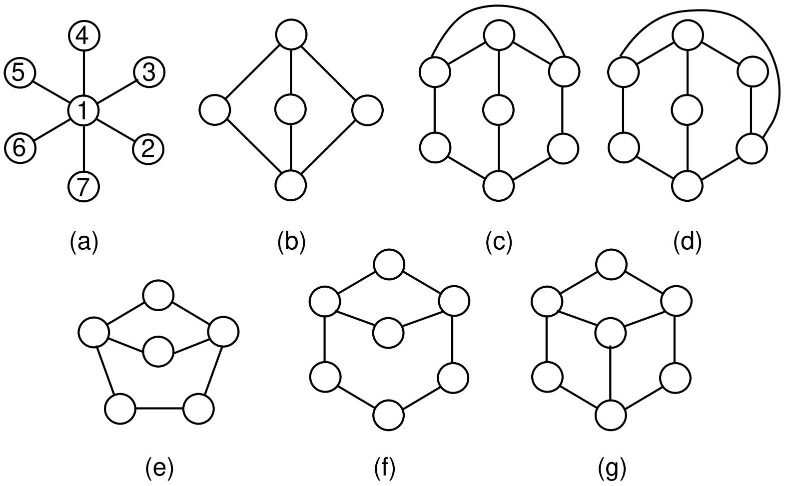

Figure 3: Forbidden subgraphs of planar point graphs

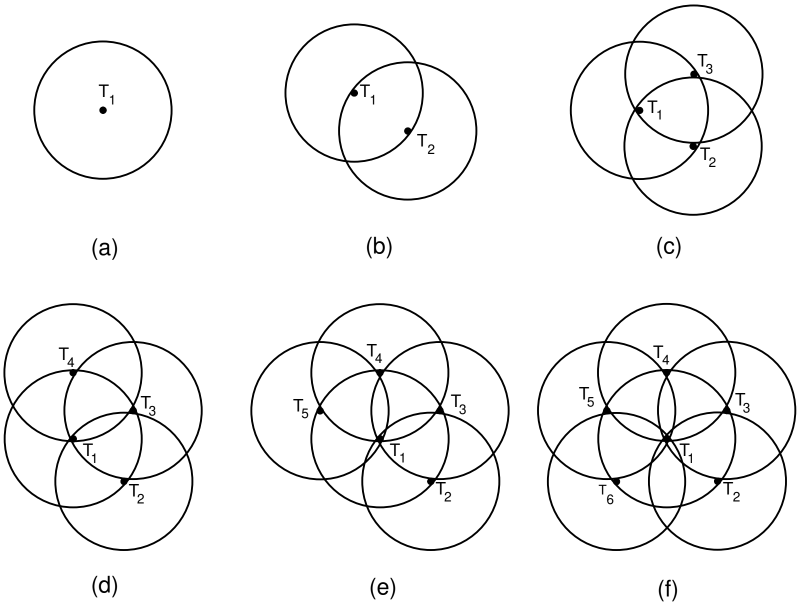

Figure 4: Impossibility of fig. 3(a) being a planar point graph

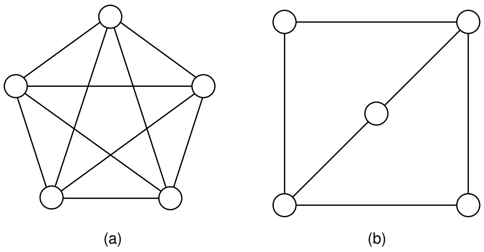

Figure 5: Planar point graph and Planar graph

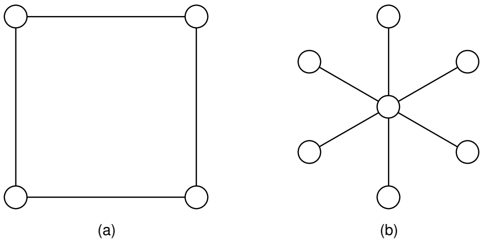

Figure 6: Planar point graph and Tree

transmission circle overlapped on the transmission circle of 1. Transceivers 1 and 3 are within one another's transmission range but 2 and 3 are not within range. A possible position of transmitter 3 and it transmission circle is shown in fig. 4(c). Proceeding in this way, we show the position of transceivers 1,2,3, and 4 in fig. 4(d), the position of transceivers 1,2,3,4, and 5 in fig. 4(e) and the position of transceivers 1,2,3,4,5, and 6 in fig. 3(f). At this time it is clear that there is no room for the placement of transmitter 7, without violating the interconnection pattern displayed in the graph shown in fig. 3(a). We thus conclude that fig. 3(a) is not a planar point graph. Similar explanations can be provided for all the graphs of fig. 3.

Ramanathan and Lloyd realized the inappropriateness of modeling radio networks with arbitrary graphs and proposed the use of planar graphs and trees as the models. However, we show that planar point graphs are distinctly different from planar graphs and trees. Fig. 5(a) shows a planar point graph that is not a planar graph and fig. 5(b) shows a planar graph but not a planar point graph. Fig. 6(a) shows a planar point graph that is not a tree and fig. 6(b) shows a tree but not a planar point graph. The relations between arbitrary graphs, planar graphs, planar point graphs and trees are shown in fig. 7.

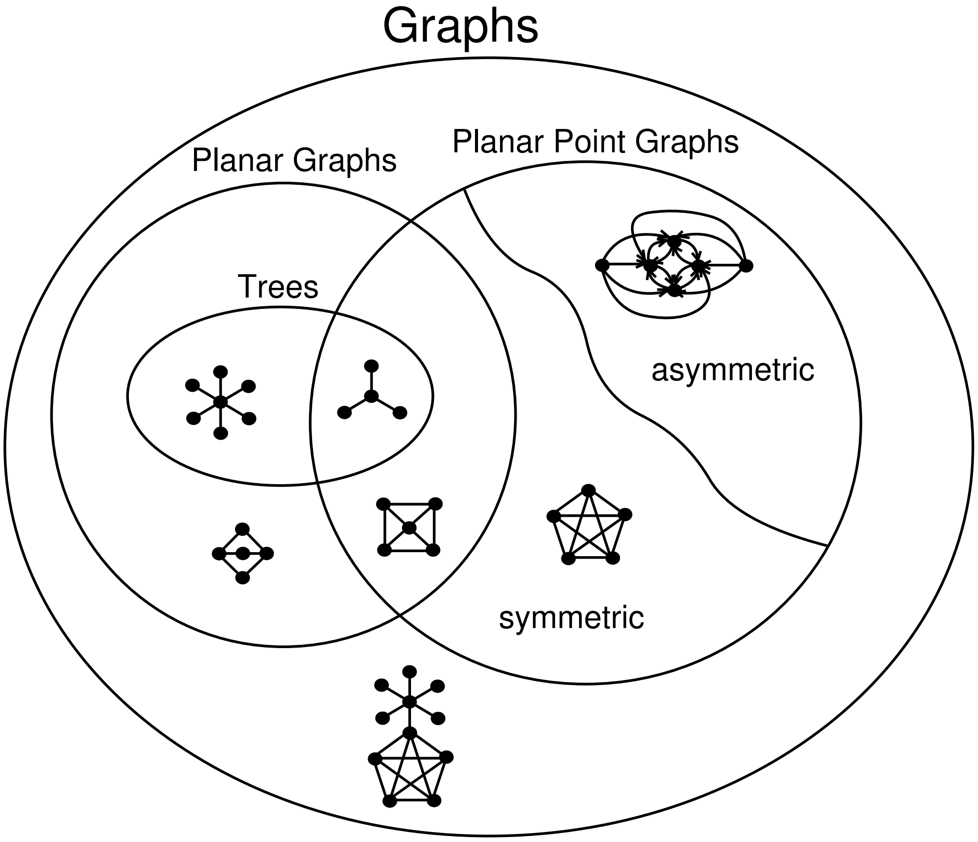

Figure 7: Relations between planar point graphs, planar graphs and trees

4 Complexity of Broadcast Scheduling

In the previous section we demonstrated the appropriateness of using planar point graphs as models of packet radio networks. The optimal broadcast schedule construction (OBSC) problem can now be stated in terms of coloring the nodes of a planar point graph. As discussed earlier, any broadcast schedule must be free of primary interference of type 1 and 2. In terms of the model, the problem can be stated as follows:

Optimal Broadcast Schedule Construction (OBSC) Problem

INSTANCE: Given a planar point graph $G=(V,E)$ and an integer constant k.

QUESTION: Is there a partition $V=V_1 \cup \ldots \cup V_k$ such that if u and v are distinct nodes of some set V_i in the partition then (i) there is no edge from u to v, (ii) there is no edge from v to u and (iii) there is no node w in the graph G such that there is an edge from u to w and from v to w?

We will also refer to the OBSC problem as either *distance-2 k-colorability of planar point graphs* or *distance-2 coloring of planar point graphs.*

A similar problem was considered by McCormick [15]. The *distance-d chromatic number* χ_d, of a graph $G = (V, E)$ is the smallest integer k for which the node set V can be partitioned into $V_1, ..., V_k$ such that if u and v are distinct nodes in some set V_i in the partition, then there is no path from u to v in G of length d or less. He proved that distance-d chromatic number problem for an arbitrary graph is an NP-complete problem [15, 16].

In this paper we show that the OBSC problem is NP-complete. It may be noted that the NP-completeness of the OBSC problem does not immediately follow from McCormick's results because his results are true for arbitrary graphs whereas the OBSC problem relates to one specific class of graphs - the planar point graphs.

A geometric version of the distance-2 k-colorability of planar point graphs can be stated as follows:

Optimal Broadcast Schedule Construction (OBSC) Problem (Geometric Version)

INSTANCE: Given (i) a set of points $P = (p_1, ..., p_n)$ in a 2-dimensional plane, each specified by their (x, y)-coordinates, (ii) a range r_i associated with each p_i and (iii) an integer constant k.

QUESTION: Is there a partition $P = P_1 \cup ... \cup P_k$ such that if u and v are distinct points in some set P_i of the partition, then (i) $d(u, v) > max(r_u, r_v)$, where $d(u, v)$ represents the Euclidean distance between the points u and v and (ii) there is no point w in the set P such that $d(u, w) \leq r_u$ and $d(v, w) \leq r_v$)?

It is easy to see that the two versions of the OBSC problem are equivalent. Next we show that the OBSC problem is NP complete.

We prove the distance-2 k-colorability of planar point graphs NP-complete by proving the distance-2 3-colorability of planar point graphs NP-complete.

Distance-2 3-Colorability of Planar Point Graph Problem

INSTANCE: Given a planar point graph $G = (V, E)$.

QUESTION: Is there a partition $V = V_1 \cup V_2 \cup V_3$ such that if u and v are distinct nodes of some set V_i in the partition then (i) there is no edge from u to v, (ii) there is no edge from v to u and (iii) there is no node w in the graph G such that there is an edge from u to w and from v to w?

Theorem 1 *The distance-2 3-colorability of planar point graphs is NP-complete.*

Proof: We prove the distance-2 3-colorability of planar point graphs NP-complete by giving a polynomial time transformation from the *planar graph 3-colorability with node degree at most 4* problem which was shown to be NP-complete by Garey, Johnson, and Stockmeyer in [17].

It is easy to see that the distance-2 3-colorability of planar point graphs is in NP, since a non-deterministic algorithm can be used to assign colors to the nodes of the graph and check in polynomial time if nodes u and v are assigned the same color then (i) there is no edge from u to v, (ii) there is no edge from v to u and (iii) there is no node w in the graph G such that there is an edge from u to w and v to w.

We use the following result due Valiant [18, 14] in our proof.

> *A planar graph with maximum degree 4 can be embedded in the plane using $O(|V|)$ area in such a way that its vertices are at integer coordinates and its edges are drawn so that they are made up of line segments of the form $x=i$ or $y=j$, for integers i and j.*

We also use a result due to Biedl and Kant [19] where they present a linear time algorithm which produces an embedding of graph $G = (V, E)$ on a grid such that each vertex is at a grid point and each edge is composed of vertical and horizontal segments with at most *two bends.*

To prove the NP-completeness of the distance-2 3-colorability of planar point graphs we need to generate an instance of the problem which is a planar point graph. To ensure that the graph is indeed a planar point graph we go back one step further and generate from an instance of the planar graph 3-colorability with node degree at most 4 problem, a set of points on the plane (specified by their x, y coordinates) and a range value associated with each point. The graph constructed from this set of points and their associated range values will indeed be a planar point graph.

We start with an arbitrary instance of a planar graph with node degree at most 4, and use Biedl's algorithm to produce an embedding in the grid. We scale the embedding by a factor of three to ensure that all pairs of points in the embedding are separated by a distance of at least three units.

There will be two categories of points which will make up the instance of the distance-2 3-colorability of planar point graph problem. Corresponding to each embedded vertex of the planar graph on the grid, there will be a *vertex replacement* point whose (x, y) coordinates will be specified by its position in the grid. Corresponding to each edge of the planar graph (which is embedded in the grid by a sequence of horizontal and vertical segments) there will be a set of *edge replacement points*. The (x, y) coordinates of the edge replacement points will be specified in the next paragraph. There will be two different types of edge replacement points: A and B. The range value associated with each vertex replacement point is 1 unit. Each type A edge replacement point has a range of 0.25 units and each type B edge replacement point has a range of 0.45 units.

The (x, y) coordinates of the edge replacement are computed as follows: Suppose u and v are two adjacent nodes of the planar graph that makes up an instance of the planar graph 3-colorability with node degree at most 4 problem. Also suppose that in the embedding of the graph on a grid the node u is located at the grid point (0,0) and the point v at the point (0,1). Since we scale the embedding by a factor of 3, after scaling, we may assume that u is located at point (0,0) and v at point (0,3).

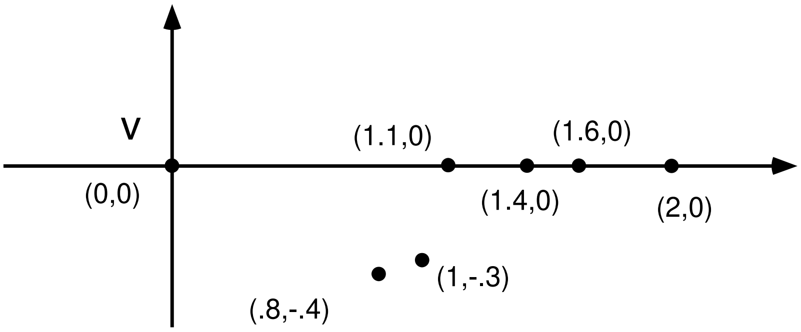

Figure 8: Edge replacement point positions relative to the start vertex v and the first segment of the edge being replaced.

Type A edge replacement points corresponding to the edge (u, v) of the planar graph will be located at coordinates (0.8, -0.4) and (1, -0.3). with a type B point at (1.1,0). There will be two more type A points at (1.4,0) and (1.6,0) followed by a type B point located at (2,0). In case the embedding of the edge between u and v takes more than one segment of the grid then the pattern is repeated for each additional segment with two more type A points located at (x.4,0) and (x.6,0) and a type B point located at (x+1,0), where x is an integer number. For example if in the embedding u is located at the grid point (0,0) and the point v at the point (2,0) then after scaling u will be located at point (0,0) and the v at point (6,0). There will be vertex replacement points corresponding to the nodes u and v at locations (0,0) and (6,0). There will be type A edge replacement points at locations (0.8, -0.4), (1,-0.3), (1.4,0), (1.6,0), (2.4,0), (2.6,0), (3.4,0), (3.6,0), (4.4,0) and (4.6,0). There will be type B edge replacement points at locations (1.1, 0), (2,0), (3,0), (4,0) and (5,0).

Figure 9 shows the edge replacement for an edge between two vertex replacement points 3 units apart in the scaled embedding.

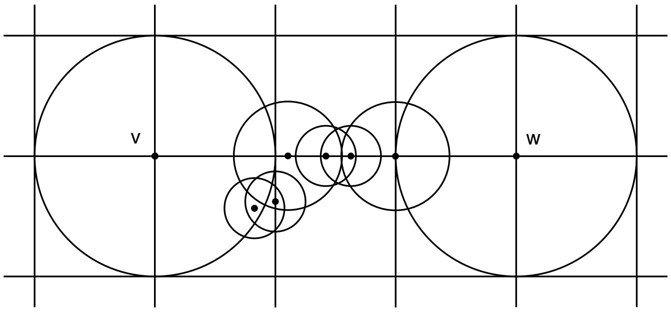

Figure 9: Complete edge replacement between vertex replacement points v and w, 3 units apart in the scaled embedding.

In another example, assume the adjacent nodes u and v of the planar graph in their grid embedding are located at positions (0,0) and (1,-2) (fig. 10) and the edge is embedded by three segments ((0,0),(1,0)),((1,0),(1,-1)) and ((1,-1),(1,-2)), after scaling u will be in (0.0) and v will be in (3,-6) and the edge will be embedded by nine segments as shown in fig. 10 The vertex replacement points will have coordinates (0,0) and (3,-6). Type A edge replacement points will be in locations (0.8.-0.4), (1,-0.3), (1.4,0), (1.6,0), (2.4,0), (2.6,0), (3,-0.4), (3,-0.6), (3,-1.4), (3,-1.6), (3,-2.4), (3,-2.6),(3,-3.4), (3,-3.6),(3,-4.4) and (3,-4.6). Type B edge replacement points will be in locations (1.1,0), (2,0), (3,0), (3,-1), (3,-2), (3,-3), (3,-4) and (3,-5).

The edge replacement points are generated in such a way that in the resulting planar

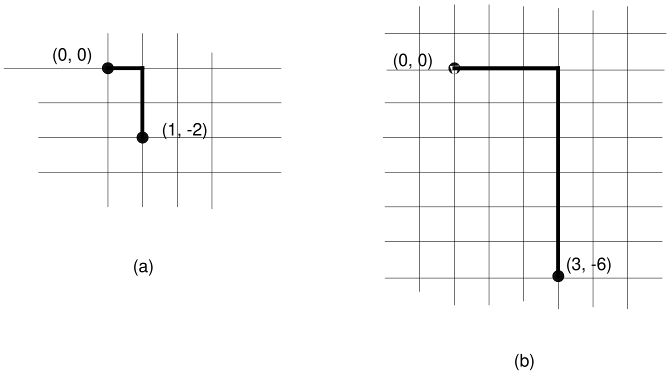

Figure 10: Embedding of an edge in a grid before and after scaling

point graph they create a specific directed edge pattern. If p and q are two adjacent nodes in the planar graph and in the grid embedding the edge (p, q) is realized by one grid segment (before scaling), then the edge replacement points are created in such a way that the resulting planar point graph has the directed edge pattern as shown in fig. 11. The nodes p_1 and q_1 in the planar point graph correspond to the vertex replacement points corresponding to the nodes p and q of the planar graph. The u_i and v_i nodes correspond to type A edge replacement points for the edge (p, q) of the planar graph and the w_i nodes correspond to the type B edge replacement points. If the edge (p, q) is realized by s grid segments in the scaled embedding, then the edge replacement points are created in such a way that in the resulting planar point graph the directed edge pattern in one box of fig. 11 is repeated $s - 1$ times as shown in fig. 12.

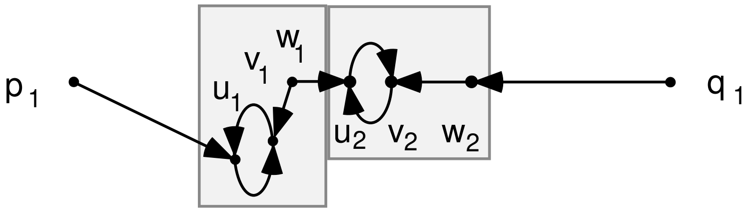

Figure 11: Interconnection pattern in the planar point graph generated from an edge of the planar graph when the adjacent nodes are connected by 3 segments in the scaled grid embedding.

It may now be observed that in any distance-2 3-coloring of the generated instance of the planar point graph, the colors at nodes u_1 and v_1 should be different from each other

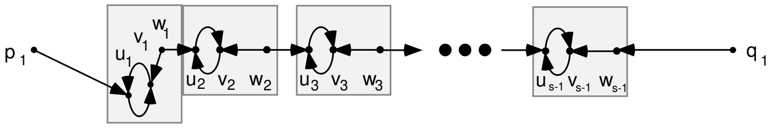

Figure 12: Interconnection pattern in the planar point graph generated from an edge of the planar graph when the adjacent nodes are connected by s segments in the scaled grid embedding.

and also different from the node p_1. This forces the node w_1 to have the same color as the node p_1. In fact if the scaled embedding of the edge (p, q) of the planar graph is realized by s grid segments then all the w_i nodes $(1 \leq i \leq s-1)$, are forced to have the same color as the node p_1 in the distance-2 3-coloring of the planar point graph. This is shown in figure 13.

As figure 9 and figure 8 show, the combination of positions and ranges of type A and type B edge replacement points requires a scaling factor of 3.

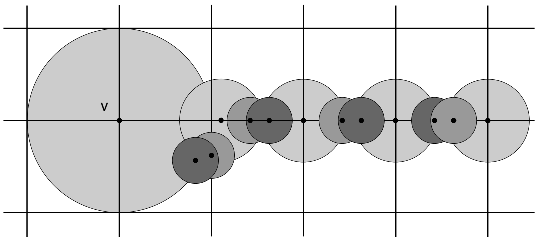

Figure 13: 3–coloring of vertex replacement point, v, and an edge replacement starting at that point. The type–1 and type–3 points are the same color.

To show that this is a viable replacement process two properties must be maintained. First it must be possible to make a "bend" in the edge without changing the arc pattern shown in fig. 11 (recall Biedl and Kant showed each edge replacement will have at most two bends). A "bend" in the embedded edge is a 90° turn, a change from vertical to horizontal or horizontal to vertical trace.

The second property is that the edge constructs must allow each node of the planar graph to have four non-interfering edge replacements. Figure 15 shows one such node.

For each edge of the planar graph in its scaled embedding in the grid with embedded segments of length s, there will be $3(s-1)$ points in the instance of the *distance-2 3-*

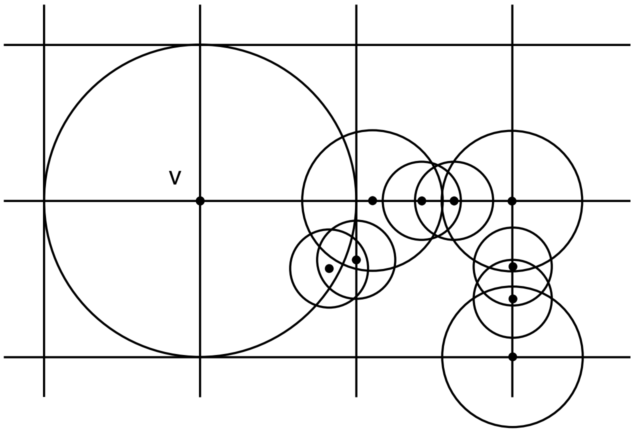

Figure 14: An edge construct with a 90 degree bend, showing that such bends are possible without causing interference

colorability problem. If m is the maximum length of any edge embedding of the planar graph, there are then $O(|V|+3m|E|)$ nodes in the instance of planar point graph produced by the transformation. The total time complexity to produce the instance of *distance-2 3-colorability problem* is a polynomial function of the size of the input.

Let G be an instance of the planar graph 3-colorability with node degree 4 problem and G' be the instance of the planar point graph constructed from G using the above construction rules. We need to show that G is 3-colorable if and only if G' is 3-colorable.

Suppose G is 3-colorable. Then we can assign colors to the nodes of G' in the following way: the nodes of G' that correspond to the vertex replacement points (corresponding to the nodes of G) are assigned the same color as the corresponding node of G. If p and q are two adjacent nodes in the planar graph and p_1 and q_1 are the corresponding nodes in the planar point graph, then p_1 has same color as p and q_1 has same color as q. In the planar point graph there are a number of nodes between p_1 and q_1 corresponding to the edge replacement points. If, ignoring the orientations of the edges for a moment, we consider a path between the nodes p_1 and q_1, only one of these nodes will have a neighboring node on this path which corresponds to a type A edge replacement point. The other node will have a neighboring node on this path which corresponds to a type B edge replacement point. Suppose the node p_1 has a type A neighbor. The undirected path from p_1 to q_1 will then be p_1 followed by a type A neighbor, followed by a second type A neighbor, followed by a type B neighbor and so on. Without loss of generality, if we assume that

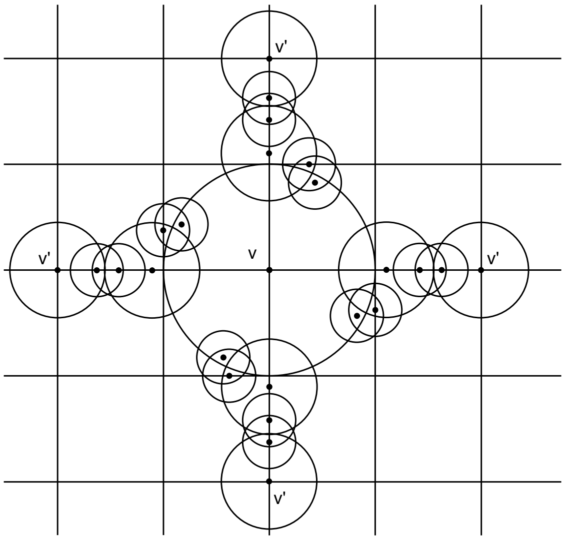

Figure 15: The construction for a vertex, v, of degree 4 from the original planar graph G. Each vertex, v' has the same color as vertex v in a valid distance–2 3–coloring of the constructed graph, G'.

the node p_1 is assigned color 1, then its first type A neighbor may be assigned color 2 and the second type A neighbor may be assigned color 3. The next node in the path - the type B neighbor may again be assigned color 1 without violating the requirements of the distance-2 coloring. Proceeding this way the last type B node in the path (neighbor of node q_1) will be assigned color 1. Since the node q_1 has the same color as node q of graph G, the color of node q_1 must be different from color 1. (This is true because the color of p_1 is same as color of p and color of q_1 is same as color of q, p and q are adjacent nodes in G and G is 3-colorable). It is clear that this coloring scheme will not violate any distance-2 coloring requirement. Therefore, G' is distance-2 3-colorable.

Now suppose that G' is distance-2 3-colorable. Consider the set of nodes of G' that correspond to the vertex replacement points (note, there is a one–to–one correspondence between the nodes of G and the vertex replacement points). Assign to the nodes of G the same colors as are assigned to the corresponding nodes of G'. Suppose, if possible, in this assignment two adjacent nodes p and q of G are colored using the same color - a clear violation of the coloring requirement. This implies that the corresponding nodes p_1 and q_1 in G' must have the same color. Due to the nature of the construction of G', the only way this can happen without violating the requirements of a distance-2 coloring, is if 4 different colors are used to color the nodes corresponding to the edge replacement points that appear between the nodes p_1 and q_1. Since G' is distance-2 3-colorable, this is impossible. Therefore G is 3-colorable. □

An example of a planar graph, its embedding in a grid and the locations of the A and B type edge replacement points are shown in fig. 16.

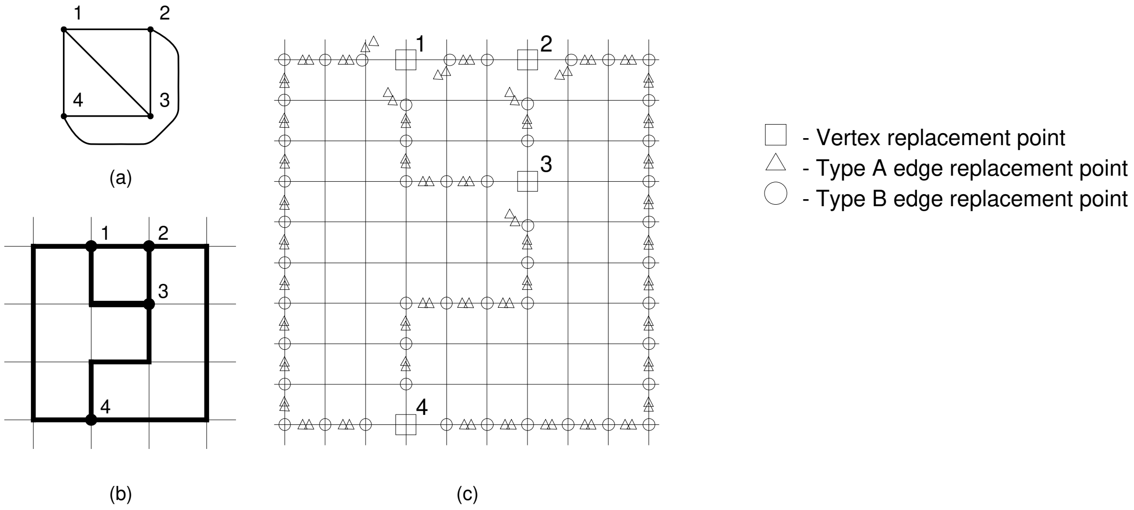

Figure 16: Planar graph, its embedding in grid and the locations of vertex and edge replacement points.

5 Optimal Schedule for Transceivers on a Line

In this section we present an algorithm for the optimal broadcast schedule construction problem when the transceivers are placed on a line. In the last section we showed the more general problem of optimal schedule construction when the transceivers are located in a two dimensional plane is NP-complete. In this section we show that in this restricted case where the transceivers are located on a line, the problem can be solved using a polynomial time algorithm. The model of the radio network with transceivers located on a line is a *linear point graph* introduced in section 3. The optimal schedule construction problem now reduces to distance-2 coloring problem of linear point graph. A variation of McCormick's [15] definition of *distance-d chromatic number* of a graph is used to define distance-2 chromatic number of a *directed* graph. The distance-2 chromatic number of a *directed* graph is the smallest integer k for which the node set V can be partitioned into $V_1, ..., V_k$ such that if u and v are distinct nodes in some set V_i in the partition, then (i) there is no directed edge from u to v, (ii) there is no directed edge from v to u and (iii) there is no node w in the graph G such that there is a directed edge from u to w and v to w. The following algorithm computes the distance-2 chromatic number of a linear point graph.

The location-range representation of the linear point graph is provided as the input. Since in this case the location is specified by a scalar, the location-range representation is a set of n ordered pairs, where the first number provides the location of the point (transceiver) and the second number provides the range.

Our algorithm is based on the interval graph coloring algorithm used for VLSI channel assignment by Gupta, Lee, and Leung in [20].

First, from the location and the range of a point a *transmission interval* associated with the point is computed. If a point i is located at p_i and has a range r_i, then its transmission interval spans from $p_i - r_i$ to $p_i + r_i$. Thus corresponding to n nodes of the graph we have $3n$ points, of which n are left end points of the transmission intervals, n are right end points, and n are positions of the points on the line. The points are numbered from 1 through n. Without loss of generality we assume that $p_i < p_j$ if $i \; < \; j$. The left and right end points of the point i will be denoted by L_i and R_i respectively.

The algorithm scans the $3n$ points from left to right. Suppose Π is a permutation of the numbers 1,...,n such that $L_{\Pi(1)} \leq L_{\Pi(2)} \leq ... \leq L_{\Pi(n)}$. The nodes of the graph are colored in the following order: $\Pi(1), \Pi(2), ..., \Pi(n)$. As soon as the left endpoint of the transmission interval for point i is encountered during the left to right scan of the $3n$ points, the point i is put on a list of points waiting to be colored. All the points in this waiting list are assigned a color one after another as soon as the position of a point is encountered during the scan. The nodes $\Pi(s), \Pi(s+1), ..., \Pi(t)$ are assigned colors when the algorithm while scanning the points from left to right reaches the first point q at position p_q such that $p_{q-1} < L_{\Pi(s)} \leq L_{\Pi(s+1)} \leq ... \leq L_{\Pi(t)} \leq p_q$. At this time the nodes $\Pi(s), \Pi(s+1), ..., \Pi(t)$ are all assigned different colors. When the right endpoint of an interval is encountered, the color used to color the corresponding interval is freed up and is made available to color the next node.

Algorithm for distance-2 coloring of linear point graphs

```
Step 1: Sort the 3N endpoints so $z_i \leq z_{i+1} \ldots z_{3N}$.[1]
Step 2: $COUNTER \leftarrow 0$, $MAXCOLOR \leftarrow 0$
    $COLOR \leftarrow 1$, $NUMTOPROC \leftarrow 0$
Step 3: for i = 1 to N do:
    $NEXT(i) \leftarrow i + 1$
Step 4: for j = 1 to 3N do:
    if $z_j$ is a left endpoint, $x_k$
        $NUMTOPROC \leftarrow NUMTOPROC + 1$
        $WAITING(NUMTOPROC) \leftarrow k$
    else if $z_j$ is a right endpoint $y_k$
        $COUNTER \leftarrow COUNTER - 1$
        $TEMP \leftarrow ASGN(k)$
        $NEXT(TEMP) \leftarrow COLOR$
        $COLOR \leftarrow TEMP$
    else $z_j$ is a position point $p_k$
        for i = 1 to NUMTOPROC do:
            $COUNTER \leftarrow COUNTER + 1$
            $MAXCOLOR \leftarrow max\{COUNTER, MAXCOLOR\}$
            $ASGN(WAITING(i)) \leftarrow COLOR$
            $COLOR \leftarrow NEXT(COLOR)$
        $NUMTOPROC \leftarrow 0$
```

Theorem 2 *The algorithm for distance-2 coloring of linear point graphs optimally colors the nodes of the graph.*

Proof: From the definition of distance-2 chromatic number of a graph, two distinct nodes u and v should not be assigned the same color if any one of the following conditions holds: (i) there is a directed edge from u to v, (ii) there is a directed edge from v to u

---

[1] If $z_i = z_{i+1} = \ldots = z_j$, the left endpoints precede position points which precede right endpoints.

(iii) there is a node w in the graph G such that there is a directed edge from u to w and a directed edge from v to w.

First, we show that the algorithm colors the nodes of the graph in such a way that none of the conditions (i), (ii) or (iii) is violated.

Case 1: Consider two nodes i and j of the graph with $i < j$ (from our assumption this implies $p_i < p_j$). Suppose that there is a directed edge from i to j. This implies $R_i > p_j$. Suppose a color has been assigned to the node i and a color is about to be assigned to the node j. The algorithm in its scan of points from left to right at this time has encountered the points L_i and L_j but has not encountered R_i. In this situation the algorithm will assign a color to j which is different from that was assigned to i.

Case 2: Consider nodes i and j of the graph with $i < j$ and suppose that there is a directed edge from j to i. This implies $L_j < p_i$. Suppose a color has been assigned to the node i and a color is about to be assigned to the node j. The algorithm in its scan of points from left to right at this time has encountered the points L_i and L_j but has not encountered R_i. In this situation the algorithm will assign a color to j which is different from that was assigned to i.

Case 3: Consider nodes i and j of the graph with $i < j$ and suppose that there is no directed edge from i to j or from j to i but there is a node k in the graph such that there is a directed edge from i to k and a directed edge from j to k. This implies $R_i > p_k$ and $L_j < p_k$. Suppose a color has been assigned to the node i and a color is about to be assigned to the node j. The algorithm in its scan of points from left to right at this time has encountered the points L_i and L_j but has not encountered R_i. In this situation the algorithm will assign a color to j which is different from the one assigned to i.

Therefore the algorithm colors the nodes of the linear point graph without violating the distance-2 coloring requirements (i), (ii) and (iii).

If χ_{opt} is the optimum number of colors necessary to distance-2 color the nodes of the linear point graph then $\chi_{opt} > max(d_{in}(v))$ over all nodes of the graph, where $d_{in}(v)$ is the number of incoming edges to node v. This must hold to satisfy requirement (iii) of the distance-2 coloring of the graph.

We define the *crossing number* of a point i, $c(i)$ to be the cardinality of the set of transmission intervals that span over the point i (i.e., $c(i) = |S(i)|$, where $S(i) = \{j|L_j \leq$

$p_i \leq R_j\}$). If χ is the number of colors used by the algorithm to distance-2 color the nodes of the linear point graph, then $\chi = max(c(i))$, $1 \leq i \leq n$, where n is the number of nodes in the graph. If $\chi = c(k)$ for some k, $1 \leq k \leq n$, the indegree of the node k of the graph is $c(k) - 1$.

Since this algorithm distance-2 colors using χ colors and there exists a node in the graph whose indegree is $\chi - 1$, no algorithm can distance-2 color this graph with fewer than χ colors. This implies $\chi_{opt} = \chi$ and this algorithm optimally distance-2 colors the nodes of the linear point graph. □

The complexity of the algorithm is $O(nlogn)$ which is determined by the sorting time of the $3n$ points.

6 Approximate Schedules for Transceivers on a Plane

In the previous sections we showed that the optimal schedule construction is possible in polynomial time if the transceivers are located on a line, while the problem is NP-complete if the transceivers are on a plane. We develop the idea of an approximation algorithm for the NP-complete problem from this observation. The idea is to *project* the points from a 2-dimensional plane to 1-dimension (i.e. a line) and then use the optimal algorithm presented in section 5 to construct the schedule. If u and v are two points on the plane and u' and v' the corresponding projected points on a line, then $d(u', v') \leq d(u, v)$, where $d(u, v)$ is the Euclidean distance between the points u and v. For this reason, it may be possible that although $d(u, v) > max(r_u, r_v)$, $d(u', v') \leq max(r_u, r_v)$, where r_u is the range associated with point u. This implies that the schedule constructed from the projected points may have more slots than the optimum schedule. So this method will only produce an approximate schedule. The difference in the number of slots used in the approximate schedule to the optimum schedule is the *error* introduced in the approximation process. Obviously, we would like to make the error as small as possible. Consider a set of three uniform range transceivers located at points (1,1), (2.1,1) and (1.5,4) The range of the transceivers is 1. The optimum schedule for these transceivers will have only one slot. If the points are projected on the line X-Y as shown in fig. 17 the schedule will have three slots, but if they are projected on the line S-T the schedule will have two slots. Thus the error will clearly depend on the choice of the line on which the points are being projected. The projection line that will introduce the least amount of error will be called the *best projection line*.

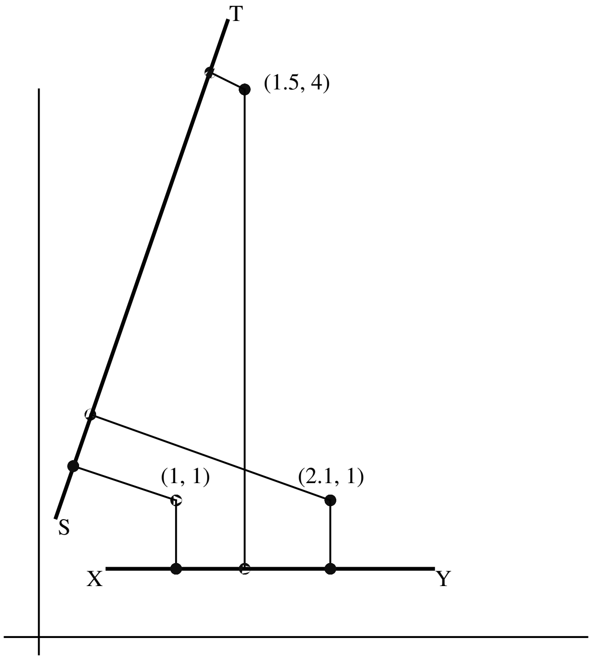

Figure 17: Projection of the points on line X-Y and S-T

6.1 Computation of Best Projection Line

In this section we describe a technique for finding the best projection line from a given set of n points on a two dimensional plane specified by their (x, y) coordinates and the range values associated with them.

Consider a circle enclosing the n points as shown in fig. 18 and two points i and j inside the circle. If the Euclidean distance between i and j, $d(i, j) \leq min(r_i, r_j)$ then in the planar point graph constructed from these set of points, there is a directed edge from the node v_i to node v_j and a directed edge from v_j to v_i. If i' and j' are the projected points on a line corresponding to i and j, then $d(i', j') \leq d(i, j)$. As a result in the linear point graph constructed from the set of projected points there is a directed edge from the node $v_{i'}$ to node $v_{j'}$ and a directed edge from $v_{j'}$ to $v_{i'}$. In this situation the projection line has no impact on the connection pattern between the nodes of the graph. For this

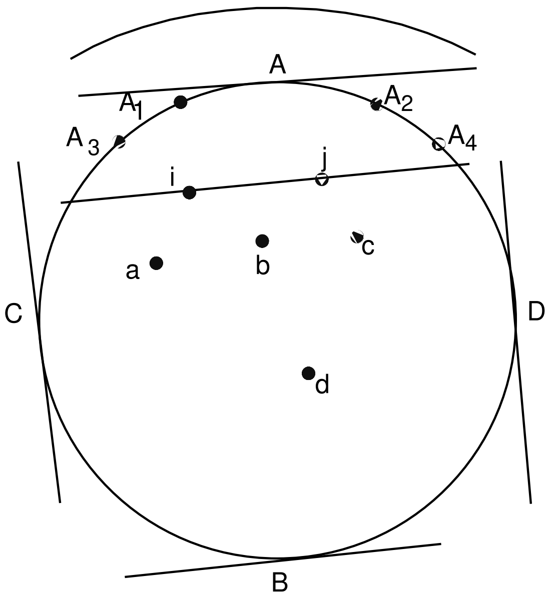

Figure 18: Valid projection range for the pair of points i,j.

reason those pairs of points (i,j) with $d(i,j) \leq min(r_i, r_j)$ need not be considered during the computation of the best projection line.

Consider a pair of points (i,j) such that $d(i,j) > min(r_i, r_j)$. Draw a line passing through the points i and j as shown in fig. 18. If the points i, j are projected on any line parallel to the line passing through i and j then the distance between the projected points will be exactly the same as the distance between the original points. Therefore this projection line is the *best* projection line for this pair of points. If the points i, j are projected on any line perpendicular to the line passing through i and j then the distance between the projected points will be zero. Therefore this projection line will be the *worst* projection line for this pair of points. Draw four tangents to the circle, two parallel and two perpendicular to the line through i, j as shown in fig. 18. Suppose the parallel tangents touch the circle at points A and B and the perpendicular tangents touch the circle at points C and D. A projection line which is tangent at point A is the best possible projection line for the points i, j and a projection line which is tangent at point C is the worst possible projection line for the points i, j. However, a projection line which is a tangent (or a line parallel to it) at point A_1 or A_2 may still be considered a *good* projection line if the distance between the projected points i' and j' is such that $d(i', j') > min(r_i, r_j)$.

Moving away from the point A on the circumference of the circle, let A_3 and A_4 be the furthest points from A on the circumference such that projected points i', j' on the tangents drawn at these points will still satisfy the condition $d(i', j') > min(r_i, r_j)$. This part of the circumference of the circle from the point A_3 to A_4 will be referred to as the *valid projection range* for the points i, j.

For every pair of points (i, j) such that $d(i, j) > min(r_i, r_j)$ we can construct a valid projection range as shown fig. 19. In fig. 19 the arc $i' - j'$ represents the valid projection range for the pair of points i and j. From these set of arcs, we find that part of the circumference of the circle, where there is the largest number of overlap of valid projection ranges. In fig. 19 the part of the circumference between the points A and B is this part. This part of the circumference will be referred to as the *best projection range*. If we draw a tangent at a point lying within the best projection range and project the n points on this line, then it will introduce the fewest occurrences of the condition that $d(i', j') \leq min(r_i, r_j)$ although $d(i, j) > min(r_i, r_j)$. In graph terms, if the points are projected on such a line, there will be fewest occurrences of the situation that the linear point graph has an edge between the nodes i and j, although there is no such edge in the corresponding planar point graph. A tangent to the circle at a point within the best projection range will be a best projection line for the given set of n points.

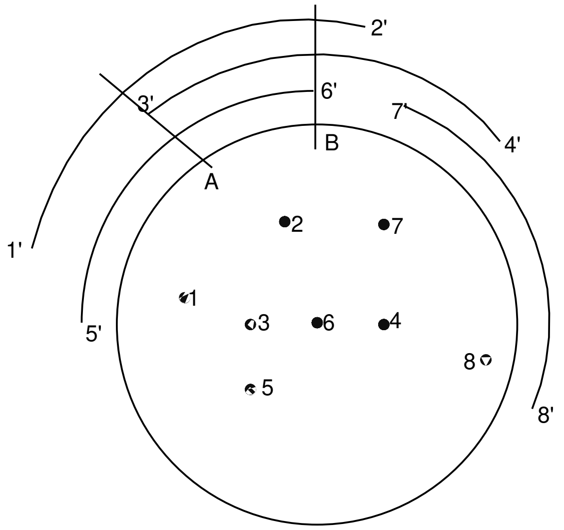

Figure 19: Valid projection ranges for pairs of points and the best projection range

6.2 Experimental Evaluation

We have done extensive experimentation to evaluate the performance of the approximate algorithm suggested in the previous section. We refer to it as the *linear projection algorithm*. The performance of this algorithm was compared with some other approximation algorithms using techniques such as (i) depth first search, (ii) breadth first search, (iii) random coloring, (iv) static maximum degree first [1] and others. The main difference between these algorithms with the linear projection algorithm is that these algorithms use topological information for coloring whereas the projection algorithm uses geometric information. However, the projection algorithm has one disadvantage. As the points are projected from the two dimensional plane to one dimension there is a certain amount of information lost. As a result two nodes which are not adjacent in the original graph may become adjacent in the graph constructed from the projected points. The number of such false adjacencies between the nodes turned out to be substantial in some of the experiments. In such cases, the projection algorithm performed poorly in comparison with other algorithms.

Table 1 gives a rough comparison of the performance of various algorithms. These results reflect the average performance of each algorithm compared to the algorithm which had the best average performance for the same sets of test data. Each entry in the table represents 150 test cases, where each test case represents uniformly distributed transceivers of varying ranges in a 200–by–200 grid. The uniform distribution leads to a relatively poor performance of the linear projection algorithm due to the large number of false adjacencies introdued by the projection.

The projection algorithm performed better than or at least as well as the other algorithms when the number of false adjacencies were low. Limitations on the size of this paper prevents us from discussing the experimental results in greater detail. For further details the reader is referred to [21].

7 Conclusions

In this paper we have shown that an arbitrary graph is not an accurate model of a packet radio network. This is true because there exists a large class of graphs which will

Table 1: Algorithm comparison showing average percent difference from the best coloring algorithm for a series of 150 trials. Note: the results reflect the average of trials at various ranges, therefore the "best" algorithm may not have a value of 0% as it's entry.

	Number of transceivers					
Algorithm	100	200	300	400	500	600
Linear Projection	183%	207%	222%	231%	236%	244%
Linear Projection Ordering	7%	12%	15%	16%	18%	19%
Linear Projection Ordering 2	4%	5%	5%	5%	6%	6%
Topological Algorithm #1	4%	8%	10%	11%	13%	14%
Topological Algorithm #2	13%	19%	23%	24%	25%	28%
Topological Algorithm #3	6%	11%	10%	12%	12%	13%
Topological Algorithm #4	1%	2%	2%	3%	2%	3%
Greedy Matrix (Random)	6%	9%	11%	12%	12%	13%
Breadth First	0%	1%	0%	1%	0%	1%
Depth First	2%	4%	5%	6%	6%	7%
Static Maximum Degree First	1%	3%	4%	5%	5%	7%
Random	6%	10%	10%	12%	12%	14%
Maximum Degree First	2%	5%	5%	6%	6%	8%
Minimum Degree Last	7%	3%	2%	2%	1%	1%
Distance–2 Min Degree Last	8%	5%	2%	1%	1%	0%

never be generated from radio networks. We have defined a new class of graphs called point graphs. A subset of point graphs called planar point graphs, captures the essential features of a packet radio network and, as such, is most appropriate for modeling the network. As the NP-completeness of a problem in a certain domain, does not necessarily imply the NP-completeness of the problem in a substantially restricted domain, the NP-completeness results for scheduling a radio network using an arbitrary graph as a model, do not correctly reflect the complexity of the problem. In this paper we have studied the broadcast scheduling problem using the planar point graphs as the model. We have shown that the problem remains NP-Complete even in this restricted domain. We have presented an $O(nlogn)$ algorithm for the case where all the transceivers are located on a line. We have also presented the idea of an approximate algorithm for the NP-complete problem and discussed the results of our experimental evaluation of the algorithm.

References

[1] S. Ramanathan and E. L. Lloyd, "Scheduling algorithms for multi–hop radio networks," in *SIGCOM*, pp. 211–222, ACM, 1992.

[2] S. Ramanathan and E. L. Lloyd, "Scheduling algorithms for multi-hop radio networks," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 166–172, April 1993.

[3] S. Even, O. Goldreich, S. Moran, and P. Tong, "On the np–completeness of certain network testing problems," *Networks*, vol. 14, pp. 1–24, 1984.

[4] E. Arikan, "Some complexity results about packet radio networks," *IEEE Transactions on Information Theory*, vol. IT-30, pp. 681–685, July 1984.

[5] B. Hajek and G. Sasaki, "Link scheduling in polynomial time," *IEEE transactions on Information Theory*, vol. 34, pp. 910–917, Sept. 1988.

[6] R. Ramaswami and K. K. Parhi, "Distributed scheduling of broadcasts in a radio network," in *INFOCOM*, pp. 497–504, IEEE, 1989.

[7] A. Ephremides and T. V. Truong, "Scheduling broadcasts in multihop radio networks," *IEEE Transactions on Communications*, vol. 38, pp. 456–461, Apr. 1990.

[8] I. Chlamtac and S. Kutten, "A spatial reuse tdma/fdma for mobile multihop radio nertworks," in *INFOCOM*, pp. 389–394, IEEE, March 1985.

[9] I. Chlamtac and A. Lerner, "A link allocation protocol for mobile multi-hop radio networks.," in *GLOBECOM*, pp. 238–242, IEEE, 1985.

[10] R. Ogier, "A decomposition method for optimal link scheduling," in *24th Allerton Conference*, pp. 822–823, October 1986.

[11] R. Bar-Yehuda, A. Israeli, and A. Itai, "Multiple communication in multi-hop radio networks," in *Eighth Annual ACM Symposium on Princ. Distrib. Comput.*, pp. 329–338, ACM, 1989.

[12] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, 1979.

[13] H. Maehara, "Space graphs and sphericity," *Discrete Applied Mathematics*, vol. 7, pp. 55–64, 1984.

[14] B. N. Clark, C. J. Colbourn, and D. S. Johnson, "Unit disk graphs," *Discrete Mathematics*, vol. 86, pp. 165–177, 1990.

[15] S. McCormick, "Optimal approximation of sparse hessians and its equivalence to a graph coloring problem," Tech. Rep. SOL 81-22, Stanford University, Department of Operations Research, 1981.

[16] D. S. Johnson, "The np-completeness column: An ongoing guide," *Journal of Algorithms*, vol. 3, pp. 182–195, 1982.

[17] M. R. Garey, D. S. Johnson, and L. J. Stockmeyer, "Some simimpified *np*–complete graph problems," *Theoretical Computer Science*, vol. 1, pp. 237–267, 1976.

[18] L. G. Valiant, "Universality considerations in vlsi circuits," *IEEE Transactions on Computers*, vol. c–30, pp. 135–140, Feb. 1981.

[19] T. Biedl and G. Kant, "A better heuristic for orthogonal graph drawing," in *2nd Annual European Symposium on Algorithms*, 1994.

[20] U. I. Gupta, D. T. Lee, and J. Y.-T. Leung, "An optimal solution for the channel–assignment problem," *IEEE Transactions on Computers*, vol. C-28, pp. 807–810, Nov. 1979.

[21] M. L. Huson, *A new model for scheduling radio networks.* PhD thesis, Arizona State University, Tempe, Arizona, August 1995.