Channel assignment on graphs of bounded treewidth

Colin McDiarmid
Department of Statistics
Oxford University

Bruce Reed
Equipe Combinatoire CNRS
Universite de Paris VI

8 September 2001

Abstract

The following 'constraint matrix span problem' arises in the assignment of radio channels in cellular communications systems. Given a graph G with a positive integer length $l(xy)$ for each edge xy, and given a positive integer B, can we assign to each vertex x a channel $\phi(x)$ from $1, .., B$ such that $|\phi(x) - \phi(y)| \geq l(xy)$ for each edge xy?

We show that this problem is NP-complete for graphs of treewidth at most 3, but there is a fully polynomial time approximation scheme for the problem on graphs of bounded treewidth. We see also that it is NP-complete for graphs which can be made bipartite by deleting a single vertex.

1 Introduction

Generalizations of graph colouring arise in the design of cellular communications systems, such as mobile telephone networks, where we need to assign radio channels to transmitters. The graph G has a vertex for each transmitter, and two vertices are joined by an edge if assigning them channels (radio frequencies) which are too close together could cause interference. A length $l(xy)$ is specified for each edge xy of G, which gives the minimum channel separation to avoid interference. We are interested not in the minimum number of channels used (which is just the chromatic number of G) but in the least width of the frequency spectrum that an assignment can occupy. Thus, we are led to the following problem.

Constraint Matrix Span Problem ($CMSP$)
Given a graph G with a positive integer length $l(xy)$ for each edge xy, and given a positive integer B, can we assign to each vertex x a channel $\phi(x)$ from $1,..,B$ such that $|\phi(x) - \phi(y)| \geq l(xy)$ for each edge xy?

The least B for which there is a feasible assignment as above is the *span* of the problem. See [6, 8, 9] for more on this and other versions of the channel (or frequency) assignment problem.

Since it is a generalization of graph colouring, $CMSP$ is NP-complete, and if $P \neq NP$ then we cannot even obtain good approximation algorithms for it, see for example [2]. In this paper, we shall observe that the problem is trivially solvable in polynomial time on bipartite graphs, but then see that it is NP-complete for graphs which can be made bipartite by deleting a single vertex. Our main results concern graphs of bounded treewidth. The problem is NP-complete for graphs of treewidth at most three, and for graphs of pathwidth at most four. In contrast, for every fixed k, there is a fully polynomial time approximation scheme for the problem on graphs of treewidth at most k.

2 Bipartite and Nearly Bipartite Graphs

2.1 Bipartite graphs and odd cycles

Given any instance (G, l) of $CMSP$, we shall always use L to denote the maximum edge length. Obviously, the span is at least $L + 1$.

Proposition 2.1 *If G is bipartite then the span is $L + 1$.*

To see this, observe that we can obtain a feasible assignment with span $L+1$ by setting $\phi(x) = 1$ for each vertex x in one part of G and $\phi(y) = L + 1$ for each vertex y in the other part. Thus, we can solve $CMSP$ on bipartite graphs in linear time. After bipartite graphs the next thing to consider is odd cycles. Here again it is easy to determine the span.

Proposition 2.2 *If G is an odd cycle then the span is* $\max(L, M)+1$*, where $M = M(G, l) = \min\{l(uv) + l(vw)|uv, vw \in E(G)\}$.*

Proof. In any feasible assignment ϕ for G, there exist edges uv and vw of G such that $\phi(u) \leq \phi(v) \leq \phi(w)$, and then $|\phi(w) - \phi(u)| \geq l(uv) + l(vw) \geq M$. Thus the span of G is at least $M + 1$.

On the other hand, choose two edges uv and vw in G with $l(uv)+l(vw) = M$. Form the even cycle G' by deleting v and adding the edge uw. Consider the length function l' on $E(G')$ which satisfies $l'(uw) = M$ and agrees with l elsewhere. Since G' is bipartite, we see that an optimal feasible assignment ϕ for G' has span $\max(M, L) + 1$. Furthermore, since u and w are at distance at least $l(uv) + l(uw)$, we can choose $\phi(v)$ between $\phi(u)$ and $\phi(w)$ to obtain a feasible assignment for G with the same span. ∎

The above proof yields a linear time algorithm for computing the span of instances of $CMSP$ for which G is an odd cycle. In fact there is a polynomial time algorithm for a more general case, namely when for a fixed k, G is k-nearly acyclic, that is there is a set of at most k vertices hitting each cycle in G – see (b2) in Subsection 3.3.

2.2 Nearly bipartite graphs

So far we have seen that it is easy to determine the span when the graph G is bipartite, or is an odd cycle or more generally is k-nearly acyclic. That is about as far as it goes! Let us consider now graphs which are 'nearly bipartite'. By a *k-nearly bipartite* graph, we mean a graph such that by deleting at most k vertices we may obtain a bipartite graph. For fixed k, we can test in polynomial time if a graph is k-nearly bipartite. Since our algorithm for solving bipartite instances of $CMSP$ is so simple, we might expect that we could solve it quickly also over k-nearly bipartite graphs for small k. Unfortunately, this turns out not to be the case, even for $k = 1$.

Note that it is very easy to determine the chromatic number of a 1-nearly bipartite graph, as it is at most 3. However, we shall see below that it is NP-hard to determine the span, even if we restrict the edge lengths to be 1 or 2. Indeed it remains NP-hard for the subcase when $G - v$ is bipartite for some vertex v such that all edges not incident with v have length 1. For this subcase, the span must be at most 4 (set $\phi(v) = 1$ and do not use channel 2), and it is NP-complete to tell if it is at most 3. Thus, we cannot hope to obtain a polynomial time approximation algorithm with performance ratio better than $\frac{4}{3}$ for $CMSP$ even over such simple instances.

Let us call the following problem 1-NEARLY BIPARTITE SPAN. Given a graph G and vertex v such that $G - v$ is bipartite, and given lengths 1 or 2 on the edges such that each edge not incident with v has length 1, is the span at most 3?

Theorem 2.3 *1-NEARLY BIPARTITE SPAN is NP-complete.*

Proof. We prove the theorem by giving a polynomial time reduction to it from the problem 2-NEARLY BIPARTITE 3-COLOURABILITY. The latter problem is to tell whether a given 2-nearly bipartite graph G has a 3-colouring. It is NP-complete, as may be shown by a reduction from general 3-colourability – see [7].

Suppose then that $G = (V, E)$ is a 2-nearly bipartite graph. Let s and t be distinct vertices such that $G - s - t$ is bipartite. We may assume that s and t are adjacent, for if not then G is 3-colourable. Let $N(t)$ be the set of neighbours of t other than s.

From G we form a network consisting of a graph $G' = (V', E')$ together with edge lengths 1 or 2, as follows. Delete vertex t. For each neighbour $w \in N(t)$, let w' be a new vertex adjacent to w and to s. Give each edge sw' length 2, and give each other edge length 1. Observe that $G' - s$ is bipartite. Clearly this construction takes polynomial time. We must show that G is 3-colourable if and only if the network on G' has span at most 3.

Suppose first that G has a 3-colouring $c : V \to \{1, 2, 3\}$. We may assume that $c(s) = 1$ and $c(t) = 3$. Then we obtain a feasible assignment for the network by setting $\phi(w') = 3$ for each $w \in N(t)$, and $\phi(x) = c(x)$ for each other vertex $x \in V'$.

Now suppose that there is a feasible assignment $\phi : V' \to \{1, 2, 3\}$. We may assume that $\phi(s) = 1$ and $\phi(w') = 3$ for each $w \in N(t)$. Then we obtain a (proper) colouring c of G by setting $c(t) = 3$ and $c(x) = \phi(x)$ for each other vertex $x \in V$. ∎

To close this subsection, we shall use the result mentioned at the end of the last subsection above about k-nearly acyclic graphs, to show that for any fixed k we can at least approximate the span when the graph is k-nearly bipartite. Indeed, given such an instance (G, l) of $CMSP$, we can obtain a feasible assignment within a factor 2 of optimal in polynomial time.

We first find a set X of at most k vertices such that $G - X$ has a 2-colouring, with colour sets U and W. We then compute an optimal assignment ϕ for the restriction of the $CMSP$ problem to the k-nearly acyclic subgraph induced by $X \cup W$, with corresponding span S. Finally, we set

$\phi'(v) = 1$ for each vertex $v \in U$, and $\phi'(v) = \phi(v) + L$ for each vertex $v \in X \cup W$. This polynomial time algorithm clearly yields a feasible assignment ϕ' with span $S + L$. But $S + L \leq 2\max\{S, L\}$, which is at most twice the span of the instance, and so indeed we have found a solution within a factor 2 of optimal.

3 CMSP on graphs of bounded treewidth

We can solve $CMSP$ over trees in linear time, because they are bipartite. Many problems for which there are linear time algorithms over trees can also be solved in linear time on graphs of bounded treewidth. Such graphs resemble trees in that they can be decomposed into small pieces using small cutsets which fit together in a tree-like way. We see below that $CMSP$ remains hard for graphs of bounded treewidth, though in contrast there are efficient approximation algorithms.

3.1 Graphs of bounded treewidth

A *tree decomposition* of a graph G consists of a tree T and a subset W_t of $V(G)$ for each node t of G such that:

(i) for each edge $xy \in E(G)$, there exists t with $\{x, y\} \subseteq W_t$, and

(ii) for each vertex x, $S_x = \{t \mid x \in W_t\}$ induces a subtree of T.

The removal of an arc st of T separates the component T^s of $T - st$ containing s from the component T^t of $T - st$ containing t. Condition (ii) implies that for any arc st of T, $V(G)$ can be partitioned into the set $V_s = \{x \mid S_x \subseteq T^s\}, V_t = \{x \mid S_x \subseteq T^t\}$, and $W_s \cap W_t$. Just as removing st separates T^s from T^t, the removal of $W_s \cap W_t$ separates V_s from V_t (to see this, simply apply condition (i)). Similarly, W_s separates $V - W_s$ into pieces corresponding to the components of $T - s$.

If these separating sets (the W_s) are small then many problems can be solved quickly using dynamic programming. So we define the *width* of a tree decomposition to be the maximum of $\{|W_t| - 1 \mid t \in V(T)\}$ and define the *treewidth* of G to be the minimum of the widths of its tree decompositions (the -1 here is so that trees have treewidth 1). *Pathwidth* is defined similarly, except that T must be a path.

As mentioned above, many NP-complete problems can be solved in polynomial time on graphs of bounded treewidth. Examples include Edge Colouring, Vertex Colouring, Maximum Independent Set, and Hamilton Cycle. Indeed often these problems can be solved in linear time over this class. We refer the reader to the surveys [1] and [10] for more details on the algorithmic aspects of treewidth, and for further references.

The following is a simple consequence of the definition of treewidth and the fact that the Helly property holds for trees (that is, that for any family of subtrees of a tree every pair of which have a common node, there must be a node which belongs to each tree).

Lemma 1 *If G has a clique cutset C and $G - C$ has components $U_1, ..., U_k$ then the treewidth of G is the maximum of the treewidths of the subgraphs of G induced by $C \cup U_1, ..., C \cup U_k$.*

3.2 Exact Solution is Hard

We now show that given an instance (G, l) of $CMSP$, and a positive integer B, determining if the span of (G, l) is at most B is NP-complete, even if we insist that G has treewidth at most 3 (or pathwidth at most 4). We give a reduction from the decision version of an 'Equality' variant $ECMSP$ of $CMSP$, where we require that $|\phi(x) - \phi(y)|$ be exactly equal to $l(xy)$. This problem is NP-complete even if G is a path, as is shown in [11]. In that paper ECMSP for paths is disguised as 'ruler folding', where the edges of the path are the links of a flexible ruler which may be folded around its joints, the vertices. The question is: can the ruler be folded within the length L of its longest link? This is a special case of $ECMSP$ where we want the span to be $L + 1$.

For completeness, let us show quickly that the ruler folding problem is NP-complete, by giving a reduction from the familiar NP-complete problem PARTITION, see for example [5]. Let $a_1, a_2, \ldots, a_n$ be a list of n positive integers, and let S be their sum. Form the ruler with $n + 4$ links of lengths $2S, S, a_1, a_2, \ldots, a_n, S, 2S$. Let u denote the second joint (after the links of length $2S$ and S) and let v denote the second last joint (before the links of length S and $2S$). If there is a partition of $\{1, \ldots, n\}$ into $A \cup B$ such that $\sum_{i \in A} a_i = \sum_{j \in B} a_j$, then the ruler may be folded within the length $2S$ of its longest link – put the links in A 'up', put the links in B 'down', and at each end put one link up and one down. This works since the joints u and v are at the same height. Conversely, if there is no such partition, then however we fold the ruler, the joints u and v are not at the same height, and so the folded ruler occupies length at least $2S + 1$.

Now let us describe our reduction for $CMSP$. The first step of the reduction from the decision version of $ECMSP$ for paths to the decision version of $CMSP$ is to take exactly the same path P and length function l on the edges of the path. Because we then add on extra 'gadgets', the bound B on the span of the $CMSP$ instance will be much bigger than the bound $L + 1$ on the span of the $ECMSP$ instance. By doubling all edge lengths if necessary, we may assume that L is even, say $L = 2K$ for some positive integer K. We choose the bound B to be $2C + 1$ where $C = 4K^2$. The graph G will be constructed by glueing a separate gadget on to each edge of the path P.

Let us describe the gadgets. We are given a positive integer K, and let $C = 4K^2$. (The reason for this choice of C will appear later.) The pairs (r, k) of integers such that $0 \leq r < 2K$, k is of the form $2^i K$ for some non-negative integer i, $k \leq C$ and $2rk \leq C$ will be called the *K-relevant pairs*. We describe a gadget $G_{r,k}$ for each K-relevant pair (r, k). These gadgets are graphs with edge lengths, and each also has two (symmetrical) 'ports' x and y, which are nodes which will be identified with nodes in other graphs to 'glue' the gadget on. They are constructed so that the values taken on the ports x and y in any feasible assignment with span at most B must differ by exactly the edge length $l(xy)$, and must both lie in a certain interval, where this interval has length $2k$ and grows to be all of $\{1, \ldots, B\}$ when $k = C$ (and thus $r = 0$).

First consider the case when $r = 0$. Let $(0, k)$ be a K-relevant pair. To form the gadget $G_{0,k}$ we start with the complete graph on the four vertices x, y, u, v. The ports are x and y. We set $l(xy) = 2k$, $l(uv) = 2C$, and $l(e) = C - k$ for the other four edges e.

Now let (r, k) be a K-relevant pair with $r > 0$, and suppose that we have described how to construct all the gadgets $G_{r',k'}$ for all the K-relevant pairs (r', k') with $r' < r$. In particular, both $(0, k)$ and $(\lfloor r/2 \rfloor, 2k)$ are K-relevant pairs, and we can construct the corresponding gadgets $G_{0,k}$ and $G_{\lfloor r/2 \rfloor, 2k}$. To construct the gadget $G_{r,k}$ we start with the complete graph on the four vertices x, y, u, v. The ports are x and y as before. We set $l(ux) = l(uy) = k$ and $l(vx) = l(vy) = k + \lceil r/2 \rceil$. Now we glue on $G_{0,k}$ to x and y but set $l(xy) = 2k - r$ (not $2k$); and glue on $G_{\lfloor r/2 \rfloor, 2k}$ to u and v, and set $l(uv) = 4k - \lfloor r/2 \rfloor$. Note that $|V(G_{0,k})| = 4$; and for each $r \geq 1$, $|V(G_{r,k})| = 4 + |V(G_{\lfloor r/2 \rfloor, 2k})|$, and so $|V(G_{r,k})| \leq 4 \log_2 r + 8$.

This completes the description of the gadgets. We need two lemmas.

Lemma 2 *Consider a K-relevant pair (r, k) and the corresponding gadget $G_{r,k}$. For each pair of integers a and b, there is a feasible assignment ϕ for $G_{r,k}$ with span at most $B = 2C + 1$ and with $\phi(x) = a$, $\phi(y) = b$ if and only if*

$$C + 1 - k \leq a, b \leq C + 1 + k \quad \textit{and} \quad |a - b| = 2k - r. \tag{1}$$

Proof. Consider first the case $r = 0$. Let $(0, k)$ be a K-relevant pair. This case is easy since we can identify all the feasible assignments for $G_{0,k}$ with span at most B. There are exactly four of them. We have (i) either $\phi(u) = 1$, $\phi(v) = B$ or $\phi(u) = B$, $\phi(v) = 1$ and (ii) either $\phi(x) = C + 1 - k$, $\phi(y) = C + 1 + k$ or $\phi(x) = C + 1 + k$, $\phi(y) = C + 1 - k$.

Now let (r, k) be a K-relevant pair with $r > 0$, and assume for induction that the result of the lemma holds for any K-relevant pair (r', k') with $r' < r$. Suppose first that there is a feasible assignment ϕ for $G_{r,k}$ with span at most B, and with $\phi(x) = a$, $\phi(y) = b$. We must show that (1) holds. Note first that the glued-on gadget $G_{0,k}$ ensures that $C+1-k \leq a, b \leq C+1+k$ holds. Also, by the induction hypothesis applied to $G_{\lfloor r/2 \rfloor, 2k}$, we have $C+1-2k \leq \phi(u), \phi(v) \leq C+1+2k$ and $|\phi(u)-\phi(v)| = l(uv) = 4k - \lfloor r/2 \rfloor$. Since this last quantity is at least $3k$, it follows that both $\phi(x)$ and $\phi(y)$ must lie between $\phi(u)$ and $\phi(v)$. But now

$$|\phi(x) - \phi(y)| \leq (4k - \lfloor r/2 \rfloor) - k - (k + \lceil r/2 \rceil) = 2k - r,$$

and so (1) holds.

Conversely, suppose that a, b satisfy (1). We must define a feasible assignment ϕ for $G_{r,k}$ with span at most B and with $\phi(x) = a$, $\phi(y) = b$. We start by setting ϕ to take the values 1 and B on the two non-port nodes of the glued-on $G_{0,k}$. By symmetry we may assume that $a < b$, so $b = a + l(xy)$. Since $l(xy) = 2k - r$, either $a \geq C + 1 - k + \lceil r/2 \rceil$ or $b \leq C + 1 + k - \lceil r/2 \rceil$. In the former case, set $\phi(u) = a - k - \lceil r/2 \rceil$ and $\phi(v) = b + k$. In the latter case, set $\phi(u) = a - k$ and $\phi(v) = b + k + \lceil r/2 \rceil$. In each case,

$$C + 1 - 2k \leq \phi(u), \phi(v) \leq C + 1 + 2k$$

and

$$|\phi(u) - \phi(v)| = |a - b| + 2k + \lceil r/2 \rceil = 4k - \lfloor r/2 \rfloor.$$

Now the values assigned so far for ϕ satisfy the relevant constraints, and by the induction hypothesis they may be extended to a feasible assignment for the glued-on $G_{\lfloor r/2 \rfloor, 2k}$ with span at most B. We thus obtain a feasible assignment ϕ for all of $G_{r,k}$ with span at most B. ∎

To each edge uv of the path P, we glue on a gadget $G_{r,K}$ where $r = 2K - l(uv)$. Note that each such pair (r, K) is a K-relevant pair, since $2rK \leq 2(2K)K = C$, which explains our choice of C. By the bound above on the sizes of the gadgets, we see that this construction can be done in polynomial time. Furthermore, the graph we have constructed is obtained from cliques of size four and paths by glueing along edges. So, since paths have treewidth 1 and a clique of size four has treewidth three, Lemma 1 implies that the final graph has treewidth at most three. A slightly longer analysis allows us to verify that it has pathwidth at most four. The next lemma will complete the proof.

Lemma 3 *The original ECMSP instance has span at most $L+1$ if and only if the constructed CMSP instance has span at most B.*

Proof. Suppose that the original path instance of $ECMSP$ has span at most $L + 1 = 2K + 1$. Then it has a feasible assignment ϕ such that

$$C + 1 - K \leq \phi(u), \phi(v) \leq C + 1 + K \quad \text{and} \quad |\phi(u) - \phi(v)| = l(uv)$$

for each edge uv on the path. But by Lemma 2, this assignment extends to a feasible assignment for the constructed $CMSP$ with span at most B.

Conversely, suppose that the $CMSP$ has a feasible assignment with span at most B. Then by Lemma 2, the restriction of this assignment to the nodes on the original path yields a feasible assignment for the original instance of $ECMSP$ with span $L + 1$. ∎

3.3 Approximating is Easy

Now let k be a fixed positive integer, and consider the problem $CMSP_k$, which is the restriction of $CMSP$ to graphs of treewidth at most k. We shall see that we can obtain approximation algorithms for this problem using dynamic programming techniques.

Given a graph G of treewidth at most k, there is a linear time algorithm to compute a tree decomposition of width k for G [4], so we can assume we have such a decomposition when constructing our algorithms. To apply dynamic programming given a tree decomposition $[T, \{W_t | \; t \in V(T)\}]$ of G, we root T at some node r, and for each node s let T_s be the maximal subtree of T rooted at s. We define G_s to be the subgraph induced by $\cup\{W_t | \; t \in T_s\}$. As discussed above, W_s is a cutset separating $G - G_s$ from $G_s - W_s$.

So, we consider an instance of $CMSP_k$ on G with associated length function l and span bound B. Our approach will be to construct, for each node s, a table containing all the feasible assignments for W_s which extend to feasible assignments of span at most B for G_s. The instance has span at most B if and only if the table for r is non-empty, since $G_r = G$.

We construct these tables starting at the leaves and working towards the root, only constructing the table for a node after we have constructed the table for each of its children (that is, we perform a post-order traversal of T).

It is easy to compute the table for a leaf s because G_s is the graph induced by the at most $k+1$ vertices of W_s. To compute the table for a non-leaf node s we use the fact that a feasible assignment ϕ for W_s extends to a feasible assignment for G_s of span at most B if and only if for every child t of s, the restriction of ϕ to $W_s \cap W_t$ extends to a feasible assignment for G_t of span at most B. This fact is easy to prove using the fact that the $W_s \cap W_t$ separate G. Also, to determine if ϕ extends as desired to G_t we just need to look into our table for t and see if any of the assignments there agree with ϕ on $W_s \cap W_t$.

This simple to describe process runs (assuming we can systematically generate all possible table entries efficiently) in time $O(n\tau)$ where τ is an upper bound on the maximum possible size of a table, and $n = |V(G)|$. Now, if $|W_s| = k+1$ then there are at most B^{k+1} possible feasible assignments for W_s, and so we may take τ as B^{k+1}. Since we needed only about $\log B$ bits to input B, we see that this algorithm is pseudo-polynomial rather than polynomial.

However, suppose that in polynomial time we can find for each vertex v a polynomially bounded set S_v of integers such that, if there is a feasible assignment with span at most B, then there is such an assignment ϕ with $\phi(v) \in S_v$ for each v. Then the above method yields a polynomial time algorithm. We can find such sets S_v for example

(a) when B is polynomially bounded; or more generally when all edge lengths are multiples of a positive integer t and B/t is polynomially bounded.

(b) when the number of distinct path lengths is polynomially bounded.

To see why these examples work, note the following result from [3] (see also [9]), related to the Gallai-Roy theorem on graph colouring (see for example [12]).

Lemma 4 *There is an acyclic orientation of G such that, if $\phi(v)$ is 1 plus the maximum length of a path ending at v (thus $\phi(v) = 1$ if v is a source, with no incoming arcs), then ϕ is an optimal assignment.*

There are at least two natural cases where (b) above holds, one depending on the edge lengths and one on the graph, namely

(b1) when for some fixed t, for each n the number of distinct edge lengths is at most t, or

(b2) when for some fixed t, the graph G is t-nearly acyclic, that is there is a set X of at most t vertices hitting each cycle of G.

(With (b1) the number of distinct path lengths is at most $(n+1)^t$; for if the edge lengths are restricted to $a_1, .., a_t$ then each path length is $b_1 a_1 + b_2 a_2 + .. + b_t a_t$ for some non-negative integers b_i which sum to at most n. With (b2) the total number of paths is at most $(t+1)^t n^{2t+2}$, since there are at most $n^2 - 1$ paths avoiding X.)

We used (b2) above in the last section. Now let us see how to use (a) to find approximate solutions for $CMSP_k$, and indeed we obtain a Fully Polynomial Time Approximation Scheme for the problem. Suppose then that we are given an instance (G, l) of $CMSP_k$, and $\varepsilon > 0$. We must find a feasible assignment with span at most $(1+\varepsilon)S$, where S denotes the span of (G, l), in polynomial time (that is, in time bounded by a polynomial in n, $\log L$ and $1/\varepsilon$).

If $\varepsilon L < 2n$ then $S \leq nL \leq \frac{2}{\epsilon} n^2$, and so by (a) we can determine the span exactly in polynomial time. Suppose then that $\varepsilon L \geq 2n$, and let $t = \lfloor \frac{\varepsilon L}{n} \rfloor$. Round up each edge length to the nearest multiple of t. By Lemma 4, the span increases by at most $nt \leq \varepsilon L$. Further the span is at most $nL \leq (\frac{2}{\epsilon} n^2)t$, and so by (a) again we can find an optimal assignment for the new problem, which is then a feasible assignment for the original problem as required.

4 Final Remarks

We draw one main conclusion from our analysis: small cutsets are unhelpful in decomposing the problem into subproblems if we want an exact solution. We obtained our NP-complete instance by pasting together trivial instances using clique cutsets of size two. We do not even know if the problem can be solved exactly if every block of G is an odd cycle. On the other hand, if we are willing to settle for an approximate solution then we can use these cutsets to 'divide and conquer'.

References

[1] S. Arnborg, J. Lagergren and D. Seese, Easy problems for tree decomposable graphs, *J. Algorithms* **12** (1991) 308-340.

[2] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela and M. Protasi, *Complexity and Approximation*, Springer, 1999.

[3] F. Barasi and J. van den Heuvel, Graph labelling, orientations, and greedy algorithms, 2001, in preparation.

[4] H. Bodlaender, A linear time algorithm for finding tree decompositions of small treewidth, *Proc. 25th Annual ACM Symposium on the Theory of Computing* (1993) 226-234.

[5] M.R. Garey and D.S. Johnson, *Computers and Intractability*, Freeman, 1979.

[6] W.K. Hale, Frequency assignment: Theory and applications, *Proceedings of the IEEE* **68** (1980) 1497-1514.

[7] J. Kratochvil and A. Sebő, Coloring precolored perfect graphs, *J. Graph Theory* **25** (1997) 207 – 215.

[8] S. Hurley and R. Leese (editors), *Models and Methods for Radio Channel Assignment*, Oxford University Press, to appear.

[9] C. McDiarmid, Discrete mathematics and channel assignment, to appear as a chapter in *Recent advances in theoretical and applied discrete*

mathematics, C. Linhares-Salas and B. Reed, editors, Springer, to appear.

[10] B. Reed, Tree width and tangles, a new measure of connectivity and some consequences, in *Surveys in Combinatorics*, R. Bailey, editor, Cambridge University Press, 1997.

[11] S. Whitesides, Algorithmic issues in the geometry of planar linkage movement, *The Australian Computer Journal*, Special Issue on Algorithms, May 1992, 42-50.

[12] D.B. West, *Introduction to Graph Theory*, Prentice Hall, 1996.