Area-Delay Constrained Trees in Graphs

Philip Chong

November 20, 2001

1 Area-Delay Constrained Layout

We pose the problem of finding a suitable layout for a module with area and delay constraints. For simplicity, we assume there is a single input pin to the module and a single output pin from the module, and that the locations of these pins have already been determined. The logic implementing the module consumes a specified amount of area; in the general case, there is both an upper and a lower bound on this area. The delay from the input pin to the output pin is bounded from above. If distance is equivalent to delay (i.e. the delay is determined by the distance of the path through the module from input to output), then the layout of the module affects this delay.

More concretely, consider a layout of a module onto a grid of unit squares. The number of squares the module consumes is constrained, and a path from the input pin to output pin is constrained in length, and must run through squares occupied by the module. For simplicity (although this may be a reasonable assumption anyways), the squares occupied by the module must be contiguous, and the path from input pin to output pin contiguous as well.

This problem can be abstracted into a graph problem which we call here the Area-Delay Constrained Tree (ADCT) Problem. The squares are considered vertices in a graph, and contiguous squares are joined by an edge in the graph. Since the squares occupied by the module are contiguous, there exists a tree in the underlying graph which contains all occupied squares. The following section describes a generalized version of the layout problem in this graph context, and shows that it is NP-complete in the general case.

2 The Area-Delay Constrained Tree Problem

We formally state the ADCT Problem:

Given: Graph $G = (V, E)$, distinguished vertices $s, t \in V$, weights on edges $w(e), e \in E$, integers $B, K_1, K_2, B \leq K_1 \leq K_2$.

Question: Does there exist a tree $H \subseteq E$ (not necessarily spanning) in G such that H contains a path P from s to t, $\sum_{e \in P} w(e) \leq B$, and $K_1 \leq \sum_{e \in H} w(e) \leq K_2$?

ADCT is in NP. A certificate H can be verified in polynomial time using variants of breadth-first search. In particular, the properties that H is a tree of total weight between K_1 and K_2, and H contains a path from s to t of total weight at most B can be verified.

A reduction from Exact 3-Set Cover (X3C) to ADCT shows that ADCT is NP-complete. The X3C Problem stated formally:

Given: Set $R = \{r_i\}$, $|R| = 3n$, and set $C = \{C_i\}, |C| = m, m > n$, where each $C_i = \{r_{c_{i1}}, r_{c_{i2}}, r_{c_{i3}}\}, r_{c_{ij}} \in R$, and $c_{ij} \neq c_{ik}$ if $j \neq k$.

Question: Is there a subset $D \subseteq C$ such that every element in R appears in exactly one element of D?

We reduce X3C to ADCT as follows: For any instance of X3C, construct an instance of ADCT. Let

$$\begin{aligned} V &= \{s, t\} \\ &\quad \cup \{c'_i, c''_i : 1 \leq i \leq m\} \\ &\quad \cup \{r'_i, r''_i : 1 \leq i \leq 3n\} \end{aligned}$$

so that there are two vertices c'_i and c''_i for each element of C, and two vertices r'_i, r''_i for each element of R, in addition to the distinguished vertices s, t. Let

$$\begin{aligned} E &= \{(s, c'_1), (c'_1, c'_2), \ldots, (c'_m, t)\} \\ &\quad \cup \{(c'_i, c''_i) : 1 \leq i \leq m\} \\ &\quad \cup \{(c''_i, r'_j), (c''_i, r'_k), (c''_i, r'_l) : 1 \leq i \leq m, C_i = \{r_j, r_k, r_l\}\} \\ &\quad \cup \{(r'_i, r''_i) : 1 \leq i \leq 3n\} \end{aligned}$$

so that there are edges for the path $P' = (s, c'_1, c'_2, \ldots, c'_m, t)$, edges between each c'_i and its corresponding c''_i, edges between each c''_i and all the vertices corresponding to elements of C_i, and finally edges between each r'_i and its corresponding r''_i. Let the weights of the edges be

$$w(e) = \begin{cases} 1 & \text{if } e \in P' \\ 6mn & \text{if } e = (c'_i, c''_i) \\ (6mn)^2 & \text{if } e = (c''_i, r'_j) \\ (6mn)^3 & \text{if } e = (r'_i, r''_i) \end{cases}$$

For an example of this construction, see Figure 1.

Choose $B = m+1, K_1 = K_2 = (m+1) + 6mn(n) + (6mn)^2(3n) + (6mn)^3(3n)$.

Note that the constructed instance of ADCT has size polynomial in the size of the X3C problem, and can be constructed in polynomial time: since the edges of the ADCT problem are constructed by examining each set C_i individually, this transformation only requires a single pass through the C_i. Hence this is a polynomial-time reduction from X3C to ADCT.

Now we show the equivalence of solutions to X3C and the reduced ADCT problem. If a solution set $D = \{C_{d_i} : 1 \leq i \leq n\}$ exists for the original X3C problem, we can choose H to be

$$H = P'$$

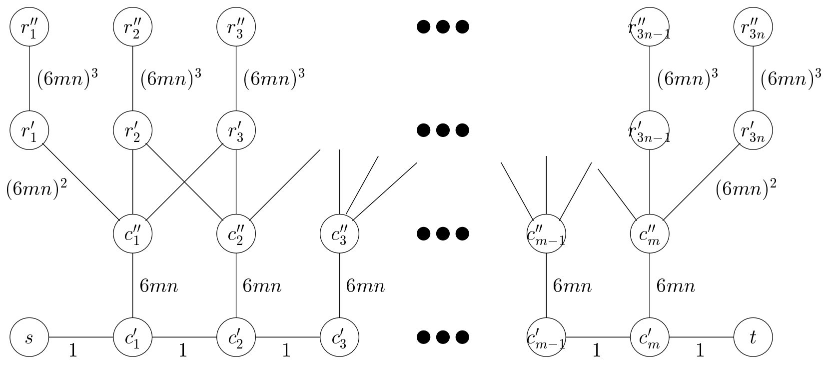

Figure 1: ADCT Instance Construction From X3C

$$\begin{aligned}
&\cup\{(c'_{d_i}, c''_{d_i}) : C_{d_i} \in D\} \\
&\cup\{(c''_{d_i}, r'_j) : C_{d_i} \in D, r_j \in C_{d_i}\} \\
&\cup\{(r'_i, r''_i) : 1 \leq i \leq 3n\}
\end{aligned}$$

so that H is the path P', together with the edges (c'_x, c''_x) corresponding to the elements of D, together with all the edges (c''_x, r'_y) incident to the chosen vertices c''_x, together with all edges (r'_y, r''_y).

We can show H is a tree. Suppose a cycle exists in H. Since all r''_y have only one incident edge, these vertices cannot appear in any cycle. Since no element r_y appears twice in the covering set D, all the corresponding vertices r'_y will have only two incident edges in H, one edge which is incident to r''_y, so the vertices r'_y cannot appear in any cycle in H. Since every vertex c''_i is adjacent only to vertices r'_{y1}, r'_{y2}, r'_{y3}, and vertex c'_i, no c''_i can be contained in a cycle in H. Thus any cycle in H must contain only the vertices s, t and c'_i, but it is easy to see that no possible cycle exists in G. Thus H is cycle-free, hence a tree.

It is easy to verify that H contains path P' from s to t of weight B, and $K_1 \leq \sum_{e \in H} w(e) \leq K_2$ by the construction of H. Hence H is a solution to the reduced ADCT problem.

To complete the equivalence, consider a solution H to the reduced ADCT problem. Let $Z = \sum_{e \in H} w(e)$. Note that there are four different edge weights $\{1, 6mn, (6mn)^2, (6mn)^3\}$ in the graph, and that the number of edges of any given weight is less than $6mn$. In particular, this means that the value of Z completely determines the number of edges of each weight present in H. But $Z = K_1 = K_2 = (m+1) + 6mn(n) + (6mn)^2(3n) + (6mn)^3(3n)$, since H is a solution to the reduced ADCT problem. Thus, all $3n$ edges of the form (r'_y, r''_y) must be in H. Since there are also $3n$ edges of the form (c''_x, r'_y), and H describes a tree (connected graph), by Dirichlet's Principle there must be exactly one edge (c''_x, r'_y) in H for each r'_y.

Now note there are exactly n edges of the form (c'_x, c''_x) in H. Since H is connected, and each c''_x is adjacent to exactly three vertices $\{r'_j, r'_k, r'_l\}$, we conclude that if $c''_x \in H$, then all r'_y adjacent to c''_x must be in H as well. Take $D = \{C_i : c''_i \in H\}$. From the above analysis, D induces an exact cover on C, and hence is a solution to the original X3C Problem.

Since X3C is NP-complete, ADCT must also be NP-complete.

3 ADCT On Unweighted Graphs

A polynomial time algorithm exists for ADCT on unweighted (i.e. $w(e) = 1$ for all $e \in E$) graphs. Any path from s to t can be augmented in a greedy fashion by finding any vertex v_i in H adjacent to a vertex v_j not in H, and adding (v_i, v_j) to H. This process can be terminated when the number of edges in H is at least K_1, and if the starting path from s to t is at most length B, then the result is a tree which satisfies the unweighted ADCT problem.