Protecting Telecommunications Networks: Toward a Minimum-Cost Solution

M. J. O'Sullivan, C. G. Walker, M. L. O'Sullivan, T. D. Thompson
Department of Engineering Science
University of Auckland, New Zealand

Abstract

The problem of designing fibre-optic networks for telecommunications can be decomposed into (at least) three non-trivial subproblems. In the first of these subproblems one must determine how many fibre-optic cables (fibres) are required at either end of a street. In the next subproblem a minimum-cost network must be designed to support the fibres. The network must also provide distinct paths from either end of the street to the central exchange(s). Finally, the fibre-optic cables must be placed in protective covers. These covers are available in a number of different sizes, allowing some flexibility when covering each section of the network. However, fibres placed within a single cover must always be covered together for maintenance reasons.

In this paper we describe two formulations for finding a minimum-cost (protective) covering for the network (the third of these subproblems). This problem is a generalised set covering problem with side constraints and is further complicated by the introduction of fixed and variable welding costs. The first formulation uses dynamic programming (DP) to select covers along each arc (in the network). However, this formulation cannot accurately model the fixed costs and does not guarantee optimality. The second formulation, based on the DP formulation, uses integer programming (IP) to solve the problem and guarantees optimality, but is only tractable for smaller problems.

The cost of the networks constructed by the IP model is less than those designed using the DP model, but the saving is not significant for the problems examined (less than 0.1%). This indicates that the DP model will generally give very good solutions despite its limitation. Furthermore, as the problem dimensions grow, DP gives significantly better solution times than IP.

1 Introduction

Since 1995, the Operations Research (OR) group at the University of Auckland has been developing telecommunications network design technology [11, 9, 10]. The networks considered link the Central Business District (CBD) of a major centre (e.g., Auckland, Wellington, etc) to one (or more) exchange buildings for the area. The exchange buildings for the major centres are then connected together to form a network for New Zealand.

The CBD subnetworks consist of fibre-optic cables (*fibres*) that start in buildings and finish at exchanges. Fibres connect from each building to two *vaults*, one at each end of the street (where the building is located). These vaults are then connected to the exchange via a network of *feeders* and other vaults. To ensure reliability there is a distinct path from each vault at the end of a street to the exchange(s), i.e., two independent ways for every building to connect to the exchange(s). Feeders are underground trenches that contain the fibres, and vaults are access areas where workers may perform maintenance and alterations to the network. Vaults either bring new fibres into the network (from one or more streets) or join the fibres along two (or more) feeders together (many vaults do both).

Fibres are placed into *tubes* either upon exiting a building or entering a vault from the street (or sometimes at both points). The tubes are then placed in protective sheathes referred to as *covers*. Whenever feeders meet at a vault the fibres from the incoming feeders may be combined into larger (cheaper) covers.

Philpott and Mason [9] developed the Fibre Diversity Optimiser (FiDO) for designing low-cost, reliable fibre-optic networks to meet customer bandwidth demand. FiDO decomposes the network design process into three stages, determining in turn:

1. the number of fibres connected to the vaults at either end of a street (*Street Optimisation*);
2. two distinct paths from the vaults at each end of a street to the exchange(s) (*Path Optimisation*);
3. how to pack fibres into the covers as they travel through the feeders and vaults to the exchange (*Cover Optimisation*).

The Street Optimisation problem involves deciding the location, size and number of multiplexers for each street, from which copper cable is ducted to the telephone receivers. These decisions represent a trade off between the use of copper or the use of multiplexers and fibre. The optimium mix of copper and fibre is determined by FiDO as the solution to a set-partitioning problem, which is known to be NP-hard [5].

The Path Optimisation problem is an example of the well-studied multi-commodity network design problem [8, 3, 6, 7, 2, 4, 1], which is NP-hard, even in the single commodity case [5].

In this paper we are concerned with solving the Cover Optimisation problem, which FiDO does using a bin-packing heuristic. We model this problem as a generalised set covering problem (known to be NP hard [5]) with side constraints, which we initially solve using DP. The problem is complicated by fixed welding costs, removing the guarantee of optimality from DP solutions. We also present an IP formulation that does guarantee optimality, but which is intractable for larger problems.

The rest of this paper is structured as follows. Section 2 describes the physical characteristics of the network we need to consider in the Cover Optimisation problem. In §3 we outline our DP formulation. We present the IP formulation for the cover optimisation in §4. Section 5 gives results comparing the two formulations (and the FiDO heuristic). Finally, in §6 we discuss the merits of the two approaches.

2 The Cover Optimisation Problem

In this section we describe the physical considerations inherent in the Cover Optimisation problem. In particular we explain how covers are used to protect the fibres in the network, including side-welding new fibres into covers and welding covers together at a vault.

Note that throughout the remainder of this paper fibres run from the building to the exchange. Fibres and/or covers *entering* a vault have originated at a building and *leaving* a vault are going to the exchange.

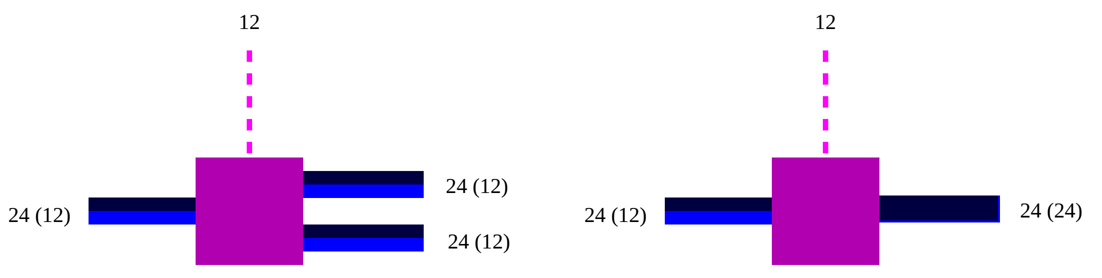
Cover Types There may be several cover types available, each with a given fibre capacity and cost per unit length. The capacity increases at a faster rate than the cost. We considered the cover types in table 1 during this research. Note that it is

Fibre Capacity	Cost per Unit Length ($)
24	7.84
48	11.68
96	17.44
144	21.28

Table 1: Available cover types

cheaper to use a single cover with capacity 48 than two covers with capacity 24 over the same distance.

New Fibres When new fibres enter at a vault they can either be placed directly into a new cover or *side-welded* into an existing cover entering the vault. Figure 1 shows the two possibilities. In figure 1 (a), the 12 new fibres are placed in a new

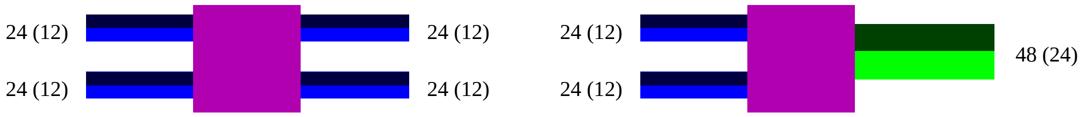


(a) New fibres in a new cover (b) New fibres side-welded into an existing cover

Figure 1: Dealing with new fibres

cover (capacity 24) that then leaved the vault. In figure 1 (b), the new fibres are side-welded into the incoming cover (of capacity 24, carrying 12 fibres) that then leaved the vault.

Existing Covers When feeders containing covers enter a vault the incoming covers can either leave the vault unchanged or their fibres are welded together into a larger cover (that then leaves the vault). The larger cover must have capacity at least as great as the total capacity of the incoming covers (due to the way welding is performed). Figure 2 shows the two possibilities. In figure 2 (a), two covers (capacity 24, carrying 12 fibres) come into a vault and leave unchanged. In figure 2 (b), the



(a) Incoming covers left alone (b) Incoming covers welded together

Figure 2: Dealing with incoming covers

two covers are welded together into a new, larger cover (capacity 48) that carries the total fibres from the covers (24 fibres). Note that a cover with capacity 24 could carry all the fibre, but as already mentioned this is not possible.

Welding Costs In the Cover Optimisation problem, it costs nothing to place fibres from the street into a new cover (although the cover will incur a cost) or to side-weld them into an existing cover (both these costs have been considered in the street optimisation). However, if two incoming covers are welded together, there is a fixed cost for performing the welding and a unit weld cost for each fibre placed into the new cover. To make maintenance easier, no fibres from an incoming cover may be split amongst leaving covers.

3 Dynamic Programming Formulation

In this section we outline our DP formulation for solving the Cover Optimisation problem.

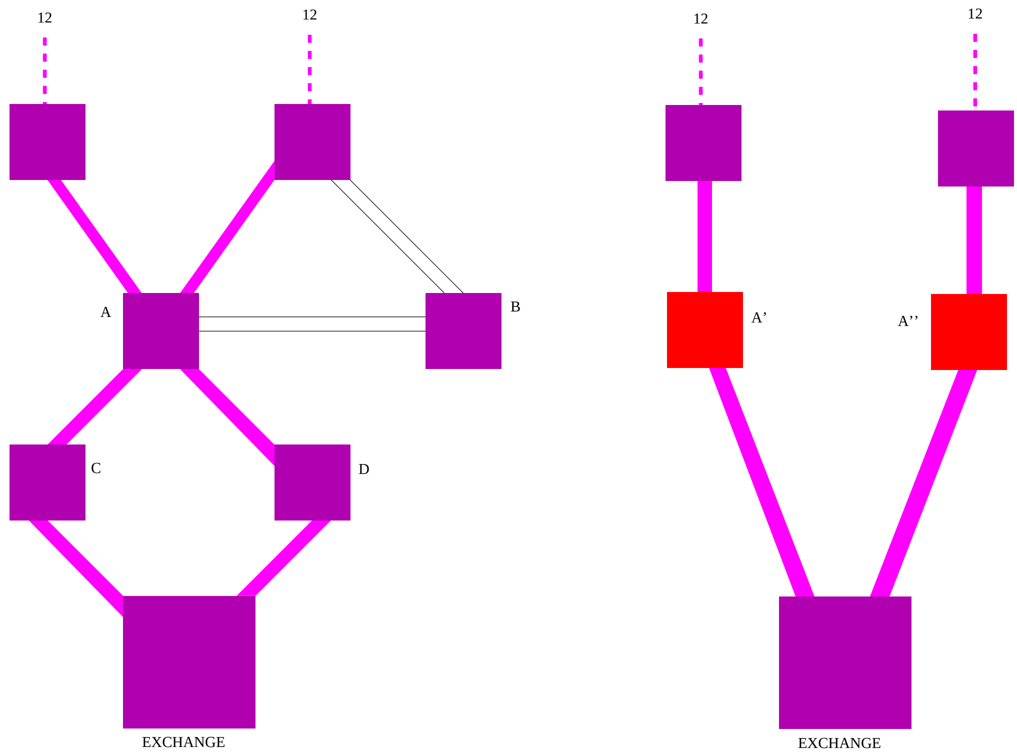
3.1 The Logical Tree

Since fibres within a cover may not be split, we can solve the Cover Optimisation problem using DP on a *logical tree*. To convert the network into a logical tree we:

1. remove any feeders and vaults that don't contain fibres;
2. remove any vaults with only one feeder coming in and one feeder coming out (no welding will ever occur at this vault);
3. duplicate any vault and/or feeder that occurs on more than one divergent path (to the exchange).

Figure 3 gives an example of a network and its corresponding logical tree. Vault A contains fibres that are following two divergent paths to the exchange (through vaults C and D, respectively), so it is duplicated (to make node A' and A") in the logical tree. Vault B and its incident feeders don't contain any fibres, so they are removed from the logical tree. Vaults C and D do contain fibres, but only have a single feeder in and out, so they are removed from the logical tree.

Each arc in the logical tree starts at a *node* (representing a vault in the original network).



(a) Example Network (b) Corresponding Logical Tree

Figure 3: Logical Tree Example

3.2 The DP formulation

The DP formulation uses the logical tree and decides how to cover the fibres along each arc. This formulation improves upon the initial approach from FiDO, where a first-fit decreasing bin-packing heuristic was used to pack fibre into covers. The heuristic does not consider the cost of the covers or welding when packing the fibres.

The total cost of covering the entire network is simply the sum of covering each branch leading into the exchange(s). The cost of covering a branch is the cost of covering the last arc in the branch, plus the cost of welding at the start of that arc plus the cost of covering the remainder of the branch. This decomposition gives rise to the DP recursion we use to find the minimum-cost covering of the entire network.

Choosing a combination of covers for an arc affects how we cover any incoming arcs. Also, the cost of welding depends on this decision. However, the set of possible combinations of covers for any arc is not immediately obvious from the number of fibres travelling along that arc. This set is determined by the sets both further from and closer to the exchange. However, we are able to determine bounds on the fibre capacity that needs to be covered along any arc. Using these bounds we can construct the set of all possible combinations of covers along every arc. These sets make up the state space for the DP formulation.

Determining the State Space We find bounds on the fibre capacity for any combination of covers along an arc using two quantities: minimum effective capacity and maximum effective capacity. We will demonstrate the calculations required to determine these quantities using the subnetwork depicted in figure 4. Each arc is labelled and displayed below its label is the minimum effective capacity/maximum

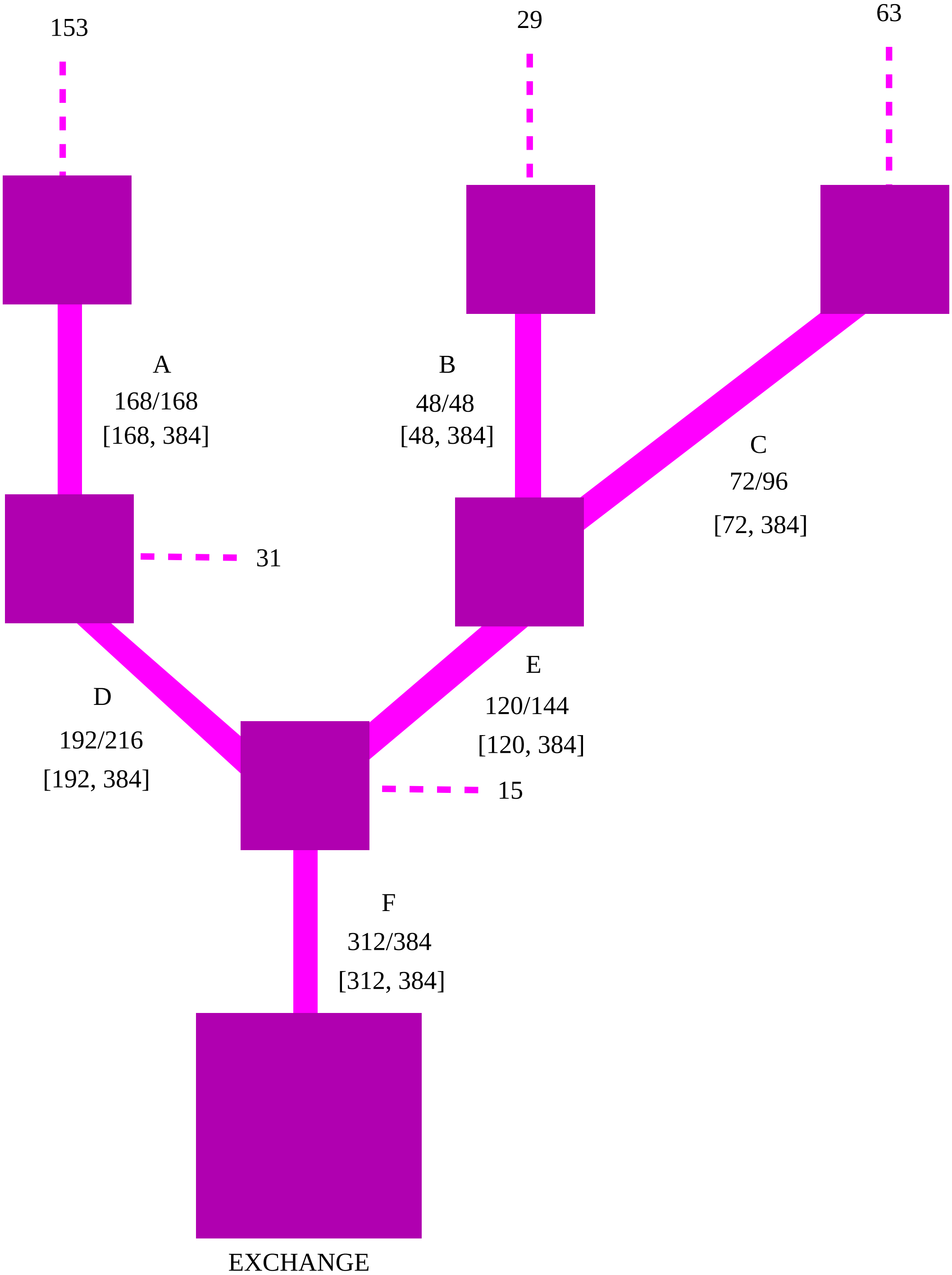


Figure 4: Determining the state space

effective capacity and [lower bound, upper bound] for the combinations of covers.

The *minimum effective capacity* of an arc is a lower bound on the fibre capacity for any combination of covers used along that arc. We determine the minimum effective capacity for a arc in two steps. First, we find the maximum of the total fibres along the arc and the sum of minimum effective capacities along any incoming arcs. Then, we find the minimum capacity of any combination of covers that can hold that maximum. Any arc that starts at a leaf node of the logical tree has no incoming arcs, so we can calculate the minimum effective capacity for all arcs by starting from the leaf nodes and moving towards the exchange.

For arcs A, B, and C, the minimum effective capacity is the minimum capacity for any combination of covers that holds the new fibres: 153 fibres may be contained in one 144 (fibre capacity cover) and one 24; 29 fibres may be covered in one 48 and 63 fibres may be covered in one 48 and one 24. Along arc D, the incoming arcs (arc A) have a total minimum effective capacity of 168, but the arc carries 184 fibres (153 + 31), so we find the combination of covers with minimum capacity, one 144 and one 48. Along arc E there are no new fibres, so the minimum effective capacity is the sum of the minimum effective capacities of the incoming arcs (48 + 72 = 120). Along arc F there are new fibres, but the sum of the incoming minimum effective capacities (192 + 120 = 312) is greater than the total fibres along the arc (184 + 92 + 15 = 291). Therefore, the minimum effective capacity is 312.

The *maximum effective capacity* of a arc is an upper bound on the fibre capacity for any combination of covers used along that arc. Given only new fibre along a arc, we would never provide extra fibre capacity along an arc unless it was cheaper to do so. Therefore, the maximum capacity of any combination of covers along the arc is given by the cheapest combination of covers (in terms of cost per unit length) that holds the new fibres. Given a node with incoming arcs and (possibly) new fibres, the maximum capacity of any combination of covers along the outgoing arc would arise if all the new fibres were covered separately. By adding the maximum effective capacity of the incoming arcs to the new fibres then finding the cheapest combination of covers for that sum we get the maximum effective capacity for the arc itself. Again, we can calculate the maximum effective capacity for all the arcs by starting from the leaf nodes and moving towards the exchange.

For arcs A, B, and C, the maximum effective capacity is given by the capacity for the cheapest combination of covers that holds the new fibres: 168 for arc A (one 144 and one 24); 48 for arc B (one 48); and 96 for arc C (one 96). Along arc D, the 31 new fibres may be covered cheaply with a 48 giving maximum effective capacity of 168 + 48 = 216. Along arc E, there are no new fibres so the maximum effective capacity is 48 + 96 = 144 (the sum of the maximum effective capacities of the incoming arcs). Finally, along arc F the new fibres may be covered cheaply by a 24 so the maximum effective capacity is 216 + 144 + 24 = 384.

Unfortunately, the maximum effective capacity does not account for welding. We may sometimes add unused capacity along a arc if it reduces welding costs later. We don't need to worry about welding costs at those arcs incident to the exchange (*exchange arcs*), but all other arcs are affected. However, we would never use a combination of covers along an arc with greater capacity than the combination of covers used into the exchange. We get a true upper bound on the fibre capacity that needs to be covered along any arc by using the maximum effective capacity of the exchange arc it (eventually) feeds into.[1]

The maximum effective capacity of the exchange arc (F) becomes the upper bound for all the arcs feeding into it (A, B, C, D, and E), so the fibre capacity bounds for combinations of covers along all the arcs is shown. For arc F this means the possible combinations of covers along the arc are: $9\times24, 7\times24$ and $1\times48, \ldots, 2\times 144$ and 1×96.

Given the bounds on the fibre capacity for any combination of covers along an arc, we can then find all possible combinations of covers that lie within those bounds. These combinations represent the different DP states that may be present along this arc.

3.3 Solving the DP formulation

Now that we know all states (possible combinations of covers) for every arc we can solve our DP recursion. Each branch of the logical tree may be solved separately,[2] starting at the exchange arc and moving out through the rest of the branch. The cost of selecting a state for an arc is given by the cost of laying the associated combination of covers along the arc added to the minimum cost of selecting states for the incoming arcs and welding the associated covers into the covers along the arc itself.

The more states (combinations of covers) for each arc, the larger the computational cost of solving the DP recursion. For the problems considered (the Auckland and Wellington CBDs), we found the DP recursion intractable. To overcome this, we set a user-specified limit on the number of covers in any combination of covers. Using this limit to (significantly) reduce the state space allowed us to solve the DP recursion in a reasonable time. As we increased the limit on the number of covers the DP recursion took longer to solve, but the solution becomes closer to optimal. In §5 we show that the effect of increasing this limit decreases rapidly.

3.4 Guaranteeing Optimality

The solutions we obtained from DP provided cheaper coverings than the FiDO heuristic (see §5). However, even if we could achieve an optimal solution from DP, it would not necessarily be optimal for the network covering problem. When forming the logical tree, nodes were sometimes duplicated (if they appeared on more than one divergent path, see §3.1). The DP recursion relies on these nodes being considered separately. However, the fixed welding cost means this may not always be the case. If DP decides to weld at each copy of a single node it will pay this fixed cost twice. In reality, there is only a single vault, so there will only be a single fixed cost for welding.

Therefore, DP solves a slightly different problem from the actual network covering problem. We can recalculate the cost of the DP solutions to properly account for fixed weld costs, but the DP will still be guided by a slightly inaccurate costing. We were unsure how great an effect this inaccuracy was having on the solutions. We used IP to remove this inaccuracy while solving the DP formulation. We could not

[1] There are tighter upper bounds that may be calculated, but the time to calculate these bounds is similar to the solution time for DP.

[2] Parallel processing could significantly reduce the time required to solve the DP recursion.

solve the problem as quickly with IP, but it offered us a comparison between the actual optimal solution and the DP solution. We next describe the IP formulation in §4.

4 Integer Programming Formulation

Our IP formulation uses the framework from the DP formulation (including the user-specified limit on the number of covers). We still use the logical tree structure (including incoming arcs), but different arcs are allowed to start at the same node. This change is possible because IP does not need the tree structure required by the DP recursion. Binary variables represent decisions about the combination of covers used along each arc and any welding used at each node. Other constraints and variables ensure that the covers of incoming arcs can be welded into the subsequent covers.

We describe the IP formulation in detail throughout the rest of this section.

4.1 Definitions

The DP formulation generates the following sets:

$$\begin{aligned}
A &= \text{the set of all (logical) arcs;}\\
N &= \text{the set of all nodes;}\\
T &= \text{the set of all cover types;}\\
P(a) &= \text{the set of possible coverings for arc } a \in A;\\
I(a) &= \text{the set of incoming arcs into arc } a \in A;\\
S(n) &= \text{the set of arcs that start at node } n \in N;
\end{aligned}$$

with the following parameters:

$$\begin{aligned}
\psi_{\text{unit}} &= \text{the unit weld cost;}\\
\psi_{\text{fixed}} &= \text{the fixed weld cost;}\\
\nu(a) &= \text{the number of new fibres entering the network along arc } a \in A;\\
\lambda(a) &= \text{the length of arc } a \in A;\\
\gamma(t) &= \text{the fibre capacity of cover type } t \in T;\\
\pi(t) &= \text{the cost per unit length of cover type } t \in T;\\
\mu(p,t) &= \text{the number of covers of type } t \in T \text{ in covering } p \in \cup_{a\in A} P(a).
\end{aligned}$$

4.2 Decision Variables

Given the previous definitions we define the following decision variables

$$\begin{aligned}
z_{ap} &= 1 \text{ if covering } p \in P(a) \text{ is used on arc } a \in A, 0 \text{ otherwise;}\\
d_{atu} &= \text{the number of covers of type } t \in T \text{ that transfer to type } u \in T\\
&\quad\ \text{(with } \gamma(u) > \gamma(t)\text{) along arc } a \in A;\\
s_{at} &= \text{the number of covers of type } t \in T \text{ that start along arc } a \in A;\\
f_a &= \text{the fibre count at the start of arc } a \in A;\\
w_n &= 1 \text{ if there is a weld at node } n \in N.
\end{aligned}$$

4.3 Objective Function

We want to minimise the total cost of covering the network including any welds that need to be made.

$$Z = \sum_{a \in A} \sum_{p \in P(a)} \sum_{t \in T} \underbrace{\mu(p,t)\pi(t)\lambda(a)z_{ap}}_{\substack{\text{cost of using covering } p \\ \text{along arc } a}} + \sum_{n \in N} \psi_{\text{fixed}} w_n + \sum_{a \in A} \sum_{t \in T} \sum_{\substack{u \in T \\ \gamma(u) > \gamma(t)}} \psi_{\text{unit}} \gamma(t) d_{atu}$$

Note that the last term in the objective function represents the total unit cost of welding. The number of covers of type $t \in T$ that transfer to another type along $a \in A$ is given by d_{atu}, but each cover $t \in T$ requires $\gamma(t)$ welds to transfer.

4.4 Unique Covering Constraints

Each arc can only use one covering.

$$\sum_{p \in P(a)} z_{ap} = 1, a \in A$$

4.5 Demand-Supply Constraints

We need to ensure that all covers coming into a arc $a \in A$ either continue along c or transfer to another cover type.

$$\sum_{i \in I(a)} \sum_{p \in P(i)} \mu(p,t) z_{ip} - \sum_{\substack{u \in T \\ \gamma(u) > \gamma(t)}} d_{atu} = \sum_{p \in P(a)} \mu(p,t) z_{ap} - s_{at}, a \in A, t \in T$$

The left side of the constraint counts the covers of type t from the incoming arcs that continue without transferring. The right side of the constraint counts covers of type t that don't start along arc a, i.e., continue from an incoming arc. Since arcs that don't transfer continue along the arc these two quantities must be equal. This constraint determines the "demand" and "supply" for covers of type t along arc a.

We include the following constraint to ensure that any transferring covers move into a new cover.

$$\sum_{\substack{u \in T \\ \gamma(u) < \gamma(t)}} \gamma(u) d_{aut} \leq \gamma(t) s_{at}, a \in A, t \in T$$

It forces the "supply" of fibre capacity for a particular cover type t along a to be more than the "demand" for that fibre capacity from transferring covers.

4.6 Fibre Covering Constraints

The total fibre leaving an arc is equal to the fibre coming into the arc plus any new fibres along the arc (i.e., fibres are *conserved*).

$$\nu(a) + \sum_{i \in I(a)} f_i = f_a, a \in A$$

The fibre leaving an arc must fit into the covering along that arc.

$$\sum_{p \in P(a)} \sum_{t \in T} \mu(p,t) \gamma(t) z_{ap} \geq f_a, a \in A$$

4.7 Welding Constraint

If there is any transfer of covers along an arc then there is a weld at the start node for that arc.

$$\sum_{a\in S(n)}\sum_{t\in T}\sum_{\substack{u\in T\\ \gamma(u)<\gamma(t)}} d_{atu} \leq M w_n, n \in N, M \text{ large}$$

4.8 Problem-Specific Constraints

Finally, when we used our IP formulation on a problem with four cover types from table 1 we noticed some infeasible welding decisions. Specifically, 3 covers of capacity 96 would be welded into 2 covers of capacity 144. Since fibres from existing covers cannot be split, this welding should be banned. The following constraint ensures that at most one 96 cover will be placed into every 144 cover.

$$d_{a,96,144} \leq s_{a,144}, a \in A$$

4.9 Formulation Summary

The complete IP formulation is given below.

$$\min Z = \sum_{a\in A}\sum_{p\in P(a)}\sum_{t\in T}\mu(p,t)\pi(t)\lambda(a)z_{ap} + \sum_{n\in N}\psi_{\text{fixed}}w_n + \sum_{a\in A}\sum_{t\in T}\sum_{\substack{u\in T\\ \gamma(u)>\gamma(t)}}\psi_{\text{unit}}\gamma(t)d_{atu}$$

$$\text{subject to} \quad \sum_{p\in P(a)} z_{ap} = 1, a \in A$$

$$\sum_{i\in I(a)}\sum_{p\in P(i)}\mu(p,t)z_{ip} - \sum_{\substack{u\in T\\ \gamma(u)>\gamma(t)}} d_{atu} = \sum_{p\in P(a)}\mu(p,t)z_{ap} - s_{at}, a \in A, t \in T$$

$$\sum_{\substack{u\in T\\ \gamma(u)<\gamma(t)}}\gamma(u)d_{aut} \leq \gamma(t)s_{at}, a \in A, t \in T$$

$$\nu(a) + \sum_{i\in I(a)} f_i = f_a, a \in A$$

$$\sum_{p\in P(a)}\sum_{t\in T}\mu(p,t)\gamma(t)z_{ap} \geq f_a, a \in A$$

$$\sum_{a\in S(n)}\sum_{t\in T}\sum_{\substack{u\in T\\ \gamma(u)<\gamma(t)}} d_{atu} \leq M w_n, n \in N, M \text{ large}$$

$$d_{a,96,144} \leq s_{a,144}, a \in A$$

$$z_{ap} \in \{0,1\}, a \in A, p \in P(a),$$

$$d_{atu} \in \mathbb{Z}^+, a \in A, t \in T, u \in T, \gamma(t) < \gamma(u),$$

$$s_{at} \in \mathbb{Z}^+, a \in A, t \in T,$$

$$w_n \in \{0,1\}, n \in N$$

5 Results

FiDO was developed for the Windows platform (in Microsoft Visual Studio). Using much of the FiDO source code, we wrote a DP application in Microsoft Visual C++ and solved the network covering problem on an AMD Athlon 1400+ (256 MB RAM). We solved the IP formulation using CPLEX 6.6.0 in Red Hat Linux 8.0 running on a dual processor 550 MHz Pentium III (256 MB RAM).

Two different CBDs were considered, Auckland and Wellington, with the four cover types in table 1. Additionally, we considered four different cases of *tubification* (see table 2). As briefly described in §1, fibres are placed into tubes separately for each street (*street tubification*) and (may be) aggregated into tubes again at the vault (*vault tubification*). Street tubification rounds the demand for each street to a multiple of the street tube size. Vault tubification rounds the total demand at a vault to a multiple of the vault tube size. We can then work in multiples of the tubification, e. g., with `FibCost12,0` tubification, a street demand of 22 is rounded to 24 and then becomes a demand of 2 and the cover capacities become 2 (24), 4 (48), 8 (96) and 12 (144). The welding costs are given as $\psi_{\text{unit}} = \$18$ and $\psi_{\text{fixed}} = \$400$.

Tubification	Street Tube Size	Vault Tube Size
`FibCost1,0`	1	None
`FibCost1,1`	1	1
`FibCost12,0`	12	None
`FibCost1,12`	1	12

Table 2: Different tubifications

Finally, we tested cover limits of 1, 2, 3, 4, and 5.

While developing the DP application we noticed that FiDO would always place new fibres into a single cover. In many cases this pushed the cost higher, so we changed FiDO to allow multiple covers to be used for new fibres and achieve significantly cheaper coverings. The results for the old version of FiDO, our new version of FiDO, the DP application and the IP formulation are given in tables 3 and 4. Note that the cost for the DP formulation is the actual cost of covering the network with the multiple fixed cost charges removed.

It is obvious from tables 3 and 4 that DP outperforms FiDO when more than 2 covers are allowed along the arcs. It is interesting to observe the change in the improvement as the number of covers increases. Figures 5 and 6 show the percentage improvement against the solution time for Auckland and Wellington, respectively.

From these plots we see that as the limit on the number of covers increases the amount of time required to solve using DP grows far more quickly than the improvement in the cost of the solution. The plots indicate the best choice for the limit in these examples is 3–4. Beyond that the saving does not justify the computational expense.

Our results show that when IP terminates at optimality, it gives a cheaper covering for the network, but with an improvement of less that %0.1. Therefore, DP yields a solution that is very close to optimal, indicating that the way the DP formulation models welding does not have a significant effect. The high quality of the DP solutions is evident from IP's inability to outperform DP if it is terminated before optimality.

Cover Limit	DP Cost	Solution Time (ms)	IP Cost	Solution Time (ms)	Integer/Linear Gap Termination	% Improvement DP over new FiDO	% Improvement IP over DP
Auckland CBD `FibCost1,0`, old FiDO = 934650, new FiDO = 539761							
1	572656	12063	572288	12708240	0.00%	-6.09	0.06
2	516584	16641	516583	998970	0.00%	4.29	0.00
3	509012	129922	508856	284330	0.00%	5.70	0.03
4	505613	34971265	505393	370180	0.00%	6.33	0.04
5	503611	66057755	503391	217490	0.00%	6.70	0.04
Auckland CBD `FibCost1,1`, old FiDO = 934650, new FiDO = 539761							
1	572656	12000	572288	12717130	0.00%	-6.09	0.06
2	516584	16719	516584	999660	0.00%	4.29	0.00
3	509012	1290993	508856	291210	0.00%	5.70	0.03
4	505613	34936294	505393	369710	0.00%	6.33	0.04
5	503611	66123813	503391	218400	0.00%	6.70	0.04
Auckland CBD `FibCost1,12`, old FiDO = 955778, new FiDO = 758074							
1	752121	1984	752104	893600	0.00%	0.79	0.00
2	722912	2969	765182	13630170	6.72%	4.64	-5.52
3	708379	57344	761521	13542410	8.42%	6.56	-6.98
4	703227	1009219	728108	23615100	4.90%	7.24	-3.42
5	701813	26786922	745532	43121210	7.28%	7.42	-5.86
Auckland CBD `FibCost12,0`, old FiDO = 1042179, new FiDO = 827815							
1	807699	1969	807411	62000	0.00%	2.43	0.04
2	770500	2843	769817	4360160	0.00%	6.92	0.09
3	763029	16594	766779	33988290	1.51%	7.83	-0.49
4	761867	881812	814878	17889920	7.87%	7.97	-6.51
5	759437	25580031	790617	18004720	5.56%	8.26	-3.94

Table 3: Comparing IP and DP for Auckland

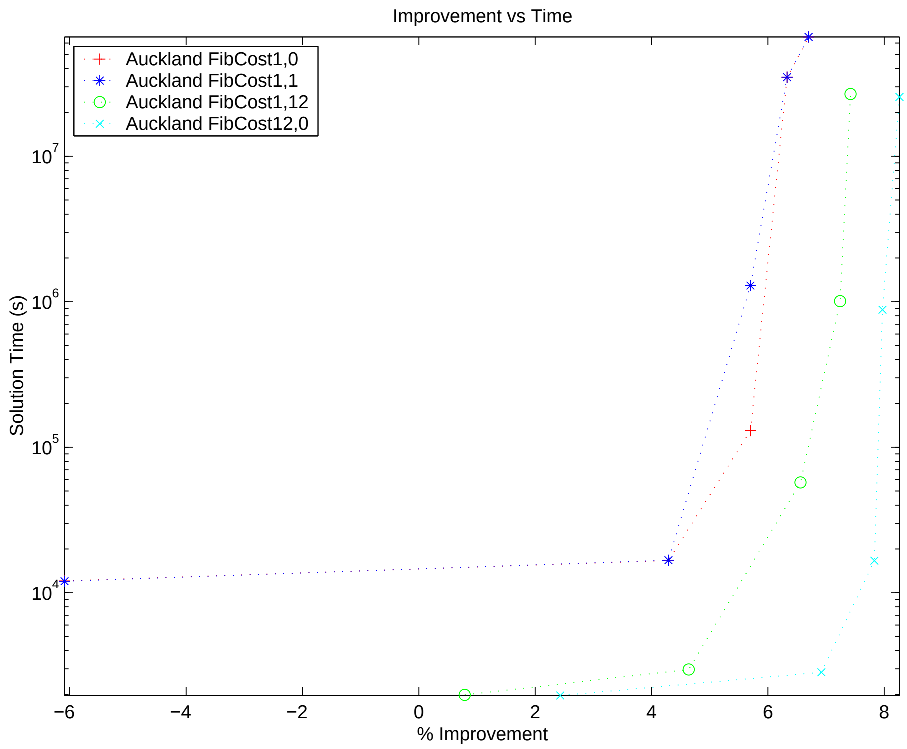


Figure 5: Improvement using DP compared to solution time for Auckland CBD

Cover Limit	DP Cost	Solution Time (ms)	IP Cost	Solution Time (ms)	Integer/Linear Gap Termination	% Improvement DP over new FiDO	% Improvement IP over DP
Wellington CBD `FibCost1,0`, old FiDO = 1158823, new FiDO = 945839							
1	920216	13672	920216	9782	0.00%	2.71	0.00
2	895370	14187	895036	76636	0.00%	5.34	0.04
3	891815	18062	891223	309726	0.00%	5.71	0.07
4	890962	59156	892759	12352070	0.40%	5.80	-0.20
5	890637	387078	900267	8456020	1.40%	5.84	-1.07
Wellington CBD `FibCost1,1`, new FiDO = 945839							
1	920216	13703	920216	9840	0.00%	2.71	0.00
2	895370	14219	895036	75993	0.00%	5.34	0.04
3	891815	18110	891223	310120	0.00%	5.71	0.07
4	890962	59328	892759	12352070	0.40%	5.80	-0.20
5	890637	82984	900267	8456020	1.40%	5.84	-1.07
Wellington CBD `FibCost1,12`, new FiDO = 1108452							
1	1082256	1422	1082143	5374	0.00%	2.36	0.01
2	1049211	1657	1048242	22380	0.00%	5.34	0.09
3	1039160	4109	1039126	110430	0.00%	6.25	0.00
4	1038096	22453	1041268	6085940	0.34%	6.35	-0.30
5	1037774	259313	1038831	9531690	0.17%	6.38	-0.10
Wellington CBD `FibCost12,0`, old FiDO = 1256389, new FiDO = 1341087							
1	1214719	1828	1214618	6682	0.00%	3.32	0.01
2	1188281	1938	1188138	24080	0.00%	5.42	0.01
3	1179657	4656	1179416	243020	0.00%	6.11	0.02
4	1179403	17578	1179244	6916200	0.00%	6.13	0.01
5	1179403	97062	1185260	11626380	0.62%	6.13	-0.49

Table 4: Comparing IP and DP for Wellington

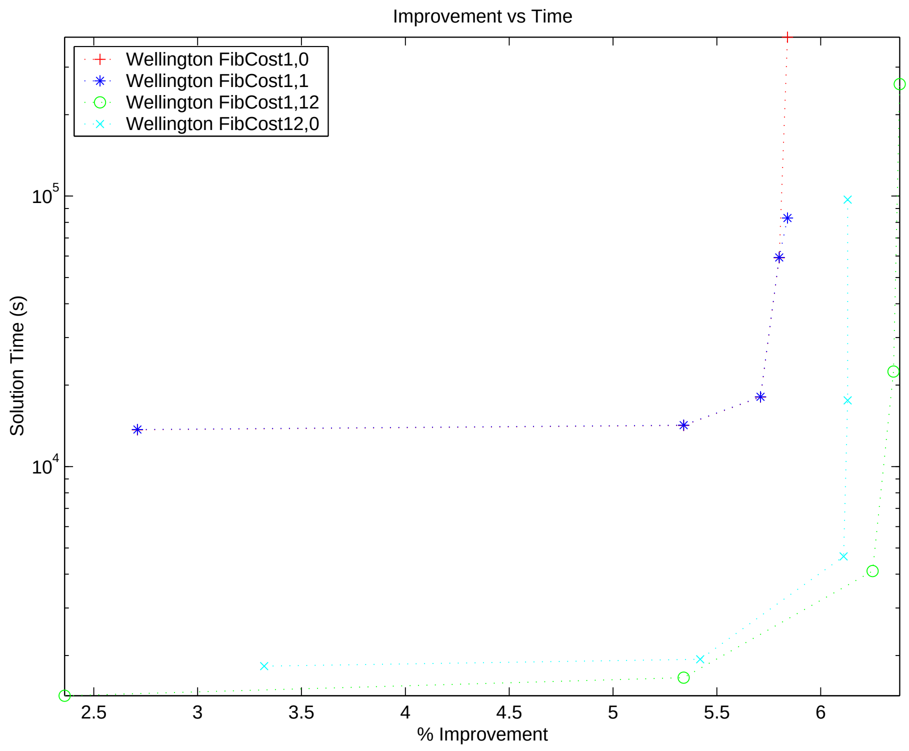


Figure 6: Improvement using DP compared to solution time for Wellington CBD

The running times for IP are interesting for the Auckland CBD with tubifications `FibCost1,0` and `FibCost1,1`. As the limit on the number of covers increases, the running times decrease. This behaviour is unexpected and is not observed in the other cases. Auckland is the larger network, and these two cases have no reduction due to tubification (as explained earlier in this section). We think that the loss in flexibility caused by tubifying fibres makes the other Auckland cases harder to solve, even as the limit on the number of covers increases. The topology of the Wellington network makes it hard to solve, even with the flexibility from tubifications `FibCost1,0` and `FibCost1,1`. For these cases, it may be that the running times start to decrease as the cover limit gets even higher. This may also be true for the Auckland CBD with tubifications `FibCost1,12` and `FibCost12,0`, but memory limitations on our computers do not allow us to determine this.

6 Conclusions

The Cover Optimisation problem is a generalised set covering problem with additional side constraints, complicated by a mixture of fixed and variable costs. We developed two approaches for solving this problem, one using a DP formulation and the other using a similar IP formulation.

The DP formulation performed better than the existing heuristic in FiDO, indicating the savings that can be made when solving the Cover Optimisation problem. Also, our comparison with IP shows how close to optimal the DP solutions are. In fact, when IP did not achieve optimality (because of early termination), DP gives a cheaper network covering. Furthermore, the computational performance of the DP model was significantly better than IP in most cases, a feature that became more pronounced as the problem size increased. For this reason we conclude that DP is a practical approach for obtaining near-optimal solutions for the Cover Optimisation problem in reasonable time.

References

[1] A. Atamtürk. On capacitated network design cut-set polyhedra. Technical report, IEOR Department, University of California at Berkeley, December 2000. Available at http://ieor.berkeley.edu/ atamturk.

[2] D. Bienstock, S. Chopra, and O. Gunluk. Minimum cost capacity installation for multicommodity network flows. *Mathematical Programming*, 81(2–1):177–199, 1998.

[3] D. Bienstock and O. Gunluk. Capacitated network design. Polyhedral structure and computation. *INFORMS Journal on Computing*, 8:243–259, 1996.

[4] S. Chopra, I. Gilboa, and S. T. Sastry. Source sink flows with capacity installation in batches. *Discrete Applied Mathematics*, 85:165–192, 1998.

[5] M.R. Garey and D. S. Johnson. *Computers and intractability: A guide to the theory of NP-completeness*. W. H Freeman and Company, San Francisco, CA, 1979.

[6] J.W. Herrmann, G. Ioannou, I. Minis, and J.M. Proth. A dual ascent approach to the fixed-charge capacitated network design problem. *European Journal of Operational Research*, 95:476–90, 1996.

[7] K. Holmberg and Di Yuan. A Lagrangean heuristic based branch-and-bound method for the capacitated network design problem. In *Proceedings of International Symposium on Operations Research*, pages 78–83, September 1996.

[8] T. L. Magnanti, P. Mirchandani, and R. Vachani. Modeling and solving the capacitated network loading problem. *Operations Research*, 43:142–157, 1995.

[9] A. J. Mason and A. B. Philpott. Development of FIDO - A Network Design Tool for Fibre Cable Layout. In *36th Annual ORSNZ Conference*. ORSNZ, 2001.

[10] M.L. O'Sullivan. Optimal Fibre-Optic in Cable Layout using Dynamic Programming. In *37th Annual ORSNZ Conference*. ORSNZ, 2002.

[11] A. B. Philpott and S. A. Miller. A cable layout model for telecomunnication distribution networks. In *31st Annual ORSNZ Conference*. ORSNZ, 1995.